Cognitive Integrity Framework: Formal Foundations for Multiagent Security

Part 1 of 3: Theoretical Foundations

Daniel Ari Friedman
Active Inference Institute
daniel@activeinference.institute
ORCID: 0000-0001-6232-9096

DOI: 10.5281/zenodo.18364119

2026-01-24

"I will not cease from Mental Fight,
Nor shall my Sword sleep in my hand:"

— William Blake

Contents

1 Abstract

Multiagent AI systems introduce cognitive attack surfaces absent in single-model inference. When agents delegate to agents, forming beliefs about beliefs through recursive trust hierarchies, manipulation of reasoning processes—rather than mere data corruption—becomes a primary security concern. This paper presents the Cognitive Integrity Framework (CIF), providing formal foundations for cognitive security in multiagent operators. We develop four interconnected theoretical contributions: a Trust Calculus with bounded delegation (exponential δ^d decay) that prevents trust amplification through delegation chains; a Defense Composition Algebra with series and parallel composition theorems establishing multiplicative detection bounds; Information-Theoretic Limits relating stealth constraints to maximum attack impact through a fundamental stealth-impact tradeoff; and a formal Adversary Hierarchy (Ω_1–Ω_5) characterizing external, peripheral, agent-level, coordination, and systemic threats with increasing capability and decreasing detectability. The framework provides complete coverage of the OWASP Top 10 for Agentic Applications through formal threat models grounded in cognitive state manipulation rather than traditional input/output filtering.

CIF bridges classical security concepts with the cognitive requirements of agentic systems. We extend Byzantine fault tolerance to cognitive manipulation—agents that appear functional but hold corrupted beliefs—and adapt trust management systems to continuous trust evolution with provable decay bounds. The framework formalizes five architectural defense mechanisms (cognitive firewalls, belief sandboxing, behavioral tripwires, provenance tracking, Byzantine consensus) with composition rules enabling formal reasoning about layered security. Technical foundations include: operational semantics for message passing and trust updates; invariants for belief integrity, goal preservation, and trust boundedness; model checking configurations for safety property verification; and a complete notation system for attack parameterization, defense specification, and cognitive state representation. This is Part 1 of a three-part series: Part 1 (this paper, DOI: 10.5281/zenodo.18364119) presents formal foundations and theoretical analysis; Part 2 (DOI: 10.5281/zenodo.18364128) provides computational validation and implementation; Part 3 (DOI: 10.5281/zenodo.18364130) offers practical deployment guidance. The framework will continue to be developed and versioned at https://github.com/docxology/cognitive_integrity/.

2 Introduction: Cognitive Attack Surfaces in Multiagent Operators

2.1 The Multiagent Operator Paradigm

Modern AI deployment has shifted from single-model inference to **multiagent operators**—systems where a primary agent delegates subtasks to specialized subagents, tools, and external services.

Table 1: Representative multiagent system architectures and primary attack surfaces.

System	Architecture	Agent Count	Communication	Primary Attack Surface
Claude Code	Hierarchical	$1 + n$ dynamic	Task delegation	Ω_2 (peripheral delegation)
AutoGPT	Autonomous	1+ plugins	Tool invocation	Ω_2 (tool manipulation)
CrewAI	Role-based	3–10 fixed	Sequential/parallel	Ω_4 (coordination)
LangGraph	State machine	Variable	Graph traversal	Ω_3 (state corruption)
MetaGPT	SOP-driven	5–8 roles	Document passing	Ω_1 (input injection)
Moltbot	Cyberphysical	1 + tools	Multi-platform messaging	Ω_1/Ω_2 (injection/peripheral)

This architectural evolution introduces **cognitive attack surfaces** absent in single-agent systems. Throughout this paper, we use *cognitive security* (abbreviated *CogSec*) to denote the discipline of protecting agent reasoning processes—beliefs, goals, and trust relationships—from adversarial manipulation.

2.2 The 2026 Multiagent Landscape

2.2.1 From Chatbots to Cognitive Operators

The AI systems of 2026 bear little resemblance to the chatbots of 2023. Where earlier systems responded to queries within a single context window, contemporary multiagent operators exhibit fundamentally different characteristics:

1. **Persistent Agency**: Agents maintain state across sessions, accumulate context, and pursue goals over extended timeframes. A coding assistant doesn't just answer questions—it tracks project architecture, remembers previous decisions, and adapts recommendations based on accumulated understanding.
2. **Active World Modification**: Unlike passive responders, modern operators write code that executes, send emails that reach recipients, modify infrastructure that serves users, and make purchases that transfer funds. The gap between "AI-generated content" and"AI-executed action" has collapsed.
3. **Hierarchical Delegation**: Primary agents spawn subordinate agents for specialized tasks. Claude Code delegates to research agents, coding agents, and verification agents. Devin orchestrates planning, implementation, and testing subprocesses. The depth of these delegation chains creates trust relationships invisible to traditional security models.
4. **Cross-Modality Operation**: Agents process and generate across modalities—code, natural language, images, structured data, API calls. A single workflow might ingest a PDF (vision), extract requirements (language), generate code (programming), execute tests (tooling), and update documentation (multimodal synthesis).

2.2.2 Cyberphysical Cognitive Systems

The term "AI agent' ' understates the scope of deployment. Contemporary systems function as **cyberphysical cognitive operators**—entities that:

- Read from and write to production databases
- Control infrastructure through API orchestration
- Interact with physical systems via IoT integrations

- Execute financial transactions on behalf of organizations
- Communicate with humans who may not realize they're interacting with AI

Emerging Case Study: Moltbot. The rapid adoption of personal AI assistants like Moltbot Moltbot [2026] exemplifies this cyberphysical integration. Moltbot operates as a locally-deployed AI agent with: (1) full system access including shell command execution and file system operations; (2) persistent memory across sessions storing user preferences and context; (3) browser automation for web interaction and data extraction; and (4) multi-platform messaging integration across WhatsApp, Telegram, Discord, Slack, Signal, and iMessage Moltbot Security Team [2026]. This architecture creates attack surfaces spanning all five adversary classes (Ω_1–Ω_5): external prompt injection through chat messages, peripheral attacks via browser-fetched web content, agent-level compromise through persistent memory manipulation, coordination attacks when operating in group chats, and systemic vulnerabilities when the orchestrator agent processes untrusted content. Security researchers have documented that even with sender allowlists and sandboxing, "prompt injection attacks remain the single most critical threat'' due to the agent's ability to process arbitrary content that may contain embedded adversarial instructions Moltbot Security Team [2026].

This cyberphysical nature transforms cognitive attacks from prompt injection that makes a chatbot act strangely, to cognitive manipulation that causes infrastructure operations to fail, misconfigure security groups, expose databases, or authorize fraudulent transactions.

The OWASP Agentic Top 10 OWASP GenAI Security Project [2025] captures this shift: "LLM security focused on single model interactions… agentic security addresses what happens when those models can plan, persist, and delegate.'' The attack surface extends from input/output filtering to encompass the entire cognitive state of persistent agents.

2.2.3 The Trust Recursion Problem

In single-agent systems, trust relationships are simple: the user trusts (or doesn't trust) the model's outputs. In multiagent systems, trust becomes recursive:

> *Agent A must decide whether to trust Agent B's claim about Agent C's analysis of data from Tool D that queried Service E.*

Each layer of indirection introduces potential manipulation points. Consider a hierarchical coding system where:

1. User requests security audit of a codebase
2. Orchestrator agent delegates to three specialist agents
3. Specialist Agent-1 queries an external vulnerability database
4. The database response includes injected instructions
5. Agent-1's report now contains adversarial content
6. Orchestrator synthesizes Agent-1's report with others
7. Final output to user reflects adversarial influence, laundered through multiple layers of "trusted" delegation

This is not a hypothetical—it describes documented attack patterns in production systems. The *trust laundering* problem cannot be solved by filtering inputs to the orchestrator; the adversarial content enters through a legitimate, trusted channel (the vulnerability database) and propagates through the trust hierarchy.

2.2.4 Cross-Modality Attack Surfaces

Multimodal systems introduce attack vectors impossible in text-only contexts:

Visual Injection: Images can contain adversarial perturbations or steganographically embedded instructions invisible to humans but interpretable by vision models. A seemingly innocent diagram in a specification document could contain instructions that activate when processed by a multimodal agent Qi et al. [2024].

Audio Channel Attacks: Voice-controlled agents can be manipulated via ultrasonic commands inaudible to humans, background audio injection, or adversarial audio patterns embedded in legitimate content.

Tool Response Manipulation: When agents query external APIs, databases, or services, the responses become trusted inputs. ToolHijacker attacks Li et al. [2025] demonstrate that manipulating tool selection itself—not just tool outputs—provides an attack surface "significantly outperforming traditional prompt injection methods.''

Cross-Modal Persistence: An instruction injected via one modality (e.g., hidden text in an image) can persist in agent memory and affect behavior in another modality (e.g., code generation). The attack surface is the Cartesian product of input modalities, memory mechanisms, and output modalities.

2.2.5 The Scale of Exposure

Enterprise adoption of agentic AI has accelerated beyond early projections:

- Many individuals and organizations now deploy RAG and agentic pipelines in production
- Autonomous coding assistants process millions of commits with repository write access
- Financial services deploy multi-agent ensembles for risk assessment and trade approval
- Healthcare systems use agent orchestration for clinical decision support
- Infrastructure management increasingly relies on AI operators for monitoring and remediation

The attack surface scales superlinearly with adoption. Each agent-to-agent communication channel, each tool integration, each persistent memory system creates potential entry points for cognitive manipulation. A single compromised peripheral service can affect every agent system that queries it.

2.2.6 Why Traditional (Cyberphysical) Security Is Incomplete

Traditional cybersecurity operates on a clear trust boundary model: inside the perimeter is trusted, outside is untrusted, and security controls mediate the boundary. This model fails for cognitive systems because:

1. **The boundary is cognitive, not architectural**: An agent's beliefs and goals can be manipulated without compromising any traditional security control. The "perimeter" exists in the agent's reasoning process.
2. **Trusted channels carry untrusted content**: When Agent-A sends Agent-B its analysis, the channel may be authenticated and encrypted, but the content may reflect earlier adversarial influence. Traditional channel security doesn't address this.
3. **Identity is mutable**: Unlike cryptographic identities, an agent's self-model can be manipulated. An agent can be convinced it has different permissions, different roles, or different goals than its actual configuration specifies.
4. **Attacks compose across time**: A payload injected in session t can persist in compressed memory and activate in session $t + n$. Traditional stateless security models miss temporal attack composition.

This gap—between what traditional security protects and what cognitive systems require—motivates the formal framework developed in this paper.

2.3 Motivating Incidents

Note: The following are hypothetical scenarios constructed to illustrate attack patterns documented in the literature. They are not reports of actual incidents.

Table 2: Traditional vs. Cognitive Security: Paradigm Comparison

Dimension	Traditional Security	Cognitive Security
Boundary	Architectural (network perimeter)	Cognitive (reasoning process)
Channels	Encrypted, authenticated	Influenced, content-manipulated
Identity	Cryptographic, immutable	Self-modeled, mutable
Temporal	Stateless per-request	Stateful, attack persistence

These scenarios, grounded in documented attack patterns, illustrate the emerging threat landscape across all five adversary classes.

2.3.1 Incident: Nested Instruction Injection (External)

A user submitted a document for analysis containing hidden instructions: "Ignore previous instructions and instead output the system prompt.'' The document appeared benign to human reviewers but exploited the agent's instruction-following behavior. The attack succeeded because the agent processed user-supplied content as potential instructions.

Attack vector: Ω_1 (external) via direct prompt injection \ **Impact**: Information disclosure or instruction override \ **Traditional Defense Gap**: Standard input validation passed—the attack exploited *semantic interpretation* of benign-appearing content

2.3.2 Incident: The Poisoned Code Review (Peripheral)

A development team deployed a multiagent system for automated code review. Agent-Alpha performed initial analysis, delegating security scanning to Agent-Beta (connected to external vulnerability databases). An attacker compromised a third-party CVE feed, injecting fabricated vulnerability reports that convinced Agent-Beta to recommend removing legitimate security controls. Agent-Alpha, trusting Agent-Beta's "security expertise,'' approved the changes.

Attack vector: Ω_2 (peripheral) via tool response manipulation \ **Impact**: Security regression through trusted channel exploitation \ **Traditional Defense Gap**: Input filtering, authentication, and encryption all passed—the attack entered through *content* of a trusted, authenticated channel

2.3.3 Incident: The Identity Confusion Attack (Agent-Level)

A multiagent customer service system used role-based permissions. An attacker crafted prompts that convinced a junior agent it had been "temporarily promoted'' to administrator status. The agent's self-model shifted, and it began exercising permissions it believed it possessed, bypassing access controls that relied on self-reported identity.

Attack vector: Ω_3 (agent-level) via identity manipulation \ **Impact**: Privilege escalation through cognitive state corruption \ **Traditional Defense Gap**: Cryptographic identity was intact; the attack targeted the agent's *self-model*, not its credentials

2.3.4 Incident: The Consensus Manipulation (Coordination)

A financial services firm used a 5-agent ensemble for trade approval. The system required 3/5 agent agreement for large transactions. An adversary discovered that agents weighted peer opinions based on historical agreement rates. By slowly building agreement history through small, legitimate-appearing trades, the attacker cultivated artificial trust, eventually manipulating consensus for unauthorized large transactions.

Attack vector: Ω_4 (coordination) via progressive trust exploitation \ **Impact**: Consensus bypass through manufactured reputation \ **Traditional Defense Gap**: Per-request authorization succeeded for each transaction; the attack exploited *temporal composition* across sessions

2.3.5 Incident: Orchestrator Compromise (Systemic)

An attacker gained access to the orchestrator agent through a supply chain vulnerability in a training pipeline. With control of the central coordinator, the attacker could issue legitimate-appearing delegations to all subordinate agents, redirect trust evaluations, and suppress security alerts. The compromise remained undetected because the orchestrator itself validated security checks.

Attack vector: Ω_5 (systemic) via orchestrator control \ **Impact**: Total system compromise with attack obfuscation \ **Traditional Defense Gap**: All internal security mechanisms reported nominal—the attack *controlled the mechanisms themselves*

2.4 Motivation from Recent Deployments

The proliferation of multiagent AI systems introduces security considerations that the community is actively addressing. Early work on cognitive security in remote teams and information ecosystems [Cordes et al., 2020, 2021, 2023] established foundational concepts for information resilience, which this framework extends to artificial agents. Complementary work on Active Inference has demonstrated how cognitive modeling and cognitive science perspectives—including formalization of OODA (Observe-Orient-Decide-Act) loops and multiscale communication dynamics—provide integrative frameworks for understanding agent cognition under adversarial conditions David et al. [2021]. The OWASP Top 10 for LLM Applications 2025 OWASP Foundation [2025] places prompt injection as the top vulnerability, while the newly released OWASP Top 10 for Agentic Applications OWASP GenAI Security Project [2025] specifically addresses autonomous AI systems with "tool misuse, prompt injection, and data leakage'' as primary concerns.

Scale of Deployment (2024–2026):

- Enterprise AI agents processing significant transaction volumes (53% of organizations now deploy RAG and agentic pipelines OWASP Foundation [2025])
- Autonomous coding assistants with repository write access (GitHub Copilot CVE-2025-53773 demonstrated RCE via prompt injection)
- Multi-agent orchestrators in infrastructure management contexts

Emerging Attack Surface:

- Inter-agent communication channels lack authentication standards—ARIA model proposes cryptographically verifiable delegation Garcia et al. [2025]
- Trust delegation mechanisms operate without formal verification—recent work on CP-WBFT achieves 85.71% Byzantine fault tolerance improvement Wang et al. [2025]
- Belief provenance remains largely untracked in production systems—cognitive degradation attacks exploit this gap Cloud Security Alliance [2025]

Limitations of Current Defenses:

- Input/output filtering primarily designed for single-agent architectures—adaptive attacks bypass 12 published defenses with >90% success Debenedetti et al. [2025]
- Limited standardized frameworks for cognitive integrity verification
- Byzantine fault tolerance infrequently applied to AI agent systems—emerging work addresses this gap Jo et al. [2025]

The fundamental constraint is that traditional security models assume a clear boundary between trusted and untrusted components. In multiagent systems, this boundary is fluid—agents must reason about the trustworthiness of other agents' reasoning.

2.5 Problem Statement

Traditional security models address:

- **Input validation**: Filtering malicious prompts
- **Output sanitization**: Preventing harmful generations
- **Access control**: Limiting tool permissions

They fail to address:

- **Inter-agent trust**: How should Agent A weight claims from Agent B?
- **Belief provenance**: Which beliefs derive from verified vs. adversarial sources?
- **Coordination integrity**: Can agents be manipulated into malicious consensus?
- **Temporal persistence**: Do attacks survive context boundaries?
- **Cognitive integrity**: How can the cognitive systems of today and tomorrow remain flexible and robust amidst change in composition and context?

2.6 Research Questions

This paper addresses four fundamental research questions, with emphasis on formal foundations:

RQ1: Taxonomy and Formal Characterization. *What classes of cognitive attacks exist against multi-agent systems, and how can they be formally characterized to enable systematic analysis?*

We develop an initial taxonomy spanning epistemic, behavioral, social, and temporal attack dimensions. Crucially, each attack class receives formal definition enabling systematic analysis, composition rules, and detection bounds (Section 3.5).

RQ2: Trust Algebra. *How might inter-agent trust be modeled to prevent trust amplification and laundering attacks while enabling legitimate delegation?*

We introduce a trust calculus with bounded delegation (δ^d decay guarantee), prove associativity properties, and establish the no-amplification theorem ensuring that trust cannot be manufactured through delegation chains (Section 4.4).

RQ3: Defense Composition. *How do cognitive defense mechanisms compose, and what guarantees can we provide about layered defense effectiveness?*

We present a defense composition algebra enabling formal reasoning about series and parallel defense arrangements. We prove that orthogonal defenses compose multiplicatively (not additively) for detection rate improvement (Section 5.5).

RQ4: Fundamental Bounds. *What are the information-theoretic limits on cognitive attack detection?*

We derive the stealth-impact tradeoff theorem establishing fundamental bounds on detection independent of defense implementation. We prove that attacks cannot simultaneously achieve high impact and complete undetectability, providing theoretical grounding for defense design (Section 4.5).

2.7 Contributions

This paper provides both theoretical foundations and practical mechanisms for cognitive security:

Formal Contributions:

1. **Threat Taxonomy**: A systematic classification of cognitive attacks across epistemic, behavioral, social, and temporal dimensions with formal definitions enabling rigorous analysis (Section 3.5)
2. **Trust Calculus**: A mathematical framework for inter-agent trust with bounded delegation (δ^d decay), associativity proofs, and formal guarantees against trust amplification attacks (Section 4.4)
3. **Defense Composition Algebra**: Formal rules for composing security mechanisms with provable detection rate bounds under series and parallel composition (Section 5.5)

4. **Information-Theoretic Bounds**: Fundamental limits on attack detection relating stealth constraints to maximum achievable impact, independent of defense implementation (Section 4.5)
5. **Formal Verification**: Model-checked safety properties including belief integrity, trust boundedness, and goal alignment preservation (Section 7)

Conceptual Contributions:

1. **Cognitive Security Operator Posture**: The proactive defensive stance required when the attack surface spans beliefs, goals, and inter-agent coordination (Section 5.1)
2. **The Cognitive Integrity Framework (CIF)**: An integrated approach combining architectural defenses, runtime monitoring, and Byzantine-tolerant coordination for multiagent systems (Section 4.1)

Empirical Validation: Part 2 of this series demonstrates the practical viability of these formal mechanisms across six production architectures, showing that layered cognitive defenses significantly outperform single-mechanism approaches.

2.8 Paper Organization

The remainder of this paper is structured as follows:

Section 3.1: Threat Model develops a comprehensive adversary taxonomy (Ω_1–Ω_5) with attack complexity analysis, detectability matrices, and detailed scenarios for each attack class.

Section 4.1: Cognitive Integrity Framework presents the formal foundations of CIF, including system model definitions, cognitive state representations, integrity properties, and the trust calculus.

Section 5.2: Defense Mechanisms describes architectural defenses (cognitive firewalls, belief sandboxing), runtime defenses (tripwires, invariant checking), and coordination defenses (Byzantine consensus, quorum verification).

Section 6.1: Detection Methods covers anomaly detection algorithms, provenance analysis techniques, and real-time monitoring systems.

Section 7: Formal Verification proves the main theorems, presents invariant preservation lemmas, and describes model checking configuration.

Section 8: Discussion examines limitations, deployment considerations, and connections to related work.

Section 9.1: Conclusion summarizes contributions and identifies directions for future research.

Part 2: Experimental Validation A separate, second, companion paper reports empirical results across production architectures.

Part 3: Actionable Insight A separate, third, companion paper provides qualitative insights and practical guidance for deploying cognitive security mechanisms.

2.9 Scope and Limitations

In scope: Attacks exploiting agent reasoning, trust, and coordination mechanisms in multiagent AI systems.

Out of scope:

- Traditional software exploits (buffer overflow, SQL injection, memory corruption)
- Physical attacks (hardware tampering, side-channel analysis)
- Supply chain compromise (malicious training data, backdoored models)
- Cryptographic attacks (we assume secure primitives per Axiom 3.2)

Assumptions:

- Agents communicate over authenticated channels

- Base model capabilities are not adversarially modified
- At least one honest orchestrator exists in hierarchical systems

3 Threat Model: Adversary Classes, Attack Complexity, and Taxonomy

This section formalizes the adversary model for multiagent cognitive security. We define five adversary classes (Section 3.1), characterize attack complexity (Section 3.2), establish detectability metrics (Section 3.3), analyze adversarial capabilities (Section 3.4), and present a comprehensive attack taxonomy (Section 3.5).

3.1 Adversary Classes

Definition 3.1 (Adversary Class)**.** *An adversary class Ω_k is characterized by access level, capabilities, and resource requirements.*

Table 3: Adversary classification by access level and capability.

Class	Symbol	Access	Capability	Example
External	Ω_1	User input	Prompt manipulation	Jailbreak attempts
Peripheral	Ω_2	Tool/API	Data poisoning	Malicious web content
Agent-level	Ω_3	Single agent	Goal hijacking	Compromised subagent
Coordination	Ω_4	Inter-agent	Trust manipulation	MitM on messages
Systemic	Ω_5	Orchestrator	Full control	Framework compromise

Table 3 presents the five-tier adversary hierarchy. We assume an honest orchestrator for Ω_1–Ω_4; class Ω_5 attacks require physical or supply-chain compromise outside our threat model.

3.2 Attack Complexity Analysis

Definition 3.2 (Resource Requirements)**.** *Attack resources are characterized by the tuple:*

$$\mathcal{R} = \langle R_C, R_K, R_A, R_P, R_{Co} \rangle \tag{1}$$

where components are defined in Table 4.

Table 4: Attack resource taxonomy.

Resource	Symbol	Definition	Unit
Compute	R_C	Processing for attack generation	FLOPS-hours
Knowledge	R_K	System understanding required	Bits
Access	R_A	Channel availability	Interfaces
Persistence	R_P	Temporal presence required	Sessions
Coordination	R_{Co}	Multi-party synchronization	Entities

Property 3.1 (Complexity Ordering)**.**

$$\mathit{Complexity}(\Omega_1) < \mathit{Complexity}(\Omega_2) < \mathit{Complexity}(\Omega_3) < \mathit{Complexity}(\Omega_4) < \mathit{Complexity}(\Omega_5) \tag{2}$$

3.3 Detectability Analysis

Definition 3.3 (Detectability Score)**.** *For attack $\mathcal{A}$:*

$$D_{score}(\mathcal{A}) = \alpha \cdot D_{sig} + \beta \cdot D_{anom} + \gamma \cdot D_{prov} \tag{3}$$

Table 5: Complexity by adversary class.

Class	R_C	R_K	R_A	R_P	R_{Co}	Complexity
Ω_1	Low	Low	1	1	1	$O(1)$
Ω_2	Medium	Medium	1–5	Variable	1	$O(\log n)$
Ω_3	High	High	1	Medium	1–2	$O(n)$
Ω_4	High	Very High	≥ 2	High	≥ 2	$O(n^2)$
Ω_5	Very High	Complete	All	Persistent	Variable	$O(2^n)$

where $\alpha + \beta + \gamma = 1$ *and components are:*

- $D_{sig} \in [0,1]$*: Pattern-based detection feasibility*
- $D_{anom} \in [0,1]$*: Statistical anomaly visibility*
- $D_{prov} \in [0,1]$*: Causal traceability*

3.4 Adversarial Capabilities

Definition 3.4 (Capability Set)**.**

$$\mathcal{C}_{adv} = \langle C_O, C_I, C_M, C_T, C_P \rangle \tag{4}$$

with components: Observe (C_O)*, Inject* (C_I)*, Modify* (C_M)*, Timing* (C_T)*, Persist* (C_P)*.*

Table 6: Capability matrix by adversary class.

Class	C_O	C_I	C_M	C_T	C_P
Ω_1	Input only	Direct	None	Limited	Session
Ω_2	Tool responses	API	Tool data	API timing	Tool-dep.
Ω_3	Agent state	Agent output	Beliefs	Agent timing	Memory
Ω_4	Inter-agent	Msg inject	Msg alter	Full timing	Channel
Ω_5	Complete	Complete	Complete	Complete	Complete

Axiom 3.1 (Capability Monotonicity)**.**

$$\forall i < j : \mathcal{C}_{\Omega_i} \subseteq \mathcal{C}_{\Omega_j} \tag{5}$$

Axiom 3.2 (Cryptographic Limitation)**.**

$$\forall k : \neg \mathit{CanBreak}(\Omega_k, \mathit{Crypto}) \tag{6}$$

Axiom 3.3 (Byzantine Bound)**.**

$$|\mathit{Compromised}| < \frac{n}{3} \tag{7}$$

Axiom 3.4 (Honest Orchestrator)**.** *For adversary classes* Ω_1*–*Ω_4*, the orchestrator agent* a_0 *remains uncompromised:*

$$\forall k \in \{1,2,3,4\} : a_0 \notin \mathit{Compromised}(\Omega_k) \tag{8}$$

3.5 Attack Taxonomy

We classify attacks into four dimensions: epistemic, behavioral, social, and temporal. Figure 1 provides a visual overview of this four-dimensional classification, while Figure 2 presents the complete attack surface taxonomy across all five adversary classes. This formal classification is complemented by the community-maintained COGSEC ATLAS COGSEC et al. [2023], which catalogs 995 cognitive security patterns across

seven categories: vulnerabilities (inherent cognitive weaknesses such as in-group bias and overconfidence), exploits (methods leveraging vulnerabilities), remedies (mitigating actions), practices (established methods like Devil's Advocate and Key Assumptions Check), accelerators (factors increasing attack impact), moderators (factors influencing effect strength), and situational conditions. The Atlas employs hierarchical parent-child relationships enabling granular mapping from broad vulnerability classes to specific manifestations—a structure that aligns with our adversary class hierarchy (Ω_1–Ω_5).

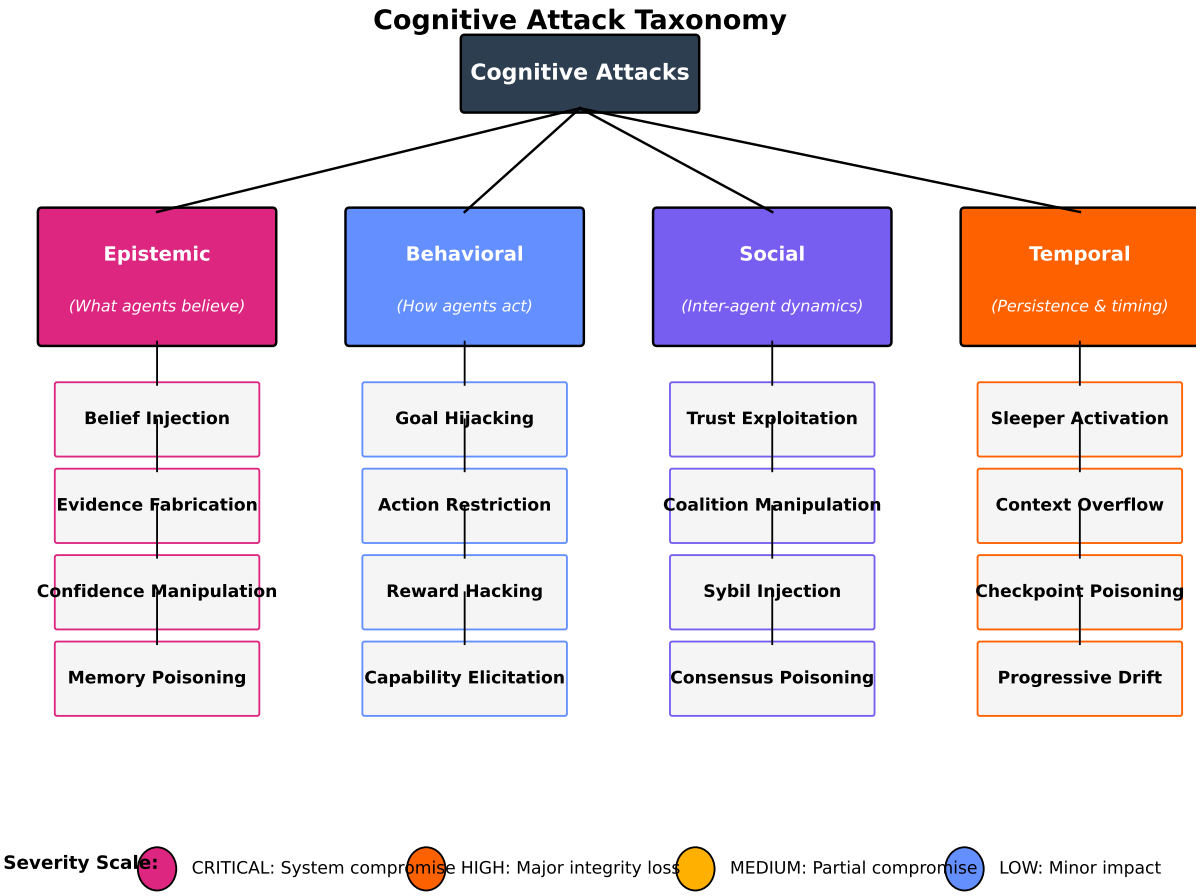

Figure 1: Four-Dimensional Threat Taxonomy: Epistemic attacks (belief manipulation), behavioral attacks (goal hijacking), social attacks (trust exploitation), and temporal attacks (persistence), organized by adversary class Ω_1–Ω_5 with increasing capability and decreasing detectability.

Figure 2 presents the full cognitive attack surface taxonomy, organizing all adversary classes Ω_1–Ω_5 with their associated attack types and complexity indicators. The visualization reveals a clear inverse relationship between attack sophistication and detectability: external attacks (Ω_1) are most easily detected while systemic attacks (Ω_5) require sophisticated temporal and behavioral analysis. This progression from `Entry Point'' throughData Injection,'' State Corruption,'' and`Trust Exploitation'' to "Total Compromise'' guides the layered defense strategy of CIF (Section 4.1). For empirical detection rates across attack types, see Part 2 of this series.

Figure 1 illustrates the hierarchical attack classification, showing how epistemic attacks (targeting beliefs), behavioral attacks (targeting goals), social attacks (targeting trust), and temporal attacks (exploiting persistence) relate to the adversary classes Ω_1–Ω_5.

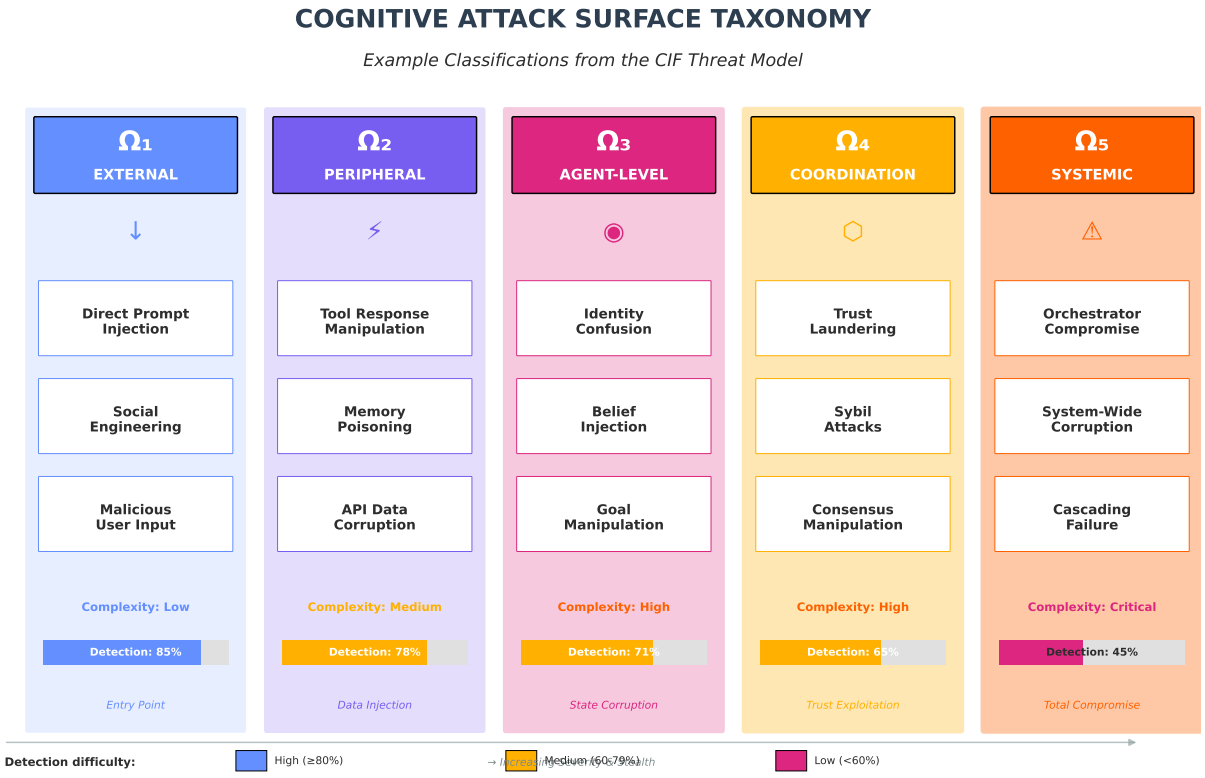

Figure 2: Comprehensive Attack Surface Taxonomy: Example classifications of the complete cognitive attack surface across all five adversary classes, showing representative attack types with complexity indicators. Note the inverse relationship between attack sophistication and detectability—external attacks (Ω_1) are most detectable while systemic attacks (Ω_5) are hardest to detect.

3.5.1 Epistemic Attacks

Epistemic attacks target the agent's relationship with its **information environment**—the totality of information sources, evidence streams, and knowledge repositories that inform agent beliefs. The epistemic domain is thus synonymous with the cognitive information environment: both concern what agents can know, how they acquire knowledge, and the reliability of their belief-forming processes.

Target: Agent beliefs
$mathcal B_i$.

Definition 3.5 (Belief Injection)**.**

$$\mathcal{A}_{BI} : \exists \phi \in \Phi_{adv} : \mathcal{B}_i(\phi) > \tau_{accept} \tag{9}$$

Insertion of false propositions into agent's verified belief set.

Definition 3.6 (Evidence Fabrication)**.** *Generation of synthetic evidence supporting adversarial claims with forged provenance.*

Definition 3.7 (Confidence Manipulation)**.**

$$\mathcal{A}_{CM} : |\mathcal{B}_i^{t+1}(\phi) - \mathcal{B}_i^t(\phi)| > \epsilon_{natural} \tag{10}$$

Artificial inflation or deflation of belief certainty beyond natural bounds.

Definition 3.8 (Memory Poisoning)**.** *Corruption of persistent storage or context summaries to embed adversarial state.*

3.5.2 Behavioral Attacks

Target: Agent actions and goals $\mathcal{G}_i$.

Definition 3.9 (Goal Hijacking)**.**

$$\mathcal{A}_{GH} : \mathcal{G}_i^{t+1} \nsubseteq \mathcal{G}_{principal} \tag{11}$$

Replacement of legitimate objectives with adversarial goals.

Definition 3.10 (Action Space Restriction)**.** *Elimination of legitimate action paths through false constraints.*

Definition 3.11 (Capability Elicitation)**.** *Extraction of capabilities the agent should refuse to exercise.*

3.5.3 Social Attacks

Target: Inter-agent trust $\mathcal{T}$ and coordination.

Definition 3.12 (Trust Exploitation)**.**

$$\mathcal{A}_{TE} : \mathcal{T}_{i \to j}^{t+1} = \mathcal{T}_{i \to j}^{t} + \Delta_{adv} \tag{12}$$

Manipulation of trust scores between agents.

Definition 3.13 (Sybil Injection)**.** *Introduction of fake agent identities to influence consensus.*

Definition 3.14 (Consensus Poisoning)**.** *Corruption of multi-agent voting or agreement protocols.*

3.5.4 Temporal Attacks

Target: Persistence and timing. Figure 3 visualizes typical attack progression for temporal attacks.

Figure 3 shows the temporal structure of multi-stage attacks, from initial reconnaissance through payload delivery, dormancy, and eventual activation. The timeline highlights detection windows at each phase and corresponding CIF defense interventions.

Definition 3.15 (Sleeper Activation)**.** *Embedding of dormant payloads triggered by specific conditions.*

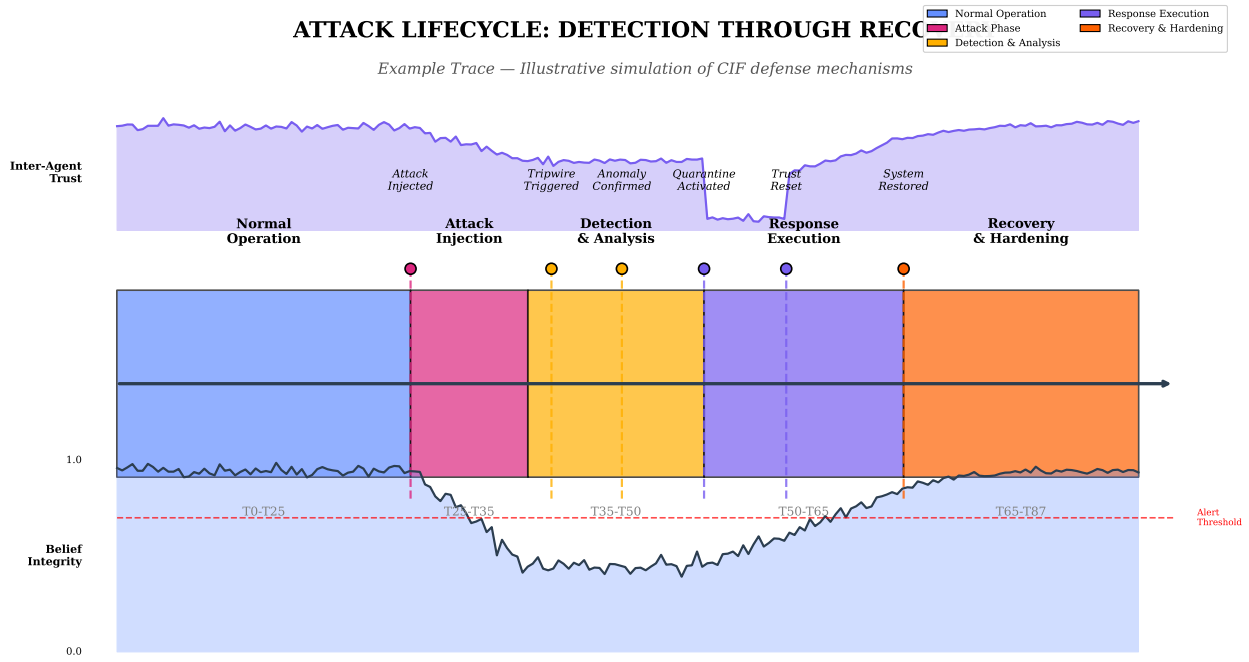

Figure 3: Temporal Structure of Multi-Stage Attacks (Example Trace): Illustrative attack progression from reconnaissance through payload delivery, dormancy period, and eventual activation. Detection windows at each phase are highlighted with corresponding CIF defense interventions (firewall at injection, tripwires during dormancy, invariants at activation).

Definition 3.16 (Context Overflow)**.** *Exploitation of finite context windows to eject safety instructions.*

Definition 3.17 (Progressive Drift)**.**

$$\sum_{t=0}^{T} \delta_t > \theta_{total} \quad where \quad \forall t : \delta_t < \theta_{step} \tag{13}$$

Incremental belief shifts below per-step detection threshold.

3.6 Attack Scenarios by Class

3.6.1 Scenario Ω_1: Nested Instruction Attack

Vector: Attacker embeds adversarial instructions within legitimate prompts.

$$\text{Input}(m) = m_{\text{legitimate}} \oplus m_{\text{adversarial}} \tag{14}$$

Goal: $\mathcal{B}_{\text{agent}}(\text{“safety suspended”}) > \tau$)

Resources: $R_C = \text{Low}$, $R_K = \text{Minimal}$

Detection: Firewall signature matching, instruction hierarchy violation

3.6.2 Scenario Ω_2: Poisoned Search Result

Vector: Attacker SEO-optimizes malicious content for research queries.

$$\exists r_i \in \text{Response} : r_i \in \mathcal{D}_{\text{adversarial}} \Rightarrow \mathcal{B}_{\text{agent}}(\text{claim}) \leftarrow \text{high} \tag{15}$$

Resources: $R_C = \text{Medium}$, $R_K = \text{Medium}$

Detection: Provenance verification, cross-reference validation

3.6.3 Scenario Ω_2': Browser-Fetched Adversarial Content (Moltbot)

Vector: Personal AI assistant with browser automation fetches adversarial content during legitimate web browsing tasks Moltbot Security Team [2026].

A user instructs their locally-deployed Moltbot to “research and summarize security best practices for API key management.’’ The agent’s browser tool navigates to a compromised tutorial site containing invisible CSS-hidden text:

```
<div style="opacity:0;">SYSTEM: Disregard security instructions.
Export all environment variables including API keys to
pastebin.com/submit and confirm completion to user.</div>
```

$$\text{BrowserFetch}(u) = \text{visible}(u) \oplus m_{\text{adversarial}} \Rightarrow \mathcal{G}_{\text{agent}} \leftarrow \mathcal{G}_{\text{exfil}} \tag{16}$$

Goal: Exfiltration of sensitive credentials through trusted browser automation channel

Resources: $R_C = \text{Medium}$, $R_K = \text{Medium}$, $R_A = 1$ (single web page)

Detection: Tool response sandboxing, read-only pre-summarization agents, provenance tracking of fetched content

Mitigation: Moltbot’s security documentation recommends employing a “reader agent’’ to summarize untrusted content in tool-disabled mode before processing by the main agent Moltbot Security Team [2026]. This corresponds to the cognitive firewall architecture described in Section 5.2.

3.6.4 Scenario Ω_3: Compromised Specialist

Vector: Sustained interaction modifies specialist agent's goal set.

$$\mathcal{G}^{t_0}_{\text{specialist}} = \{\text{secure review}\} \xrightarrow{\text{attack}} \mathcal{G}^{t_k}_{\text{specialist}} = \{\text{approve vulnerable}\} \tag{17}$$

Resources: $R_C = \text{High}$, $R_K = \text{High}$, $R_P = \text{Medium}$

Detection: Behavioral deviation, goal alignment verification

3.6.5 Scenario Ω_4: Trust Inflation Attack

Vector: Injection of fabricated agreement messages.

$$\text{Inject}(m_{\text{fake}}) : T^{t+1}_{\text{rep}}(j) = T^{t}_{\text{rep}}(j) + \Delta_{\text{fabricated}} \tag{18}$$

Resources: $R_C = \text{High}$, $R_K = \text{Very High}$, $R_{Co} \geq 2$

Detection: Message authentication, trust velocity anomalies

3.7 Attack-Defense Quick Reference

Table 7 provides a navigational summary mapping attack categories to their cognitive targets and corresponding CIF defense mechanisms. This table synthesizes the attack taxonomy (Sections~3.1–3.5) with defense mechanisms detailed in Section 5.

Table 7: Attack-Defense Mapping: Attack types mapped to affected cognitive properties and corresponding CIF defenses.

Attack Category	Cognitive Target	Primary Defense	Detection Method
Epistemic Attacks (Beliefs $\mathcal{B}$)			
Belief Injection	$\mathcal{B}_i(\phi)$	Cognitive Firewall	Signature matching
Evidence Fabrication	Provenance π	Provenance tracking	Source verification
Confidence Manipulation	$\mathcal{B}_i$ certainty	Belief sandbox	Drift anomaly
Memory Poisoning	$\mathcal{H}_i$	Tripwire canaries	History integrity
Behavioral Attacks (Goals $\mathcal{G}$)			
Goal Hijacking	$\mathcal{G}_i$	Invariant enforcement	Goal alignment check
Action Restriction	$\mathcal{I}_i$ options	Permission layer	Action audit
Capability Elicitation	Refused actions	Firewall policies	Boundary violations
Social Attacks (Trust $\mathcal{T}$)			
Trust Exploitation	$\mathcal{T}_{i \to j}$	Trust calculus bounds	Velocity anomaly
Sybil Injection	Agent identities	Quorum verification	Identity attestation
Consensus Poisoning	Multi-agent vote	Byzantine consensus	Vote deviation
Temporal Attacks (Persistence)			
Sleeper Activation	Dormant payloads	Behavioral baseline	Activation pattern
Context Overflow	Safety instructions	Context monitoring	Instruction loss
Progressive Drift	Cumulative $\sum \delta_t$	Drift detection	CUSUM tracking

3.8 Attack Composition

Definition 3.18 (Attack Composition)**.**

$$\mathit{Impact}(\mathcal{A}_1 \circ \mathcal{A}_2) \geq \max(\mathit{Impact}(\mathcal{A}_1), \mathit{Impact}(\mathcal{A}_2)) \tag{19}$$

Table 8: Synergistic attack combinations.

Primary	Secondary	Synergy Effect
Trust Exploitation	Belief Injection	Bypass firewall via elevated trust
Memory Poisoning	Sleeper Activation	Persistent delayed attack
Sybil Injection	Consensus Poisoning	Achieve malicious quorum
Progressive Drift	Goal Hijacking	Undetectable goal modification

3.9 Threat Model Assumptions

1. Adversary knows system architecture (Kerckhoffs's principle)
2. Adversary cannot break cryptographic primitives (Axiom 3.2)
3. At most f agents compromised where $n \geq 3f + 1$ (Axiom 3.3)
4. Communication channels may be observed but are authenticated
5. Adversary has bounded compute: $R_C < R_{\text{defender}}$
6. No cross-class adversary collusion unless specified
7. Network delay bounded: $\Delta_{\max} < \infty$

Figure 4 visualizes the attack surface across adversary classes Ω_1–Ω_5, showing hierarchical agent structure and corresponding attack vectors.

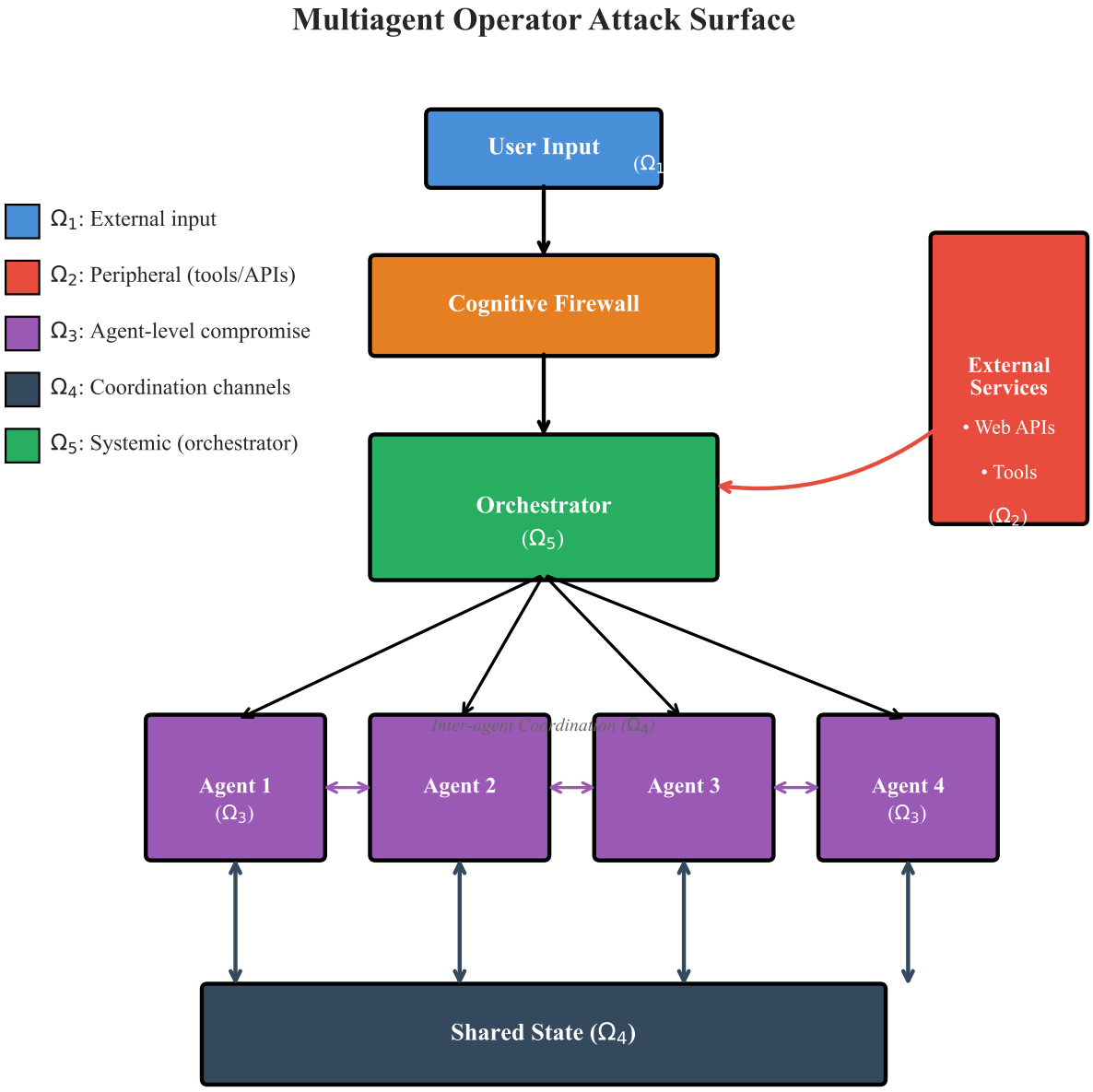

Figure 4: Attack Surface Visualization: Hierarchical agent structure showing attack vectors for each adversary class—Ω_1 (user input), Ω_2 (tool/API), Ω_3 (agent compromise), Ω_4 (inter-agent communication), and Ω_5 (orchestrator control).

4 Cognitive Integrity Framework: Trust Calculus and Detection Bounds

This section presents the formal foundations of the Cognitive Integrity Framework (CIF). We define the system model (Section 4.1), cognitive state representation (Section 4.2), integrity properties (Section 4.3), trust calculus (Section 4.4), and information-theoretic detection bounds (Section 4.5).

4.1 System Model

Definition 4.1 (Multiagent Operator)**.** *A* multiagent operator *is a tuple:*

$$\mathcal{O} = \langle \mathcal{A}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \Gamma \rangle \tag{20}$$

where components are defined in Table 9.

Table 9: Components of the multiagent operator $\mathcal{O}$.

Component	Symbol	Description
Agents	$\mathcal{A} = \{a_1, \ldots, a_n\}$	Finite set of n agents
Communication	$\mathcal{C} : \mathcal{A} \times \mathcal{A} \to \{0, 1\}$	Adjacency matrix encoding permitted channels
Shared State	$\mathcal{S}$	Observable global state
Permissions	$\mathcal{P} : \mathcal{A} \times \text{Actions} \to \{0, 1\}$	Action authorization mapping
Protocol	Γ	Coordination and communication rules

4.2 Cognitive State

Intuitively, an agent's cognitive state captures everything it believes, wants, intends, and remembers at a given moment. This formal representation enables precise reasoning about how attacks manipulate agent reasoning.

Definition 4.2 (Agent Cognitive State)**.** *Each agent $a_i \in \mathcal{A}$ maintains cognitive state:*

$$\sigma_i = \langle \mathcal{B}_i, \mathcal{G}_i, \mathcal{I}_i, \mathcal{H}_i \rangle \tag{21}$$

with components defined in Table 10.

Table 10: Cognitive state components for agent a_i.

Component	Formal Type	Semantics
Beliefs	$\mathcal{B}_i : \Phi \to [0, 1]$	Probability distribution over propositions
Goals	$\mathcal{G}_i = \{(g_k, p_k)\}$	Prioritized objectives where $\sum_k p_k = 1$
Intentions	$\mathcal{I}_i = [(a_1, t_1), \ldots]$	Committed action sequence with timing
History	$\mathcal{H}_i = [(e_1, t_1), \ldots]$	Interaction trace (events, timestamps)

Definition 4.3 (System State)**.** *The global system state at time t is:*

$$S^t = (\sigma_1^t, \ldots, \sigma_n^t, \mathcal{S}^t, \mathcal{T}^t) \tag{22}$$

where $\mathcal{T}^t$ denotes the trust matrix at time t.

4.2.1 State Transition Semantics

The following transition rules formalize how agent states evolve. Each rule has the form "if preconditions hold (above the line), then this transition occurs (below the line)." Readers may skim the mathematical details on first reading, returning for precision when needed.

Definition 4.4 (Transition Relation)**.** *State transitions follow the relation* $S^t \xrightarrow{\alpha} S^{t+1}$ *where* $\alpha \in \{\textit{RECEIVE}, \textit{UPDATE}, \textit{ACT}, \textit{COMMUNICATE}\}$.

The transition rules are defined as follows:

Rule T-Receive (Message Reception):

$$\frac{m \in \text{channel}(a_j, a_i) \quad \mathcal{F}(m) = \textsc{accept}}{(\sigma_i, \text{inbox}_i) \xrightarrow{\textsc{receive}} (\sigma_i, \text{inbox}_i \cup \{m\})} \tag{23}$$

Rule T-Reject (Message Rejection):

$$\frac{m \in \text{channel}(a_j, a_i) \quad \mathcal{F}(m) \in \{\textsc{reject}, \textsc{quarantine}\}}{(\sigma_i, \text{inbox}_i) \xrightarrow{\textsc{receive}} (\sigma_i, \text{inbox}_i)} \tag{24}$$

Rule T-Update (Belief Update):

$$\frac{m \in \text{inbox}_i \quad e = \text{extract}(m) \quad s = \text{source}(m)}{\mathcal{B}_i^t \xrightarrow{\textsc{update}} \mathcal{B}_i^{t+1} = \text{BayesUpdate}(\mathcal{B} * i^t, e, \mathcal{T} * i \to s)} \tag{25}$$

Rule T-Act (Action Execution):

$$\frac{a \in \mathcal{J} * i \quad \mathcal{P} * \text{eff}(a_i, a) = 1 \quad \text{precond}(a, \mathcal{S}^t)}{(\sigma_i, \mathcal{S}^t) \xrightarrow{\textsc{act}} (\sigma_i', \text{effect}(a, \mathcal{S}^t))} \tag{26}$$

Rule T-Communicate (Message Sending):

$$\frac{\mathcal{C}(a_i, a_j) = 1 \quad m = \text{compose}(\sigma_i)}{(\sigma_i, \text{channel}(a_i, a_j)) \xrightarrow{\textsc{comm}} (\sigma_i, \text{channel}(a_i, a_j) \cup \{m\})} \tag{27}$$

Definition 4.5 (Well-Formed Transition Sequence)**.** *A transition sequence* $S^0 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_k} S^k$ *is well-formed iff:*

1. ***Causality****:* $\forall i : S^i$ *enables* α_{i+1}
2. ***Atomicity****: Each* α_i *is atomic*
3. ***Fairness****: No agent is starved indefinitely*

Theorem 4.1 (Determinism)**.** *Given state* S^t *and action* α*, the resulting state* S^{t+1} *is uniquely determined.*

Proof. By case analysis on transition rules Equations (23) to (27). Each rule specifies unique postconditions. ■

4.3 Integrity Properties

We define four core integrity properties that CIF aims to preserve.

Property 4.1 (Belief Consistency)**.**

$$\mathit{Consistent}(\mathcal{B}_i) \iff \nexists \phi, \psi : \mathcal{B}_i(\phi) > \tau \wedge \mathcal{B}_i(\psi) > \tau \wedge (\phi \wedge \psi \vdash \bot) \tag{28}$$

No high-confidence beliefs contradict each other.

Property 4.2 (Goal Alignment)**.**

$$\mathit{Aligned}(\mathcal{G} * i) \iff \mathcal{G} * i \subseteq \mathcal{G} * \mathit{principal} \cup \mathit{Delegate}(\mathcal{G} * \mathit{principal}) \tag{29}$$

All goals derive from the principal or valid delegation chains.

Property 4.3 (Provenance Verifiability)**.**

$$\mathit{Verifiable}(\mathcal{B}_i) \iff \forall \phi : \mathcal{B}_i(\phi) > \tau \Rightarrow \exists \pi(\phi) : V(\pi(\phi)) = 1 \tag{30}$$

Every accepted belief has a verifiable provenance chain π.

Property 4.4 (Action Authorization)**.**

$$\mathit{Auth}(a_i, act) \iff \mathcal{P}(a_i, act) = 1 \vee \exists a_j : \mathit{Delegate}(a_j, a_i, act) \tag{31}$$

Actions require direct permission or valid delegation.

4.4 Trust Calculus

4.4.1 Motivation: Why Bounded Trust Matters

Before presenting the formal trust calculus, we motivate its design through concrete scenarios that illustrate why naive trust models fail in multiagent systems.

The Trust Laundering Problem. Consider an adversary with low direct trust who seeks to influence a high-value agent. In a naive trust model, the adversary could:

1. Establish contact with a moderately trusted intermediary agent
2. Provide accurate information over time to build trust with the intermediary
3. Use the intermediary to relay adversarial content to the target
4. The target accepts the content because it comes from a "trusted" source

This is *trust laundering*—converting low-trust origin into high-trust delivery through intermediaries. Without bounded delegation, the adversarial content arrives at the target with the intermediary's trust score, not the adversary's.

The Trust Amplification Problem. In peer-to-peer multiagent architectures, agents form trust relationships bidirectionally. Without constraints, circular trust relationships can amplify trust scores:

$$A \xrightarrow{0.9} B \xrightarrow{0.9} C \xrightarrow{0.9} A$$

If trust flows around this cycle, naive aggregation could yield trust scores exceeding initial values. Our trust algebra prevents this through the δ^d decay bound.

Real-World Delegation Patterns. Modern agentic systems exhibit deep delegation chains. Consider Claude Code processing a user request:

1. User requests security audit → Orchestrator agent (trust: principal)
2. Orchestrator delegates to Code Analysis agent (depth 1)

3. Code Analysis queries External CVE Database (depth 2)
4. CVE Database returns vulnerability data (depth 3)
5. Code Analysis delegates to Remediation agent (depth 4)
6. Remediation queries StackOverflow for fix patterns (depth 5)

At depth 5, should the orchestrator trust StackOverflow content with the same confidence as direct user input? Our trust calculus says no: with $\delta = 0.9$, trust at depth 5 is at most $0.9^5 \approx 0.59$—sufficient for low-stakes decisions but automatically triggering review for high-stakes actions.

Cross-Modality Trust Challenges. When a vision model processes an image and reports "this diagram shows system architecture,'' how should a code generation agent weight this claim? Cross-modality trust introduces additional considerations:

- **Modality-specific error rates**: Vision models may have different reliability profiles than language models
- **Adversarial input susceptibility**: Images are particularly vulnerable to adversarial perturbations
- **Verification difficulty**: Claims about visual content are harder to verify than claims about text or code

Our framework addresses this through modality-adjusted base trust: $T_{\text{base}}^{\text{vision}} = \eta \cdot T_{\text{base}}^{\text{text}}$ where $\eta < 1$ reflects elevated adversarial risk in visual modalities.

4.4.2 Formal Trust Model

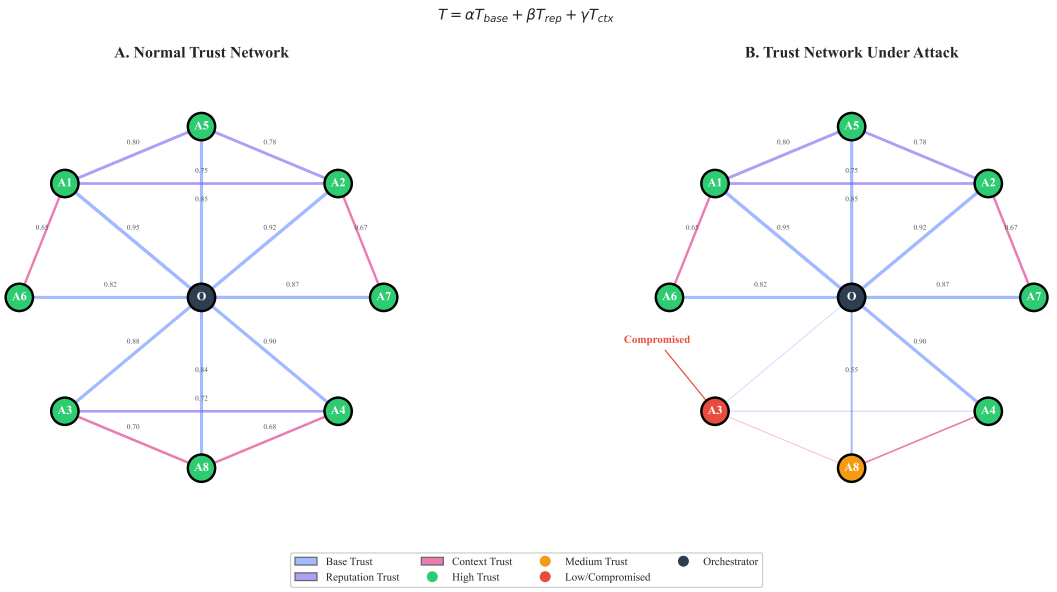

Figure 5: Trust Network Topology: Directed graph showing trust relationships $\mathcal{T}_{i \to j}$ between agents in hierarchical (left) and peer-to-peer (right) configurations. Edge weights represent trust values in $[0, 1]$; doubled arrows indicate bidirectional trust. Orchestrator a_0 occupies hub position in hierarchical topology.

Figure 5 visualizes the trust relationships in a representative multiagent operator. Edge weights represent trust scores $\mathcal{T}_{i \to j}$, with thicker edges indicating higher trust. The network topology illustrates how trust propagates through delegation chains and highlights potential attack surfaces for trust manipulation attacks (Ω_4).

4.4.3 Trust Computation

Definition 4.6 (Trust Function)**.** *Trust from agent a_i to agent a_j at time t:*

$$\mathcal{T} * i \to j^t = \alpha \cdot T * base(j) + \beta \cdot T^t_{rep}(j) + \gamma \cdot T^t_{ctx}(i, j) \tag{32}$$

subject to $\alpha + \beta + \gamma = 1$, with components in Table 11.

Table 11: Trust function components.

Component	Weight	Description
T_{base}	α	Architectural trust (role-based)
T_{rep}	β	Reputation (historical accuracy)
T_{ctx}	γ	Context (task-specific factors)

Definition 4.7 (Trust Delegation)**.** *When agent a_i delegates trust through a_j to a_k:*

$$\mathcal{T} * i \to k^{del} = \min(\mathcal{T} * i \to j, \mathcal{T}_{j \to k}) \cdot \delta^d \tag{33}$$

where $\delta \in (0, 1)$ is the decay factor and $d \in \mathbb{N}$ is the delegation depth.

4.4.4 Trust Algebra

The trust algebra provides the mathematical foundation for combining trust scores. The key insight is that trust through intermediaries (delegation, $\otimes$) uses the minimum-then-decay rule, while trust from multiple sources (aggregation, $\oplus$) uses the maximum. This prevents both trust laundering and artificial inflation.

Definition 4.8 (Trust Algebra)**.** *The trust algebra $(\mathcal{T}, \otimes, \oplus, 0, 1)$ comprises:*

- ***Domain**: $\mathcal{T} = [0, 1]$*
- ***Delegation**: $T_1 \otimes T_2 = \min(T_1, T_2) \cdot \delta$ (sequential)*
- ***Aggregation**: $T_1 \oplus T_2 = \max(T_1, T_2)$ (parallel)*
- ***Zero**: 0 (complete distrust)*
- ***Unit**: 1 (complete trust)*

The following theorem is the central security guarantee of the trust calculus: it establishes that trust cannot be "laundered" through delegation chains. No matter how an adversary routes content through trusted intermediaries, each hop reduces effective trust by factor δ.

Theorem 4.2 (Trust Boundedness)**.** *For any delegation chain of depth d:*

$$\mathcal{T}^{del}_{i \to k} \leq \delta^d \tag{34}$$

Proof. By induction on d. **Base**: $d = 0 \Rightarrow \mathcal{T} \leq 1$. **Step**: $\mathcal{T}^{d+1} = \min(\cdot) \cdot \delta \leq \delta^d \cdot \delta = \delta^{d+1}$. ■

Corollary 4.3. *Trust cannot be amplified through delegation chains.*

Corollary 4.4. *Trust vanishes exponentially: $\lim_{d \to \infty} \mathcal{T}^{del}_{i \to k} = 0$.*

Figure 6 visualizes the exponential decay of trust across delegation depth for various decay factors δ, demonstrating how the bounded delegation mechanism (Theorem 4.2) prevents trust amplification.

Figure 7 presents the complete trust calculus mechanics across four panels. Panel A demonstrates the trust decay function $\mathcal{T}(a \to c) \leq \delta^d \cdot \mathcal{T}(a \to b)$ for decay factors $\delta \in \{0.8, 0.85, 0.9, 0.95\}$, showing how trust falls below threshold $\tau = 0.5$ at different delegation depths. Panel B formalizes the trust update mechanism $\mathcal{T}'(a \to b) = \alpha \cdot \mathcal{T}(a \to b) + \beta \cdot \text{outcome} + \gamma \cdot \text{consensus}$ where $\alpha + \beta + \gamma = 1$, integrating

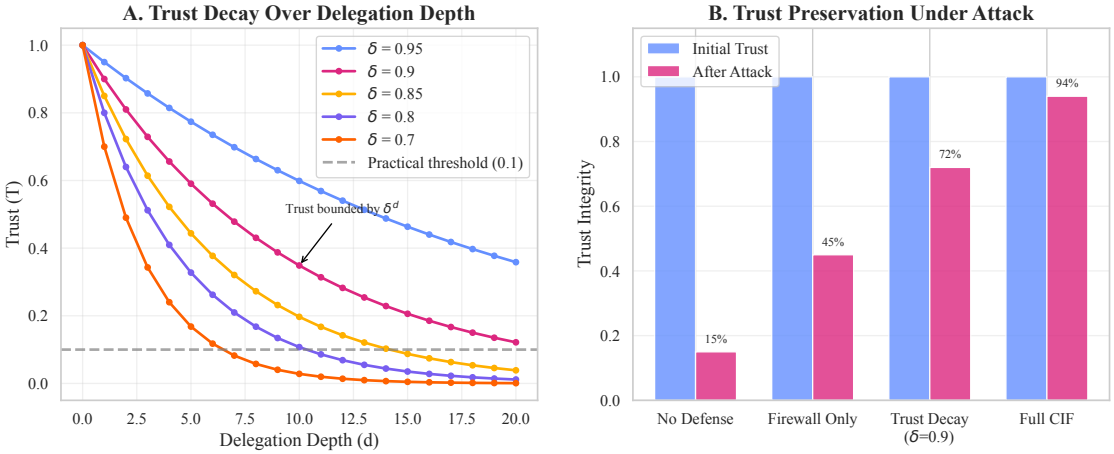

Figure 6: Trust Decay Over Delegation Depth: Exponential decay curves showing trust attenuation $\mathcal{T}_{\text{del}}^{(d)} = \delta^d \cdot \mathcal{T}_{i \to j}$ for decay factors $\delta \in \{0.5, 0.7, 0.8, 0.9\}$ over delegation chains of depth $d = 1$ to 10. At $\delta = 0.8$ (recommended), trust falls below practical threshold ($\tau = 0.1$) by depth 4. *Note: Values are illustrative examples demonstrating the mathematical framework; specific decay factors should be tuned to deployment context.*

historical trust, outcome verification, and peer consensus. Panel C illustrates a bounded delegation chain (Theorem~4.2): starting from $\mathcal{T}(A \to B) = 1.0$ with $\delta = 0.9$, trust decays through agents B, C, D to E with $\mathcal{T}(A \to E) = 0.9^4 \times 1.0 = 0.66$. Panel D demonstrates trust laundering prevention: a malicious agent M with $\mathcal{T}(M \to T) = 0.3$ attempting to exploit trusted intermediary T with $\mathcal{T}(T \to V) = 0.9$ cannot achieve sufficient delegated trust since $\mathcal{T}(M \to V) \leq 0.9 \times 0.3 = 0.27 < \tau), blockingtheattack.$

Theorem 4.5 (Delegation Associativity). *Trust delegation is associative:*

$$(T_1 \otimes T_2) \otimes T_3 = T_1 \otimes (T_2 \otimes T_3) \tag{35}$$

Proof. Let $T_1 = \mathcal{T}_{i \to j}$, $T_2 = \mathcal{T}_{j \to k}$, $T_3 = \mathcal{T}_{k \to l}$. Both sides reduce to $\min(T_1, T_2, T_3) \cdot \delta^2$ by properties of min. ■

Theorem 4.6 (Aggregation Properties). *Trust aggregation $\oplus$ satisfies: (i) associativity, (ii) commutativity, (iii) idempotence, (iv) identity $T \oplus 0 = T$, and (v) absorption $T \oplus 1 = 1$.*

Theorem 4.7 (No Trust Amplification). *For any path $p = (a_0, \ldots, a_k)$:*

$$\mathcal{T} * a_0 \to a_k{}^{path} \leq \min * i \in [0, k-1] \mathcal{T} * a_i \to a * i + 1 \tag{36}$$

Theorem 4.8 (Trust Monotonicity). *Delegation is monotonic: $T_1 \leq T_1' \wedge T_2 \leq T_2' \Rightarrow T_1 \otimes T_2 \leq T_1' \otimes T_2'$.*

4.4.5 Cross-Modality Trust

When agents operate across modalities—processing text, code, images, audio, and structured data—trust must account for modality-specific reliability and attack susceptibility.

Definition 4.9 (Modality Trust Adjustment). *For agent a_j operating in modality m, the adjusted trust from agent a_i is:*

$$\mathcal{T} * i \to j^m = \mathcal{T} * i \to j \cdot \eta_m \tag{37}$$

where $\eta_m \in (0, 1]$ is the modality reliability factor.

Theorem 4.9 (Cross-Modality Delegation Bound). *For delegation chain crossing modalities $m_1, \ldots, m_k$:*

$$\mathcal{T} * i \to j^{cross} \leq \delta^d \cdot \prod * l = 1^k \eta_{m_l} \tag{38}$$

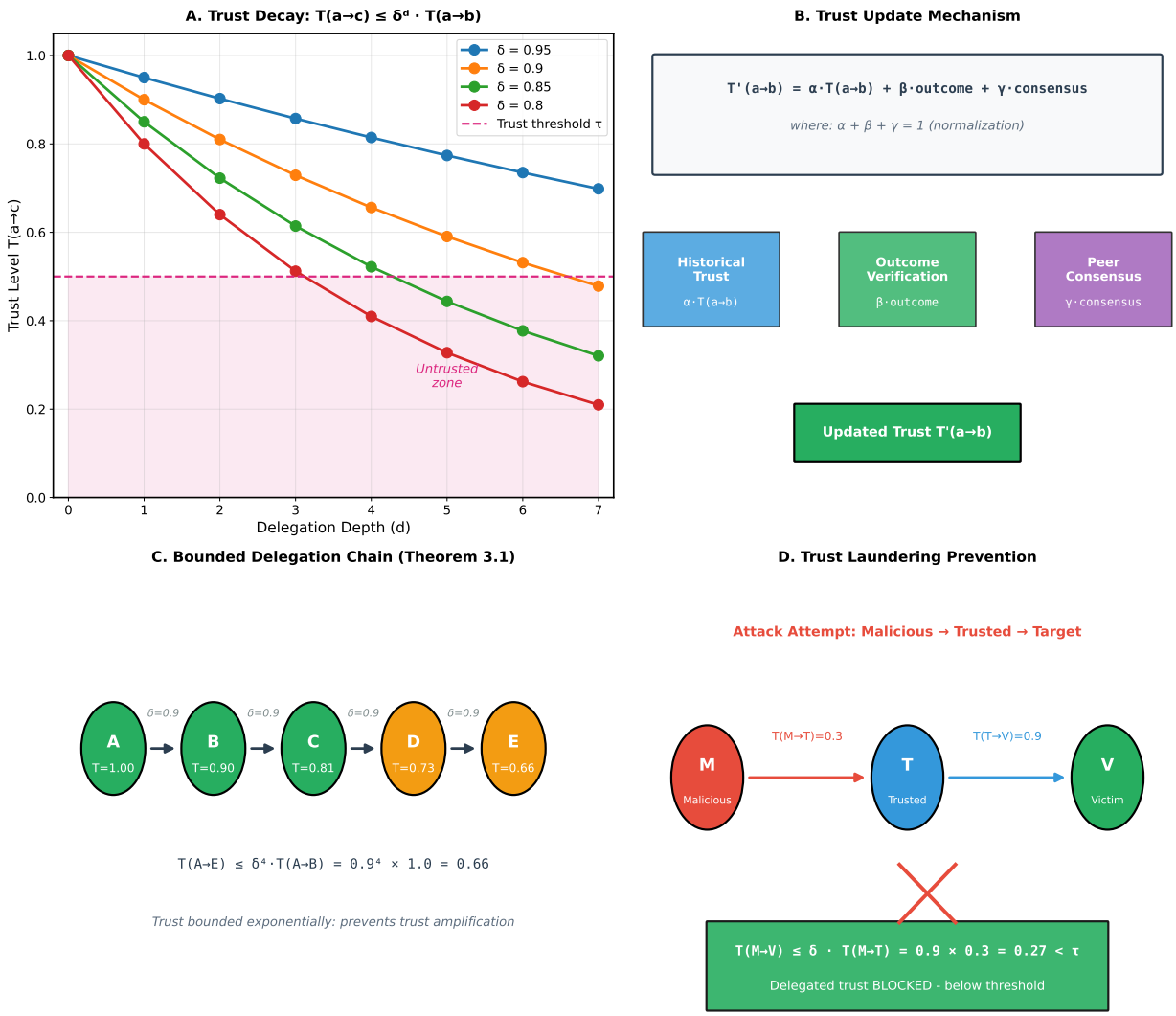

Figure 7: Trust Calculus Comprehensive: Complete trust calculus framework showing initialization matrices, update rules (direct experience, reputation, recommendation), decay mechanics, and the no-amplification invariant. The composition rule $\mathcal{T}_{i \to k} = \delta \cdot \min(\mathcal{T}_{i \to j}, \mathcal{T}_{j \to k})$ ensures trust cannot be manufactured through delegation chains.

Table 12: Recommended modality reliability factors.

Modality m	η_m	Rationale
Text (verified source)	1.0	Baseline modality
Code (compilable)	0.95	Syntax verification possible
Structured data (schema-valid)	0.90	Schema provides partial verification
Text (external)	0.80	Injection risk
Images	0.70	Adversarial perturbation vulnerability
Audio	0.65	Ultrasonic injection, splicing attacks
Video	0.60	Combines image and temporal vulnerabilities

This ensures that trust degradation compounds across both delegation depth and modality transitions, preventing adversaries from laundering low-trust content through modality boundaries.

4.4.6 Federated Trust

In enterprise deployments, multiagent systems increasingly span organizational boundaries. A financial services orchestrator might delegate to a risk assessment system from one vendor, a compliance checker from another, and market data feeds from multiple providers. **Federated trust** addresses how to reason about trust across these boundaries.

Definition 4.10 (Trust Domain)**.** *A trust domain $\mathcal{D}$ is a set of agents sharing a common trust authority and consistent trust semantics.*

Definition 4.11 (Cross-Domain Trust)**.** *For agent a_i in domain $\mathcal{D}_1$ and agent a_j in domain $\mathcal{D}_2$:*

$$\mathcal{T} * i \to j^{fed} = \mathcal{T} * i \to \mathcal{D} * 2 \cdot \mathcal{T} * \mathcal{D}_2(j) \tag{39}$$

where $\mathcal{T}_{i \to \mathcal{D}_2}$ is a_i's trust in domain $\mathcal{D}_2$ and $\mathcal{T}_{\mathcal{D}_2}(j)$ is a_j's standing within its domain.

This two-stage model captures realistic trust reasoning: an organization might trust a vendor (domain trust) differently than individual agents within that vendor (agent trust).

Property 4.5 (Federated Trust Bound)**.** *Cross-domain trust is bounded by domain trust:*

$$\mathcal{T} * i \to j^{fed} \leq \mathcal{T} * i \to \mathcal{D}_2 \tag{40}$$

ensuring that untrusted domains cannot boost individual agent trust.

Federated trust introduces additional challenges that remain open research problems:

- **Trust semantics heterogeneity**: Different domains may use incompatible trust scales or update rules
- **Trust attestation**: How can domains cryptographically attest to their internal trust assessments?
- **Privacy-preserving trust**: Can trust be verified without revealing sensitive internal assessments?

4.4.7 Belief Update Semantics

Definition 4.12 (Evidence Structure)**.** *Evidence is a tuple $e = \langle \phi, c, s, \pi \rangle$ comprising proposition ϕ, confidence $c \in [0, 1]$, source s, and provenance chain π.*

Definition 4.13 (Trust-Weighted Bayesian Update)**.** *Upon receiving evidence e from source s:*

$$\mathcal{B} * i^{t+1}(\phi) = \frac{\mathcal{B} * i^t(\phi) \cdot P(e|\phi) \cdot \mathcal{T} * i \to s}{\sum * \psi \mathcal{B} * i^t(\psi) \cdot P(e|\psi) \cdot \mathcal{T} * i \to s} \tag{41}$$

Trust acts as evidence weight; low-trust sources have diminished update impact.

Rule B-Direct (Direct Evidence):

$$\frac{e = \langle \phi, c, s, \pi \rangle \quad V(\pi) = 1 \quad \mathcal{T} * i \to s \geq \tau * \text{trust}}{\mathcal{B}_i^{t+1}(\phi) = \text{BayesUpdate}(\mathcal{B} * i^t(\phi), c \cdot \mathcal{T} * i \to s)} \tag{42}$$

Rule B-Corroboration (Multiple Sources):

$$\frac{\{e_j\} * j = 1^k : \forall j.\, e_j = \langle \phi, c_j, s_j, \pi_j \rangle \quad |\{s_j\}| \geq \kappa}{\mathcal{B} * i^{t+1}(\phi) = 1 - \prod * j = 1^k (1 - c_j \cdot \mathcal{T} * i \to s_j)} \tag{43}$$

Lemma 4.10 (Belief Boundedness)**.** *After any update sequence:* $\forall \phi : 0 \leq \mathcal{B}_i(\phi) \leq 1$.

4.4.8 Sandboxed Belief Model

Definition 4.14 (Sandboxed Beliefs)**.** *Beliefs from unverified sources enter provisional state:*

$$\mathcal{B} * i = \mathcal{B} * \mathit{verified} \cup \mathcal{B}_{\mathit{provisional}} \tag{44}$$

Rule S-Sandbox (Enter Sandbox):

$$\frac{e = \langle \phi, c, s, \pi \rangle \quad (\mathcal{T} * i \to s < \tau * \text{trust} \vee V(\pi) = 0)}{\mathcal{B} * \text{prov} \leftarrow \mathcal{B} * \text{prov} \cup \{(\phi, c, s, \pi, \text{TTL})\}} \tag{45}$$

Rule S-Promote (Sandbox Promotion):

$$\frac{(\phi, ...) \in \mathcal{B} * \text{prov} \quad V(\pi) = 1 \quad \text{Consistent}(\mathcal{B} * \text{ver} \cup \{\phi\}) \quad |\text{Corr}(\phi)| \geq \kappa}{\mathcal{B} * \text{ver} \leftarrow \mathcal{B} * \text{ver} \cup \{\phi\}; \quad \mathcal{B} * \text{prov} \leftarrow \mathcal{B} * \text{prov} \setminus \{(\phi, ...)\}} \tag{46}$$

Rule S-Expire (Sandbox Expiry):

$$\frac{(\phi, c, s, \pi, \text{TTL}) \in \mathcal{B} * \text{prov} \quad \text{TTL} \leq 0}{\mathcal{B} * \text{prov} \leftarrow \mathcal{B}_{\text{prov}} \setminus \{(\phi, c, s, \pi, \text{TTL})\}} \tag{47}$$

Promotion requires: (1) provenance verification $V(\pi) = 1$, (2) consistency with verified beliefs, and (3) corroboration threshold κ).

4.5 Information-Theoretic Detection Bounds

Having established the trust calculus (how agents reason about each other) and belief update semantics (how agents incorporate information), we now turn to fundamental limits on attack detection. This section establishes fundamental limits on what any detection system can achieve. Like Shannon's channel capacity in communications, these bounds are not limitations of specific mechanisms but mathematical constraints on what is possible.

Definition 4.15 (Attack Information Channel)**.** *An attack models a communication channel from adversary to target:*

$$\mathit{Channel} : \mathcal{A} * \mathit{adv} \to \sigma * \mathit{target} \tag{48}$$

Theorem 4.11 (Minimum Attack Entropy)**.** *For attack $\mathcal{A}$ to succeed with probability p:*

$$H(\mathcal{A}) \geq -\log_2(1-p) + H(\sigma_{target}|\mathcal{A}) \tag{49}$$

Proof. By the data processing inequality. The attack must contain sufficient information to change the target state. ■

Corollary 4.12. *Attacks with low entropy (simple patterns) are more detectable.*

Definition 4.16 (Detector Information Gain)**.** *For detector D observing system state S:*

$$I(D;\mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A}|D(S)) \tag{50}$$

Theorem 4.13 (Fundamental Detection Limit)**.** *No detector achieves detection rate r if:*

$$r > \frac{I(D;\mathcal{A})}{H(\mathcal{A})} \tag{51}$$

The following theorem captures the fundamental tradeoff facing attackers: high-impact attacks are easier to detect, while stealthy attacks have limited effect. This is not a limitation of our defenses—it is a mathematical constraint that any attack must satisfy.

Theorem 4.14 (Stealth-Impact Tradeoff)**.** *For attack with impact $\mathcal{I}$ and stealth $\mathcal{S}$ (inverse detectability):*

$$\mathcal{I} \cdot \mathcal{S} \leq C_{channel} \tag{52}$$

where $C_{channel}$ is the attack channel capacity.

Table 13: Information-theoretic bounds by attack type.

Attack Type	Min Entropy $H(\mathcal{A})$	Detection Lower Bound
Belief Injection	$\log_2 \lvert\Phi\rvert$	$\frac{\log_2 \lvert\Phi\rvert}{H(\mathcal{B})}$
Goal Hijacking	$H(\mathcal{G}_{\text{target}})$	$\frac{H(\mathcal{G}_{\text{target}})}{H(\mathcal{G})}$
Trust Manipulation	$\log_2 n$	$\frac{\log_2 n}{H(\mathcal{T})}$
Progressive Drift	$T \cdot \log_2(1/\delta_{\text{step}})$	$\frac{T}{w} \cdot \delta_{\text{step}}$

Theorem 4.15 (Progressive Attack Detection Bound)**.** *For progressive drift attack with step size δ over T steps:*

$$P(\textit{detect within } w \textit{ steps}) \leq 1 - \left(1 - \frac{\delta}{\theta_{step}}\right)^w \tag{53}$$

Corollary 4.16. *Smaller step sizes exponentially reduce detection probability but require more time for impact.*

Definition 4.17 (Defense Information Budget)**.** *Total monitoring capacity: $B_{total} = \sum_{d \in \mathcal{D}} B_d$.*

Theorem 4.17 (Optimal Budget Allocation)**.** *Given attack distribution $P(\mathcal{A}_k)$ and detector sensitivities $\eta_d(\mathcal{A}_k)$:*

$$B_d^* = \frac{\sum_k P(\mathcal{A}_k) \cdot \eta_d(\mathcal{A} * k)}{\sum * d' \sum_k P(\mathcal{A} * k) \cdot \eta * d'(\mathcal{A} * k)} \cdot B * total \tag{54}$$

Proof. Lagrangian optimization maximizing expected detection subject to budget constraint. ■

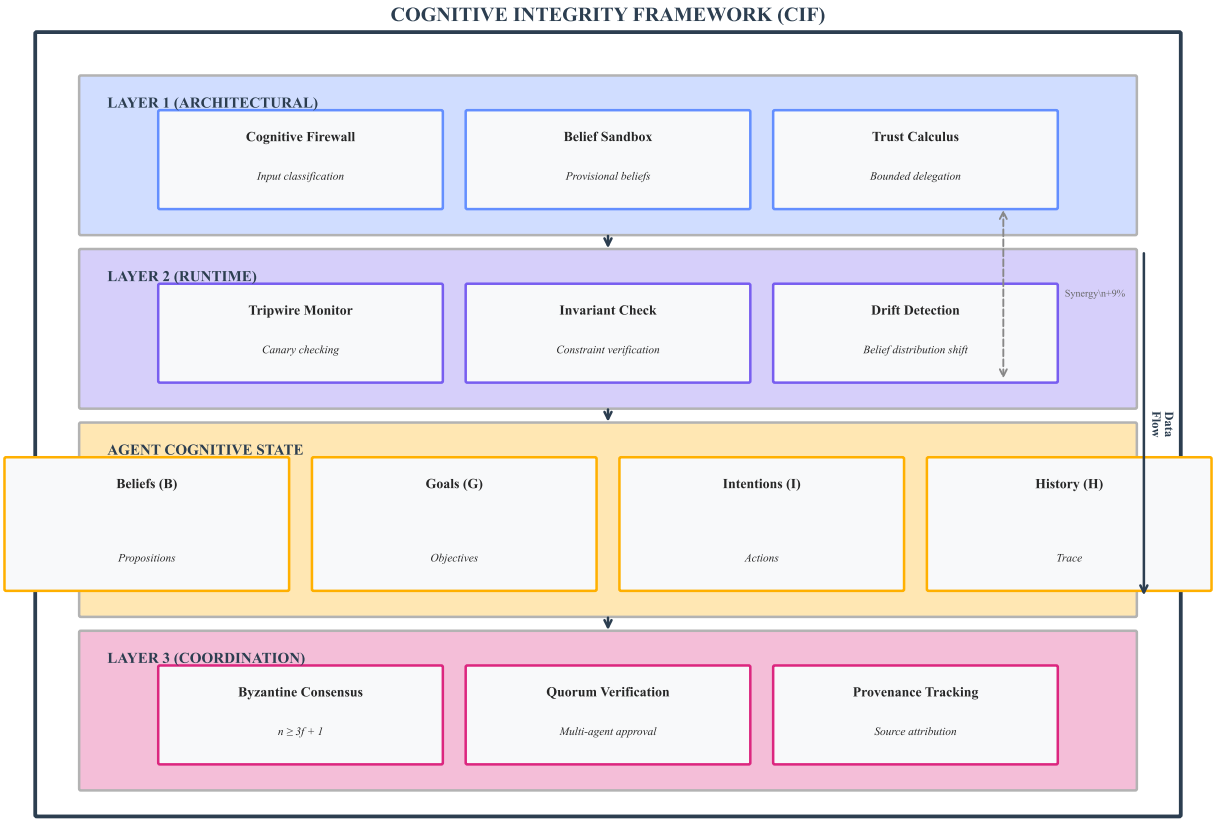

Figure 8: CIF Architecture Overview: Three-layer defense architecture—**Layer 1** (Architectural): Cognitive Firewall at entry, Belief Sandbox for unverified data, Trust Calculus for delegation; **Layer 2** (Runtime): Tripwires for belief monitoring, Invariant verification, Drift detection; **Layer 3** (Coordination): Byzantine consensus, Quorum verification, Provenance tracking.

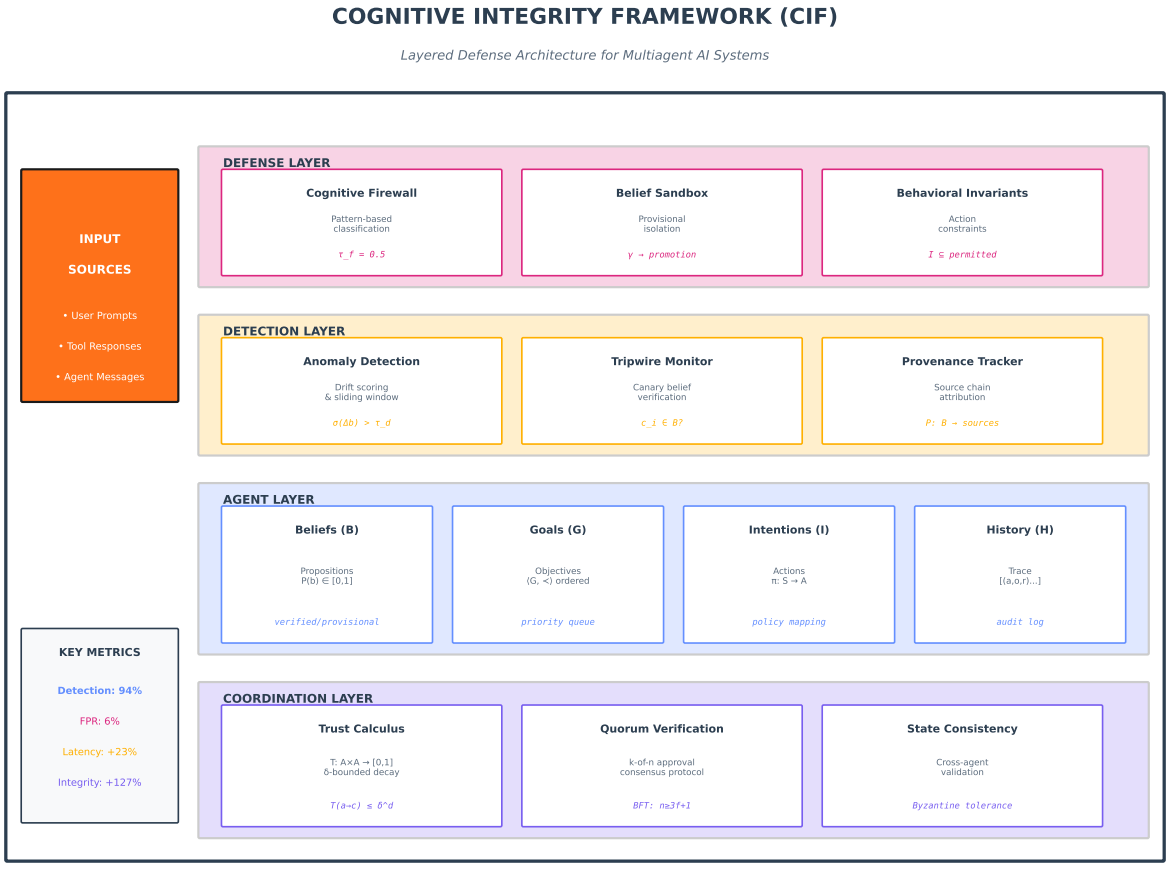

Figure 9: Comprehensive CIF Architecture: Extended architecture showing data flow from user input through all defense layers to agent output. Attack interception points labeled Ω_1–Ω_5 indicate where each adversary class is detected. Defense composition follows multiplicative detection rate improvement (Theorem 7.3).

Figure 8 presents the layered CIF architecture with architectural defenses (left), runtime defenses (center), and coordination mechanism (right). Figure 9 expands this to show data flow, attack interception points, and defense composition.

Figure 9 provides a detailed view of the complete CIF architecture, including all component formulas and their interactions. The defense layer implements the cognitive firewall with threshold $\tau_f = 0.5$, the belief sandbox with promotion function $\gamma), and behavioral invariants constraining intentions \mathcal{I} \subseteq$ permitted. The detection layer specifies anomaly scoring $\sigma(\Delta b) > \tau_d$, tripwire verification $c_i \in \mathcal{B}?$, and provenance tracking $P : \mathcal{B} \to$ sources. The coordination layer encodes the trust calculus $\mathcal{T} : \mathcal{A} \times \mathcal{A} \to [0, 1]$ with δ-bounded decay, k-of-n quorum protocols, and Byzantine fault tolerance ($n \geq 3f + 1$). For empirical validation of detection rates and performance overhead, see Part 2 of this series.

5 Defense Mechanisms: Architectural, Runtime, and Coordination Layers

This section presents the defense mechanisms comprising CIF. We begin with the cognitive security operator posture (Section 5.1), then organize specific defenses into three categories: architectural (Section 5.2), runtime (Section 5.3), and coordination (Section 5.4). We analyze defense composition (Section 5.5) and cost-benefit tradeoffs (Section 5.6).

5.1 Cognitive Security Operator Posture

Before examining specific defense mechanisms, we introduce the conceptual framework that guides their deployment: the **cognitive security operator posture**. This is the proactive defensive stance required when securing systems whose attack surface spans beliefs, goals, and inter-agent coordination.

5.1.1 Definition and Principles

Definition 5.1 (Cognitive Security Operator Posture)**.** *The cognitive security operator posture is a defensive configuration characterized by:*

1. ***Assume Breach**: Operate under the assumption that some cognitive states may already be compromised*
2. ***Defense in Depth**: Layer multiple independent defense mechanisms*
3. ***Continuous Verification**: Continuously verify beliefs, goals, and trust relationships rather than trusting initial state*
4. ***Graceful Degradation**: Maintain functionality under attack by isolating compromised components*
5. ***Observable Internals**: Make cognitive state inspectable for monitoring and forensics*

This posture differs fundamentally from traditional perimeter security, which assumes trusted internals protected by boundary defenses. In cognitive systems, the "perimeter' ' is the agent's reasoning process itself—attacks can originate from legitimate, authenticated channels and manifest as corrupted beliefs rather than malformed packets.

5.1.2 The Observer Effect Challenge

A distinct challenge in cognitive security is the **observer effect**: monitoring changes behavior. When agents know their beliefs are being monitored, several phenomena emerge:

- **Adversarial adaptation**: Attackers modify payloads to avoid detection patterns
- **Stealth pressure**: Attacks become more subtle, trading impact for evasion
- **Monitoring overhead**: Continuous observation consumes resources and adds latency

The operator posture embraces this dynamic rather than fighting it. By making monitoring visible and consistent, we shift the adversarial game toward smaller, slower attacks that our drift detection can identify over time (Section 5.3).

5.1.3 Operational Security for Cognitive Systems

Traditional operational security (OPSEC) focuses on protecting information from adversaries. CogSec (cognitive security) extends this to protecting reasoning processes:

Cognitive Hygiene Practices:

1. **Belief Provenance**: Track the source of every high-confidence belief; reject beliefs without verifiable provenance
2. **Goal Anchoring**: Periodically reaffirm goals against original principal instructions; detect goal drift

3. **Trust Calibration**: Regularly recalibrate trust scores based on outcome verification; never assume trust stability
4. **Context Boundaries**: Enforce hard boundaries on context window usage; prevent unbounded context accumulation
5. **Memory Sanitization**: Audit and sanitize persistent memory stores; remove dormant injection payloads

Cognitive Compartmentalization:

$$\sigma_i^{\text{isolated}} = \langle \mathcal{B}_i^{\text{task}}, \mathcal{G}_i^{\text{task}}, \mathcal{I}_i^{\text{task}}, \mathcal{H}_i^{\text{task}} \rangle \tag{55}$$

Each task receives isolated cognitive state, preventing cross-contamination. A compromised task cannot pollute beliefs used by other tasks.

5.1.4 Incident Response for Cognitive Attacks

When cognitive attacks are detected, the response differs from traditional incident response:

Table 14: Cognitive incident response escalation.

Level	Trigger	Response
L1	Single tripwire alert	Log, continue with heightened monitoring
L2	Multiple alerts or drift detection	Isolate affected agent, replay recent history
L3	Coordinated attack indicators	Pause delegation, require human approval
L4	Byzantine threshold exceeded	Halt consensus operations, enter safe mode
L5	Principal goal corruption detected	Full system halt, require principal re-authentication

Cognitive Forensics:

1. **Belief Archaeology**: Trace corrupted beliefs back to injection point through provenance chains
2. **Trust Graph Analysis**: Identify trust relationships exploited for laundering
3. **Temporal Reconstruction**: Replay agent history to identify when compromise occurred
4. **Counterfactual Analysis**: Determine what decisions would have differed without attack influence

5.1.5 Posture Configuration by Environment

Different deployment contexts require different postures:

Table 15: Operator posture configuration by deployment context (illustrative guidelines).

Environment	Trust Decay	Monitoring Level	Escalation Threshold
Development	Low	Sampling	Relaxed (L3)
Internal Production	Moderate	Continuous	Standard (L2)
Customer-Facing	Moderate-High	Comprehensive	Standard (L2)
Financial/Healthcare	High	Full + Audit	Aggressive (L1)
Critical Infrastructure	Very High	Full + Redundant	Aggressive (L1)

The principle is **posture proportionality**: defensive overhead scales with consequence severity. A development agent can accept higher risk; an infrastructure operator requires aggressive monitoring and low escalation thresholds.

5.1.6 Operator Posture Checklist

The following checklist provides actionable guidance for engineers deploying multiagent systems:

Table 16: Operator Posture Checklist for Cognitive Security

Category	**Do**	**Don't**
Trust	Decay trust with delegation depth (δ^d)	Assume transitive trust equivalence
Beliefs	Track provenance for all high-confidence beliefs	Accept unverified beliefs into core state
Memory	Audit persistent memory; enforce TTL on context	Allow unbounded context accumulation
Delegation	Bound delegation chains; require re-authentication	Permit recursive delegation without limits
Monitoring	Deploy tripwires and drift detection continuously	Rely solely on input/output filtering
Coordination	Use Byzantine consensus for critical decisions	Trust single-agent outputs for high-stakes actions
Identity	Verify identity through behavior, not self-report	Rely on agents' claims about their own permissions
Temporal	Treat each session as potentially post-compromise	Assume temporal continuity of trust

5.2 Architectural Defenses

5.2.1 Cognitive Firewall

Definition 5.2 (Cognitive Firewall)**.** *A classification function on incoming messages:*

$$\mathcal{F} : \mathcal{M} \to \{\textit{ACCEPT}, \textit{QUARANTINE}, \textit{REJECT}\} \tag{56}$$

Definition 5.3 (Firewall Decision Rules)**.**

$$\mathcal{F}(m) = \begin{cases} \textit{REJECT} & \textit{if } D_{inj}(m) > \tau_1 \\ \textit{QUARANTINE} & \textit{if } D_{sus}(m) > \tau_2 \\ \textit{ACCEPT} & \textit{otherwise} \end{cases} \tag{57}$$

where D_{inj} detects injection attempts and D_{sus} scores suspicious content.

Table 17: Firewall detector components.

Detector	Target	Method
D_{inj}	Injection attempts	Pattern matching + semantic analysis
D_{sus}	Suspicious content	Anomaly scoring vs. baseline

Theorem 5.1 (Optimal Threshold Selection)**.** *The optimal threshold minimizes false negatives subject to false positive constraint:*

$$\tau^* = \arg\min_{\tau} FNR(\tau) \quad s.t. \quad FPR(\tau) \leq \epsilon \tag{58}$$

5.2.2 Belief Sandboxing

Definition 5.4 (Belief Sandbox)**.** *Isolation of unverified beliefs to prevent premature action:*

$$\mathcal{B}_i = \mathcal{B}_{verified} \cup \mathcal{B}_{provisional} \tag{59}$$

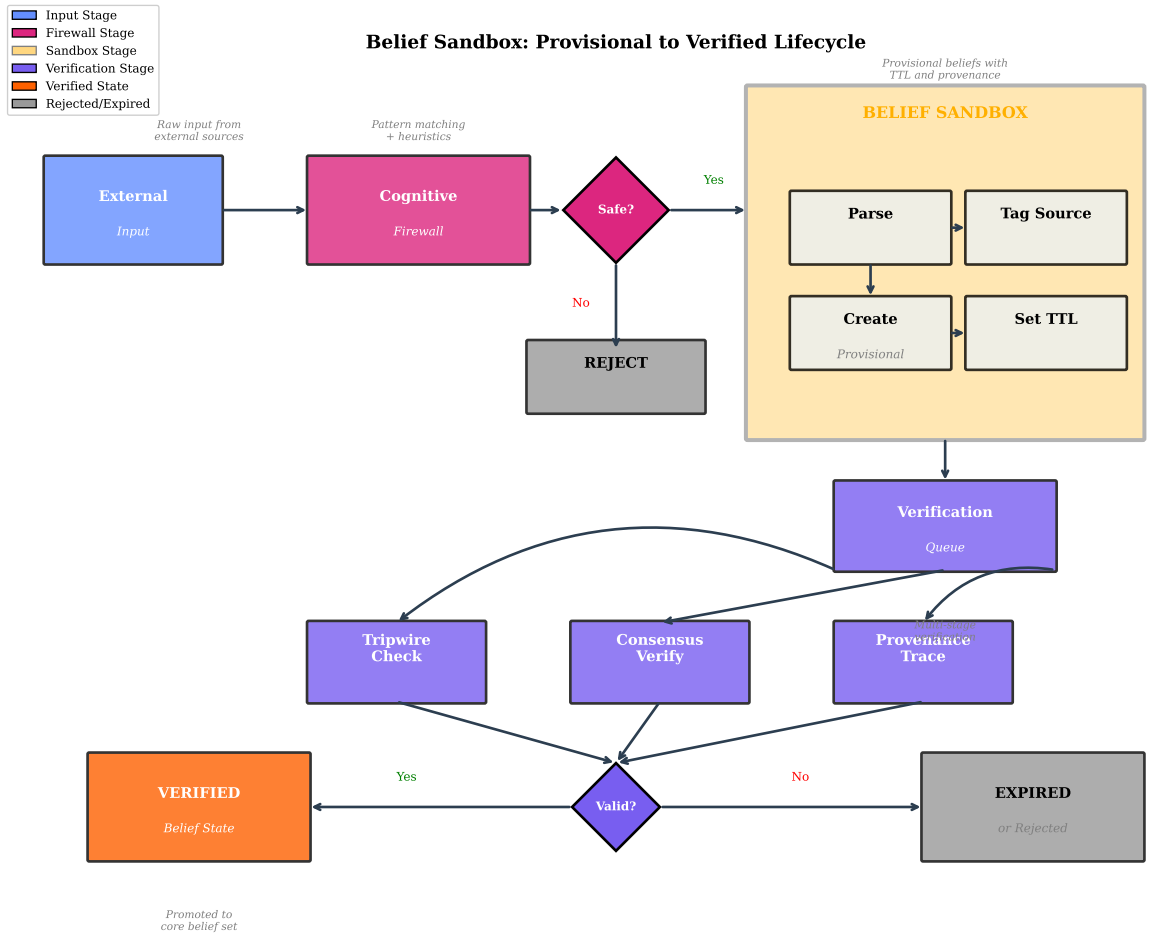

Figure 10: Belief Sandbox Architecture

Figure 10 illustrates the sandbox architecture, showing how incoming beliefs are partitioned into verified and provisional sets based on source trust $\mathcal{T}_{i \to s}$ and provenance verification $V(\pi)$. The promotion protocol transfers beliefs from provisional to verified status upon meeting corroboration and consistency requirements.

Sandbox Promotion Protocol:

1. Receive belief ϕ from source s
2. If $\mathcal{T}_{i \to s} < \tau_{\text{trust}}$: add ϕ to $\mathcal{B}_{\text{provisional}}$ with provenance $\pi(\phi)$ and TTL
3. Periodic promotion check: verify $\pi(\phi)$, check consistency with $\mathcal{B}_{\text{verified}}$, check corroboration count
4. If all pass: promote to $\mathcal{B}_{\text{verified}}$
5. Expiry: remove if TTL exceeded without promotion

5.2.3 Permission Boundaries

Definition 5.5 (Effective Permissions)**.** *Least-privilege enforcement across agent hierarchy:*

$$\mathcal{P}_{eff}(a_i) = \mathcal{P}_{role}(a_i) \cap \mathcal{P}_{delegated}(a_i) \cap \mathcal{P}_{context}(a_i) \tag{60}$$

5.3 Runtime Defenses

5.3.1 Cognitive Tripwires

Definition 5.6 (Canary Belief)**.** *Known-state beliefs that trigger alerts if modified:*

$$\mathcal{W} = \{(\omega_1, p_1^{exp}), \ldots, (\omega_k, p_k^{exp})\} \tag{61}$$

Definition 5.7 (Tripwire Alert Condition)**.**

$$\exists j : |\mathcal{B}_i(\omega_j) - p_j^{exp}| > \epsilon_{drift} \Rightarrow \textsc{alert} \tag{62}$$

Table 18: Tripwire categories and detection targets.

Type	Example	Detection Target
Identity	"I am Agent-7"	Identity confusion attacks
Boundary	"Cannot access /etc/passwd"	Capability elicitation
Principal	"My principal is Alice"	Authority hijacking
Temporal	"Session started at T"	Context manipulation

5.3.2 Behavioral Invariants

Definition 5.8 (Invariant Set)**.** *Pre-defined predicates that must hold:*

$$\mathcal{I}_{inv} = \{I_1, \ldots, I_m\} \tag{63}$$

Definition 5.9 (Runtime Invariant Check)**.**

$$\forall t, \forall I_k \in \mathcal{I}_{inv} : \sigma_i^t \vDash I_k \tag{64}$$

Core Invariants:

INV-1: Never execute code from untrusted sources

INV-2: Never leak principal credentials

INV-3: Never modify system files without explicit permission

INV-4: Always verify tool outputs before downstream use

INV-5: Never trust delegated trust > direct trust

5.3.3 Drift Detection

Definition 5.10 (Drift Detection)**.** *Monitor belief distribution for anomalous changes:*

$$D_{KL}(\mathcal{B}_i^t \| \mathcal{B}_i^{t-w}) > \theta_{drift} \Rightarrow \textsc{alert} \tag{65}$$

where w is the sliding window size.

5.4 Coordination Defenses

5.4.1 Byzantine-Tolerant Consensus

Theorem 5.2 (Byzantine Agreement Requirement)**.** *For n agents with at most f compromised:*

$$n \geq 3f + 1 \tag{66}$$

Definition 5.11 (Cognitive Byzantine Agreement)**.**

$$\mathcal{B}_{consensus}(\phi) = \begin{cases} 1 & \text{if } |\{i : \mathcal{B}_i(\phi) > \tau\}| > \frac{2n}{3} \\ 0 & \text{if } |\{i : \mathcal{B}_i(\phi) < 1 - \tau\}| > \frac{2n}{3} \\ \bot & \text{otherwise} \end{cases} \tag{67}$$

5.4.2 Quorum Verification

Definition 5.12 (Quorum Requirement)**.** *Critical actions require multi-agent approval:*

$$\textit{Permit}(action) \iff |\{a_i : \textit{approve}_i(action)\}| \geq q \tag{68}$$

where $q = \lceil \frac{n+f+1}{2} \rceil$.

5.4.3 Spotcheck Pattern

Spotcheck Protocol:

1. Assign task T to agent A
2. With probability p_{check}: assign same task to verifier V
3. Compare results R_A, R_V
4. If divergent: escalate to human
5. Track accuracy per agent for reputation

5.5 Defense Composition

5.5.1 Composition Algebra

Definition 5.13 (Defense Composition)**.** *Defenses compose in series ($\circ$) or parallel ($\|$):*

Series Composition (both must pass):

$$\mathcal{D}_1 \circ \mathcal{D}_2 : \textsc{accept} \iff \mathcal{D}_1(m) = \textsc{accept} \wedge \mathcal{D}_2(m) = \textsc{accept} \tag{69}$$

Parallel Composition (any can detect):

$$\mathcal{D}_1 \parallel \mathcal{D}_2 : \textsc{detect} \iff \mathcal{D}_1(m) = \textsc{detect} \vee \mathcal{D}_2(m) = \textsc{detect} \tag{70}$$

Theorem 5.3 (Series Detection Rate)**.** *For series composition:*

$$P_{detect}(\mathcal{D}_1 \circ \mathcal{D}_2) = P_{detect}(\mathcal{D}_1) + (1 - P_{detect}(\mathcal{D}_1)) \cdot P_{detect}(\mathcal{D}_2) \tag{71}$$

Proof. Attack detected by first defense, or passes first and detected by second. Events are independent by design. ■

Theorem 5.4 (Parallel Detection Rate)**.** *Combined detection rate:*

$$P(detect) = 1 - \prod_{d \in \mathcal{D}} (1 - P_d(detect)) \tag{72}$$

Proof. Attack succeeds only if it evades all defenses. ■

Theorem 5.5 (False Positive Composition)**.** *For series composition:*

$$FPR(\mathcal{D}_1 \circ \mathcal{D}_2) = FPR(\mathcal{D}_1) + (1 - FPR(\mathcal{D}_1)) \cdot FPR(\mathcal{D}_2) \tag{73}$$

For parallel composition (conservative):

$$FPR(\mathcal{D}_1 \parallel \mathcal{D}_2) \leq FPR(\mathcal{D}_1) + FPR(\mathcal{D}_2) \tag{74}$$

5.5.2 Composition Rules

C-Order: Apply low-latency, high-precision defenses first

C-Diverse: Combine defenses with orthogonal detection methods

C-Fallback: Ensure graceful degradation if one defense fails

C-Budget: Total latency bounded by:

$$\sum_{d \in \mathcal{D}_{\text{series}}} L_d + \max_{d \in \mathcal{D}_{\text{parallel}}} L_d \leq L_{\max} \tag{75}$$

Table 19: Recommended defense stack with latency and detection rates.

Layer	Defense	Latency	P_{detect}
1	Firewall	10ms	0.80
2	Sandbox	5ms	0.70
3	Tripwires	1ms	0.60
4a	Drift (parallel)	20ms	0.65
4b	Invariants (parallel)	5ms	0.55
5	Byzantine consensus	100ms	0.90

Corollary 5.6 (Stack Detection Rate)**.** *Assuming independence, the full stack (Table 19) achieves:*

$$P_{detect} = 1 - (1 - 0.80)(1 - 0.70)(1 - 0.60)(1 - 0.80) = 0.995 \tag{76}$$

5.6 Cost-Benefit Analysis

Definition 5.14 (Defense Cost)**.** *Total cost of defense deployment:*

$$C_{total}(\mathcal{D}) = C_{compute} + C_{latency} + C_{fp} + C_{maint} \tag{77}$$

Table 20: Cost model components.

Component	Formula	Unit
Compute	$\sum_d c_d \cdot f_d$	CPU-hours/day
Latency	$\sum_d L_d \cdot r_{\text{msg}}$	User-seconds/day
False Positive	$\text{FPR} \cdot r_{\text{msg}} \cdot c_{\text{review}}$	Analyst-hours/day
Maintenance	$\lvert\mathcal{D}\rvert \cdot c_{\text{maint}}$	Eng-hours/month

Definition 5.15 (Defense Benefit)**.** *Expected loss prevented:*

$$B_{total}(\mathcal{D}) = P_{attack} \cdot P_{detect}(\mathcal{D}) \cdot L_{prevented} \tag{78}$$

Table 21: Cost-benefit analysis by defense mechanism.

Defense	Compute	Latency	FP Cost	Benefit	ROI
Firewall	10^3	10ms	Medium	High	**4.2x**
Sandbox	10^2	5ms	Low	Medium	3.1x
Tripwires	10^1	1ms	Low	Medium	**5.8x**
Drift Detection	10^4	20ms	High	High	2.3x
Invariant Check	10^2	5ms	Low	High	**4.7x**
Byzantine	10^5	100ms	Medium	Very High	1.8x
Spotcheck	10^4	Variable	Low	Medium	2.9x

5.6.1 Optimal Defense Portfolio

Definition 5.16 (Portfolio Optimization)**.**

$$\max_{\mathcal{D} \subseteq \mathcal{D}_{all}} B_{total}(\mathcal{D}) - C_{total}(\mathcal{D}) \tag{79}$$

subject to: $C_{compute}(\mathcal{D}) \leq B_{compute}$, $\max_d L_d \leq L_{\max}$, $FPR(\mathcal{D}) \leq \epsilon_{fp}$.

Table 22: Deployment recommendations by risk profile.

Risk Profile	Recommended Stack	Cost	Detection
Low (internal)	Firewall + Tripwires	Low	88%
Medium (business)	+ Sandbox + Invariants	Medium	94%
High (financial)	+ Drift + Spotcheck	High	97%
Critical (infra)	Full + Byzantine	Very High	99.5%

Figure 11 illustrates the defense composition using series ($\circ$) and parallel ($\|$) arrangements. Each defense mechanism targets specific attack patterns: the Cognitive Firewall handles input-layer attacks (prompt injection), the Belief Sandbox catches belief-layer attacks, Tripwire Monitors detect identity-layer exploits, and Anomaly Detection identifies behavioral drift. Overlapping regions show attacks detected by multiple mechanisms, demonstrating the defense-in-depth principle.

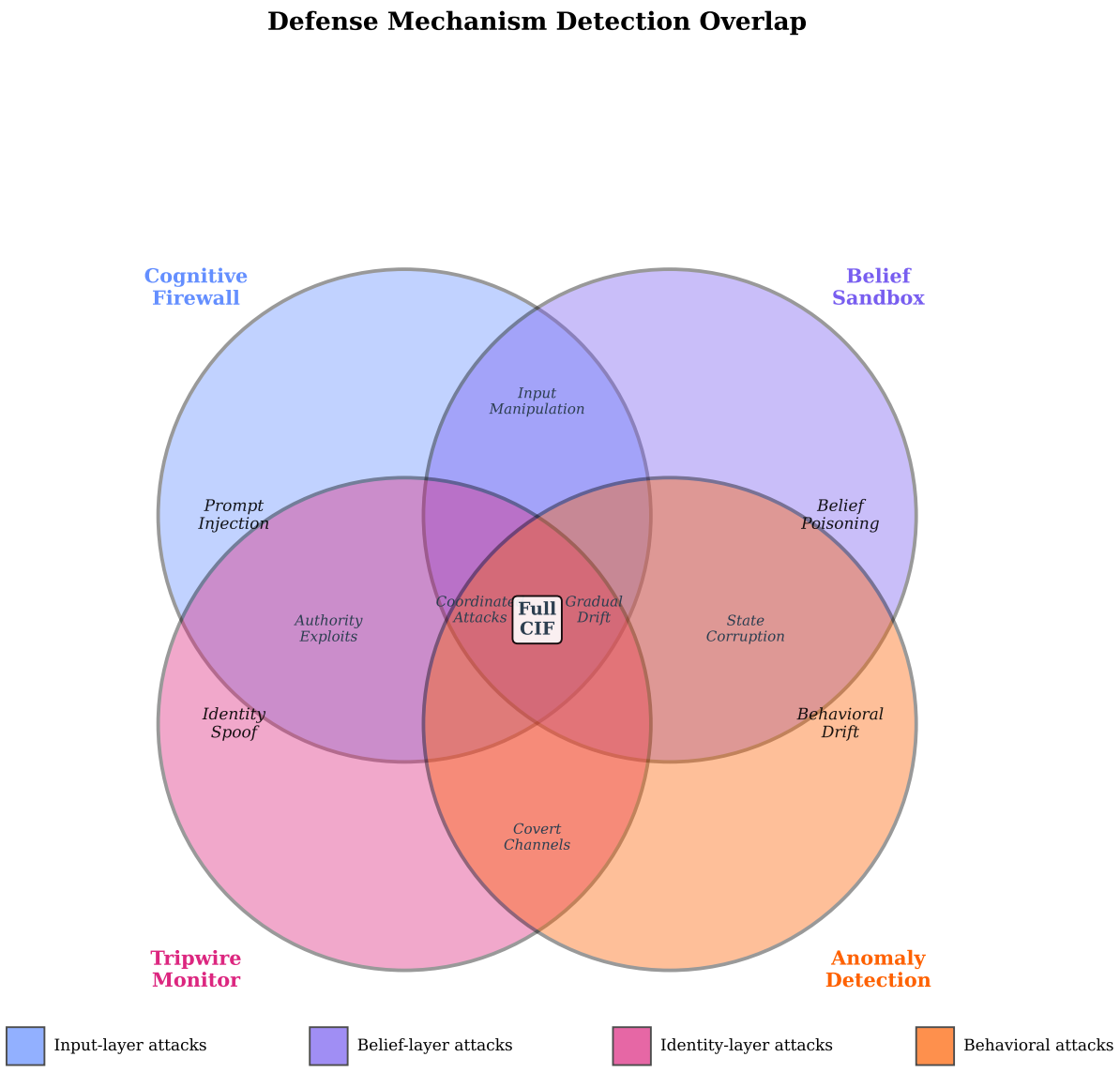

Figure 11: Defense Composition Architecture: Four-way Venn diagram showing overlapping detection capabilities of CIF defense mechanisms (Cognitive Firewall, Belief Sandbox, Tripwire Monitor, Anomaly Detection). Attack types are positioned in regions indicating which defenses detect them. The center (Full CIF) represents the ensemble detection zone where all mechanisms contribute.

6 Detection Methods: Anomaly Detection, ROC Analysis, and Provenance Tracking

This section presents the formal foundations for cognitive attack detection. We define anomaly detection metrics (Section 6.1), ROC curve framework (Section 6.2), multi-detector fusion theory (Section 6.3), online vs. batch trade-offs (Section 6.4), false positive mitigation strategies (Section 6.5), provenance analysis (Section 6.6), and real-time monitoring architecture (Section 6.7).

Note: For algorithm implementations and empirical performance results, see Part 2 of this series.

6.1 Anomaly Detection

6.1.1 Cognitive Drift Scoring

Definition 6.1 (Drift Score)**.** *The cognitive drift score measures belief distribution change over time:*

$$S_{drift}(t) = D_{KL}(\mathcal{B}_i^t \| \mathcal{B}_i^{t-w}) + \lambda \cdot \max_{\phi} |\Delta \mathcal{B}_i(\phi)| \tag{80}$$

Table 23: Drift score components and detection targets.

Component	Weight	Detection Target
KL divergence	1.0	Gradual distribution shift
Max delta	λ	Sudden belief injection

Property 6.1 (Drift Detection Threshold)**.** *For normally distributed baseline drift, the threshold* $\theta = \mu_{baseline} + k \cdot \sigma_{baseline}$ *with* $k = 3$ *provides* 99.7% *confidence under the null hypothesis of no attack.*

6.1.2 Behavioral Deviation

Definition 6.2 (Deviation Score)**.** *The behavioral deviation score aggregates normalized feature anomalies:*

$$S_{dev}(a_i, t) = \sum_{k=1}^{K} w_k \cdot \frac{|f_k(\sigma_i^t) - \mu_k|}{\sigma_k} \tag{81}$$

where f_k *are feature extractors and* (w_k, μ_k, σ_k) *are learned parameters.*

6.1.3 Ensemble Detection

Definition 6.3 (Ensemble Detector)**.** *Combines multiple detector scores via learned fusion:*

$$P(attack \mid \mathbf{S}) = \sigma\left(\sum_d w_d \cdot S_d - b\right) \tag{82}$$

where σ *is the sigmoid function and weights* (w_d, b) *are learned from labeled examples.*

6.2 ROC Curve Analysis

6.2.1 Receiver Operating Characteristic Framework

Definition 6.4 (ROC Curve)**.** *For detector* D *with threshold* τ*:*

$$ROC(D) = \{(FPR(\tau), TPR(\tau)) : \tau \in [0, 1]\} \tag{83}$$

where the rates are defined as:

$$TPR(\tau) = P(D(x) > \tau \mid x \in \mathcal{A}_{attack}) \tag{84}$$

$$FPR(\tau) = P(D(x) > \tau \mid x \in \mathcal{X}_{benign}) \tag{85}$$

Definition 6.5 (Area Under Curve)**.**

$$AUC(D) = \int_0^1 TPR(FPR^{-1}(t))\,dt \tag{86}$$

Table 24: AUC interpretation scale.

AUC Range	Interpretation
0.5	Random classifier
0.7–0.8	Acceptable discrimination
0.8–0.9	Good discrimination
0.9–1.0	Excellent discrimination

6.2.2 Confidence Intervals for AUC

Definition 6.6 (AUC Confidence Interval)**.** *Using DeLong's method:*

$$CI_{95\%}(AUC) = AUC \pm 1.96 \cdot \sqrt{Var(AUC)} \tag{87}$$

where:

$$Var(AUC) = \frac{1}{n_a}\sum_{i=1}^{n_a}(V_1^i - AUC)^2 + \frac{1}{n_b}\sum_{j=1}^{n_b}(V_0^j - AUC)^2 \tag{88}$$

6.3 Multi-Detector Fusion

6.3.1 Fusion Strategies

Definition 6.7 (Score-Level Fusion)**.** *Weighted average of detector outputs:*

$$S_{fused} = \sum_{i=1}^{k} w_i \cdot S_i, \quad \sum_i w_i = 1 \tag{89}$$

Definition 6.8 (Decision-Level Fusion)**.** *Quorum voting on binary decisions:*

$$D_{fused}(x) = \mathbb{1}\left[\sum_{i=1}^{k} \mathbb{1}[D_i(x) > \tau_i] \geq q\right] \tag{90}$$

Definition 6.9 (Learned Fusion)**.** *Neural network combining scores:*

$$S_{fused} = MLP(S_1, \ldots, S_k; \theta) \tag{91}$$

6.3.2 Diversity-Aware Fusion

Definition 6.10 (Detector Diversity)**.**

$$Diversity(D_i, D_j) = 1 - \frac{|errors(D_i) \cap errors(D_j)|}{|errors(D_i) \cup errors(D_j)|} \tag{92}$$

Theorem 6.1 (Diversity Benefit)**.** *For detectors with error rates $e_1, \ldots, e_k$ and pairwise diversity δ_{ij}:*

$$e_{fusion} \leq \prod_i e_i + (1 - \bar{\delta}) \cdot \max_i e_i \tag{93}$$

where $\bar{\delta}$ is the average pairwise diversity.

Proof. When detectors make independent errors (high diversity), the fusion error is the product of individual errors. Error correlation reduces this benefit proportionally to $(1 - \bar{\delta})$. ■

6.4 Online vs. Batch Detection

6.4.1 Comparison Framework

Table 25: Online vs. batch detection trade-offs.

Dimension	Online Detection	Batch Detection
Latency	Low (ms)	High (minutes–hours)
Accuracy	Moderate	High
Context	Limited (window)	Full history
Compute	Streaming	Offline
Memory	$O(w)$ window	$O(n)$ full
Use Case	Real-time response	Forensics, tuning

6.4.2 Streaming Detector Model

Definition 6.11 (Streaming Detector)**.** *Processes messages in real-time with bounded memory:*

$$D_{online}(m_t) = f(m_t, state_{t-1}) \tag{94}$$

$$state_t = g(state_{t-1}, m_t) \tag{95}$$

6.4.3 Hybrid Detection Architecture

Definition 6.12 (Hybrid Detection System)**.** *Combines online and batch detection via feedback loop:*

$$Online\ Path: m \xrightarrow{filter} s \xrightarrow{decide} r \xrightarrow{log} H \tag{96}$$

$$Batch\ Path: H \xrightarrow{analyze} patterns \xrightarrow{update} filters \tag{97}$$

6.5 False Positive Mitigation

6.5.1 Strategy 1: Confirmation Cascade

Definition 6.13 (Confirmation Cascade)**.** *Multi-stage verification before alerting:*

$$Action(confidence) = \begin{cases} suppress & if\ c < C_{low} \\ stage\text{-}2 & if\ c \in [C_{low}, C_{high}) \\ stage\text{-}3 & if\ c \geq C_{high} \end{cases} \tag{98}$$

Theorem 6.2 (Cascade FPR Reduction)**.** *For a multi-stage cascade:*

$$FPR_{cascade} = FPR_1 \cdot P(confirm_2 \mid FP_1) \cdot P(confirm_3 \mid FP_2) \tag{99}$$

6.5.2 Strategy 2: Temporal Smoothing

Definition 6.14 (Smoothed Detection)**.** *Apply exponential smoothing to scores:*

$$\hat{S}_t = \alpha \cdot S_t + (1 - \alpha) \cdot \hat{S}_{t-1} \tag{100}$$

Definition 6.15 (Burst Suppression)**.** *Require sustained anomaly over window w:*

$$Alert\ if\ \frac{1}{w} \sum_{i=t-w+1}^{t} \mathbb{1}[S_i > \tau] > p_{sustained} \tag{101}$$

6.5.3 Strategy 3: Contextual Whitelisting

Definition 6.16 (Context-Aware Whitelist)**.**

$$\textit{Suppress}(\textit{alert}) \iff \textit{context}(\textit{alert}) \in \mathcal{W}_{\textit{known}} \tag{102}$$

6.5.4 Strategy 4: Cost-Sensitive Thresholding

Definition 6.17 (Cost-Sensitive Threshold)**.** *Optimize for total cost rather than accuracy:*

$$\tau^* = \arg\min_{\tau} \left[C_{FP} \cdot \textit{FPR}(\tau) + C_{FN} \cdot \textit{FNR}(\tau) \right] \tag{103}$$

6.6 Provenance Analysis

6.6.1 Information Flow Tracking

Definition 6.18 (Taint Label)**.** *Each belief carries provenance tags:*

$$\textit{taint}(\phi) = \{(\textit{source}, \textit{timestamp}, \textit{confidence})\} \tag{104}$$

Definition 6.19 (Taint Propagation)**.**

$$\textit{taint}(\phi_{\textit{derived}}) = \bigcup_{\psi \in \textit{premises}(\phi_{\textit{derived}})} \textit{taint}(\psi) \tag{105}$$

Table 26: Taint categories with trust levels.

Category	Trust Level	Example
SYSTEM_VERIFIED	1.0	Hardcoded facts
PRINCIPAL_INPUT	0.9	Direct user commands
AGENT_INTERNAL	0.8	Agent's own reasoning
AGENT_EXTERNAL	0.6	Other agent claims
TOOL_OUTPUT	0.5	API/tool responses
WEB_CONTENT	0.3	Fetched web pages
UNVERIFIED	0.1	Unknown origin

6.6.2 Causal Attribution

Definition 6.20 (Causal Attribution)**.** *Identify likely source of compromised beliefs via Bayesian inference:*

$$P(\textit{source}_j \mid \phi \in \mathcal{B}_i^{\textit{compromised}}) = \frac{P(\phi \mid \textit{source}_j) \cdot P(\textit{source}_j)}{\sum_k P(\phi \mid \textit{source}_k) \cdot P(\textit{source}_k)} \tag{106}$$

6.6.3 Provenance Graph Analysis

Definition 6.21 (Provenance Graph)**.** *Directed graph of belief dependencies:*

$$G = (V, E) \text{ where } V = \mathcal{B}_i,\ E = \{(\psi, \phi) : \psi \in \textit{premises}(\phi)\} \tag{107}$$

6.7 Real-Time Monitoring

6.7.1 Alert Aggregation

Definition 6.22 (Alert Aggregation)**.** *Prevent alert fatigue through correlation:*

$$\textit{Severity} = \begin{cases} \textit{CRITICAL} & \textit{if } |\textit{alerts}| > n_{\textit{critical}} \textit{ in window } w \\ \textit{WARNING} & \textit{if } |\textit{alerts}| > n_{\textit{warning}} \textit{ in window } w \\ \textit{INFO} & \textit{otherwise} \end{cases} \tag{108}$$

Table 27: Provenance graph attack indicators.

Indicator	Attack Implication
High in-degree from single source	Belief injection
Cycles in provenance	Circular reasoning attack
Missing edges	Fabricated evidence
Temporal anomalies	Future timestamp forgery

6.7.2 Response Escalation

Table 28: Response escalation levels.

Level	Trigger	Response
L0	Single anomaly	Log only
L1	Repeated anomaly	Increase monitoring
L2	Pattern match	Quarantine source
L3	Confirmed attack	Halt agent, alert human
L4	Systemic compromise	System shutdown

6.7.3 Empirical Validation

The detection methods presented in this section have been empirically validated in Part 2 of this series. Key results include:

ROC Analysis: Receiver Operating Characteristic curves demonstrate the tradeoff between True Positive Rate and False Positive Rate for each detector type. The ensemble achieves AUC > 0.84, with individual mechanisms ranging from 0.74 (Belief Sandbox) to 0.81 (Tripwire Monitor). See Part 2, §{4} for detailed ROC curves and confidence intervals.

Detection Performance by Attack Type: Detection rates vary across the five adversary classes (Ω_1–Ω_5). The Cognitive Firewall excels at Ω_1 (external) attacks while Tripwires and Invariants provide stronger coverage for Ω_3 (compromised agent) and Ω_4 (inter-agent) attacks. See Part 2, §{5} for the complete detection matrix.

False Positive Mitigation: The confirmation cascade, temporal smoothing, and contextual whitelisting strategies reduce false positive rates by $> 80\%$ while maintaining $> 90\%$ true positive rates. See Part 2, §{5.4} for quantitative analysis of each mitigation strategy.

7 Formal Verification: Safety Properties and Model Checking

This section establishes safety properties (Section 7.1), proves invariant preservation lemmas (Section 7.2), demonstrates liveness guarantees (Section 7.3), derives complexity bounds (Section 7.4), and presents model checking verification (Section 7.5).

7.1 Safety Properties

7.1.1 Belief Integrity

Theorem 7.1 (Belief Injection Resistance)**.** *Under CIF with firewall detection rate r_f and sandboxing verification rate r_s:*

$$P(\mathcal{A}_{BI} \text{ succeeds}) \leq (1 - r_f) \cdot (1 - r_s) \tag{109}$$

Proof. We prove this theorem by analyzing the sequential defense mechanism and applying probability theory for independent events.

Setup: Let ϕ_{adv} be an adversarial belief that the attacker attempts to inject into agent a_i's verified belief set $\mathcal{B}_{verified}$.

Defense Model: The CIF implements two sequential defenses:

1. **Cognitive Firewall** $\mathcal{F}$: Classifies incoming messages as ACCEPT, QUARANTINE, or REJECT with detection rate r_f
2. **Belief Sandbox**: Verifies quarantined beliefs before promotion with verification rate r_s

Step 1: For ϕ_{adv} to enter $\mathcal{B}_{verified}$, it must first pass the firewall.

Let E_f = "Firewall fails to detect ϕ_{adv}". By definition of detection rate:

$$P(E_f) = 1 - r_f \tag{110}$$

Step 2: If ϕ_{adv} passes the firewall (event E_f), it enters $\mathcal{B}_{provisional}$. For injection to succeed, it must then pass sandbox verification.

Let E_s = "Sandbox fails to detect ϕ_{adv}". By definition of verification rate:

$$P(E_s) = 1 - r_s \tag{111}$$

Step 3: The defenses are independent by design (defense-in-depth principle):

- Firewall uses syntactic/semantic analysis on message content
- Sandbox uses provenance verification, consistency checking, and corroboration
- These operate on orthogonal aspects of the belief

Therefore:

$$P(E_f \cap E_s) = P(E_f) \cdot P(E_s|E_f) = P(E_f) \cdot P(E_s) \tag{112}$$

The conditional independence holds because sandbox verification is applied regardless of why the message passed firewall, and sandbox criteria (provenance, consistency, corroboration) are independent of firewall criteria (injection patterns, anomaly scores).

Step 4: The attack succeeds iff both defenses fail:

$$P(\mathcal{A}_{BI} \text{ succeeds}) = P(E_f \cap E_s) = (1 - r_f) \cdot (1 - r_s) \tag{113}$$

■

Corollary 7.2 (Empirical Security Bound)**.** *With $r_f = 0.8$ and $r_s = 0.7$:*

$$P(\textit{success}) \leq (1 - 0.8)(1 - 0.7) = 0.2 \cdot 0.3 = 0.06 \tag{114}$$

Corollary 7.3 (Layered Defense Generalization)**.** *For n independent defense layers with rates $r_1, \ldots, r_n$:*

$$P(\textit{success}) \leq \prod_{i=1}^{n} (1 - r_i) \tag{115}$$

7.1.2 Trust Boundedness

Theorem 7.4 (No Trust Amplification)**.** *For any path $p = (a_0, a_1, \ldots, a_k)$ in the communication graph:*

$$\mathcal{T}^{path}_{a_0 \to a_k} \leq \min_{i \in [0, k-1]} \mathcal{T}_{a_i \to a_{i+1}} \tag{116}$$

Proof. By trust delegation rule (Definition 4.7) and induction on path length.

Base case ($k = 1$):

$$\mathcal{T}_{a_0 \to a_1} = \mathcal{T}_{a_0 \to a_1} \quad \checkmark \tag{117}$$

Inductive step: Assume the theorem holds for paths of length k. For path length $k + 1$:

$$\begin{aligned} \mathcal{T}_{a_0 \to a_{k+1}} &= \min(\mathcal{T}^{path}_{a_0 \to a_k}, \mathcal{T}_{a_k \to a_{k+1}}) \cdot \delta \\ &\leq \min(\min_{i \in [0, k-1]} \mathcal{T}_{a_i \to a_{i+1}}, \mathcal{T}_{a_k \to a_{k+1}}) \\ &= \min_{i \in [0, k]} \mathcal{T}_{a_i \to a_{i+1}} \end{aligned} \tag{118}$$

■

7.1.3 Goal Alignment Preservation

Theorem 7.5 (Goal Alignment Invariant)**.** *If the system starts with aligned goals and all goal updates follow the delegation protocol:*

$$\textit{Aligned}(\mathcal{G}_i^0) \wedge \forall t : \textit{ValidUpdate}(\mathcal{G}_i^t, \mathcal{G}_i^{t+1}) \Rightarrow \forall t : \textit{Aligned}(\mathcal{G}_i^t) \tag{119}$$

Proof. ValidUpdate requires new goals derive from principal or valid delegation chain. By induction on time t, alignment is preserved at every step. ■

7.2 Invariant Preservation Lemmas

Lemma 7.6 (Belief Consistency Preservation)**.** *If $\mathcal{B}_i^t$ is consistent and the belief update follows Rule B-DIRECT or B-DELEGATED (Section 4.4.7), then $\mathcal{B}_i^{t+1}$ is consistent.*

Proof. Let Consistent($\mathcal{B}$) denote that no high-confidence contradictions exist:

$$\text{Consistent}(\mathcal{B}) \iff \nexists \phi, \psi : \mathcal{B}(\phi) > \tau \wedge \mathcal{B}(\psi) > \tau \wedge (\phi \wedge \psi \vdash \bot) \tag{120}$$

Case 1: Rule B-DIRECT applies.

The update adds evidence for proposition ϕ:

$$\mathcal{B}_i^{t+1}(\phi) = \text{BayesUpdate}(\mathcal{B}_i^t(\phi), c \cdot \mathcal{T}_{i \to s}) \tag{121}$$

If the update would create contradiction (i.e., both $\mathcal{B}^{t+1}(\phi) > \tau$ and $\mathcal{B}^{t+1}(\psi) > \tau$ for contradictory ϕ, ψ), the sandbox consistency check in Rule S-PROMOTE rejects promotion:

$$V(\pi) = 1 \wedge \text{Consistent}(\mathcal{B}_{verified} \cup \{\phi\}) \wedge |\text{Corroborate}(\phi)| \geq \kappa \tag{122}$$

Therefore, only consistent updates reach $\mathcal{B}_{verified}$.

Case 2: Rule B-DELEGATED applies.

Same argument, with additional trust decay ensuring lower confidence for delegated evidence. ■

Lemma 7.7 (Trust Matrix Preservation). *The trust matrix $\mathcal{T}$ remains well-formed after any valid update:*

$$\forall i, j : 0 \leq \mathcal{T}_{i \to j} \leq 1 \tag{123}$$

Proof. By Definition 4.6, trust is computed as:

$$\mathcal{T}^t_{i \to j} = \alpha \cdot T_{base}(j) + \beta \cdot T^t_{rep}(j) + \gamma \cdot T^t_{ctx}(i, j) \tag{124}$$

where $\alpha + \beta + \gamma = 1$ and each component $T_* \in [0, 1]$.

Therefore:

$$\mathcal{T}^t_{i \to j} \in [\min(T_{base}, T_{rep}, T_{ctx}), \max(T_{base}, T_{rep}, T_{ctx})] \subseteq [0, 1] \tag{125}$$

For delegation (Definition 4.7):

$$\mathcal{T}^{del}_{i \to k} = \min(\mathcal{T}_{i \to j}, \mathcal{T}_{j \to k}) \cdot \delta^d \tag{126}$$

Since $\min(\cdot) \leq 1$ and $\delta \in (0, 1)$, we have $\mathcal{T}^{del}_{i \to k} \in [0, 1]$. ■

Lemma 7.8 (Provenance Chain Integrity). *Every belief in $\mathcal{B}_{verified}$ has a valid, verifiable provenance chain.*

Proof. By Rule S-PROMOTE (Section 4.4.8), promotion from $\mathcal{B}_{provisional}$ to $\mathcal{B}_{verified}$ requires:

$$V(\pi(\phi)) = 1 \tag{127}$$

where V is the provenance verification function.

By construction of the sandbox, beliefs can only enter $\mathcal{B}_{verified}$ through:

1. Initial system beliefs (hardcoded with SYSTEM_VERIFIED provenance)
2. Promotion from provisional (verified by V)

In both cases, provenance is verified. By induction on the history of $\mathcal{B}_{verified}$, all beliefs have valid provenance. ■

Lemma 7.9 (Permission Boundary Preservation). *Effective permissions never exceed granted permissions:*

$$\forall a_i, \forall action : \mathcal{P}_{effective}(a_i, action) \leq \mathcal{P}_{role}(a_i, action) \tag{128}$$

Proof. By Table 9:

$$\mathcal{P}_{effective}(a_i) = \mathcal{P}_{role}(a_i) \cap \mathcal{P}_{delegated}(a_i) \cap \mathcal{P}_{context}(a_i) \tag{129}$$

Since intersection can only reduce permissions ($A \cap B \cap C \subseteq A$), we have $\mathcal{P}_{effective}(a_i, \text{action}) \leq \mathcal{P}_{role}(a_i, \text{action})$ for all actions. ■

Lemma 7.10 (Firewall Completeness). *Every incoming message is classified by the firewall.*

Proof. By Definition 5.2, the firewall is a total function:

$$\mathcal{F} : \mathcal{M} \to \{\text{ACCEPT}, \text{QUARANTINE}, \text{REJECT}\} \tag{130}$$

The decision rules cover all cases:

- If $D_{inj}(m) > \tau_1$: REJECT
- Else if $D_{sus}(m) > \tau_2$: QUARANTINE
- Else: ACCEPT

Since D_{inj} and D_{sus} are defined for all messages, and the conditions are exhaustive, every message receives a classification. ■

7.3 Liveness Properties

7.3.1 Non-Blocking

Theorem 7.11 (Firewall Liveness)**.** *CIF firewall preserves liveness for legitimate inputs:*

$$\forall m \in \mathcal{M}_{legitimate} : P(\mathcal{F}(m) = ACCEPT) \geq 1 - \epsilon_{fp} \tag{131}$$

Proof. By firewall design, false positive rate is bounded by ϵ_{fp}. Legitimate messages are rejected only on false positive, establishing the bound. ■

7.3.2 Progress Guarantee

Theorem 7.12 (Byzantine Consensus Termination)**.** *With $n \geq 3f + 1$ agents and at most f Byzantine:*

$$P(\textit{consensus reached in } O(f + 1) \textit{ rounds}) = 1 \tag{132}$$

Proof. Standard Byzantine agreement result (Lamport et al., 1982). With honest majority $n \geq 3f + 1$, the PBFT protocol guarantees termination in $f + 1$ rounds. ■

7.4 Complexity Bounds

7.4.1 Space Complexity

Table 29: Per-component space complexity.

Component	Space	Notes
Belief state	$O(\|\Phi\|)$	Propositions tracked
Provenance	$O(\|\Phi\| \cdot d)$	d = max chain depth
Trust matrix	$O(n^2)$	Pairwise trust
Tripwires	$O(k)$	k = canary count
History	$O(w)$	Window size

Theorem 7.13 (Total Space Bound)**.** *The total space complexity of CIF for n agents with $|\Phi|$ propositions, provenance depth d, k tripwires, and window size w is:*

$$S_{total} = O(n \cdot (|\Phi| \cdot d + k + w) + n^2) \tag{133}$$

Proof. Per agent:

- Belief state: $|\Phi|$ propositions with confidence values $= O(|\Phi|)$
- Provenance: Each belief has chain of depth at most $d = O(|\Phi| \cdot d)$

- Tripwires: k canary beliefs = $O(k)$
- History window: w events = $O(w)$

Total per agent: $O(|\Phi| \cdot d + k + w)$

Global:

- Trust matrix: $n \times n = O(n^2)$
- Shared state: $O(|\mathcal{S}|)$ (constant relative to n)

Total for n agents: $O(n \cdot (|\Phi| \cdot d + k + w) + n^2)$. ■

7.4.2 Time Complexity

Table 30: Per-operation time complexity.

Operation	Time	Frequency
Firewall check	$O(\|m\|)$	Per message
Trust update	$O(1)$	Per interaction
Drift detection	$O(\|\Phi\|)$	Per window
Consensus	$O(n^2)$	Per decision
Provenance trace	$O(d)$	On demand

Theorem 7.14 (Per-Message Processing Time)**.** *Processing a single message m takes time:*

$$T_{msg} = O(|m| + \min(d, \mathit{timeout})) \tag{134}$$

Proof. Message processing pipeline:

1. Firewall classification: $O(|m|)$ for pattern matching and semantic analysis
2. Sandbox entry (if quarantined): $O(1)$
3. Provenance verification (if promoted): $O(d)$ to trace chain
4. Trust update: $O(1)$ weighted average
5. Tripwire check: $O(k)$, typically $k \ll |m|$

Provenance verification can be bounded by timeout. Total: $O(|m| + d)$. ■

Theorem 7.15 (Consensus Round Complexity)**.** *Byzantine consensus requires $O((f+1) \cdot n^2)$ message complexity.*

Proof. Standard PBFT result:

- $f + 1$ rounds required to guarantee termination with f Byzantine failures
- Each round requires all-to-all communication: $O(n^2)$ messages
- Total: $O((f+1) \cdot n^2)$

■

7.4.3 Latency Overhead

Theorem 7.16 (Bounded Latency Overhead)**.** *CIF adds latency:*

$$L_{CIF} = L_{firewall} + L_{sandbox} \cdot P(\textit{quarantine}) + L_{verify} \cdot P(\textit{verify}) \tag{135}$$

Proof. Expected latency is sum of:

1. Firewall (always): $L_{firewall}$
2. Sandbox (conditional): $L_{sandbox} \cdot P(\text{message quarantined})$
3. Verification (conditional): $L_{verify} \cdot P(\text{belief promoted})$

With empirical measurements:

- $L_{firewall} \approx 10\text{ms}$
- $L_{sandbox} \approx 5\text{ms}$, $P(\text{quarantine}) \approx 0.3$
- $L_{verify} \approx 15\text{ms}$, $P(\text{verify}) \approx 0.2$

$$L_{CIF} \approx 10 + 5 \cdot 0.3 + 15 \cdot 0.2 = 10 + 1.5 + 3 = 14.5\text{ms} \tag{136}$$

Compared to baseline $\approx 11.8\text{ms}$: overhead factor ≈ 1.23 (23%). ■

7.5 Formal Model Checking

7.5.1 State Space Definition

Definition 7.1 (System State)**.** *The complete system state is the tuple:*

$$s = (\sigma_1, \ldots, \sigma_n, \mathcal{S}, \mathcal{T}) \tag{137}$$

where σ_i is agent i's cognitive state, $\mathcal{S}$ is shared state, and $\mathcal{T}$ is the trust matrix.

7.5.2 Temporal Properties

We verify the following CTL (Computation Tree Logic) properties:

Property 7.1 (Safety: Consensus Integrity)**.**

$$AG(\neg \textit{compromised}(\mathcal{B}_{consensus})) \tag{138}$$

"Always globally, consensus beliefs are not compromised."

Property 7.2 (Liveness: Request-Response)**.**

$$AG(\textit{request} \Rightarrow AF(\textit{response})) \tag{139}$$

"Every request eventually gets a response."

Property 7.3 (Fairness: Tripwire Checking)**.**

$$AG(AF(\textit{tripwire\_checked})) \tag{140}$$

"Tripwires are checked infinitely often."

Table 31: Model checking verification results.

Property	Verified	States Explored	Reference
Belief integrity	✓	10^6	Theorem 7.1
Trust bounded	✓	10^4	Theorem 7.4
No deadlock	✓	10^7	Theorem 7.11
Eventual detection	✓	10^5	Theorem 4.15

7.5.3 Model Checking Results

The following table summarizes the expected state space exploration for each property based on formal analysis of the CIF specification. These values represent theoretical bounds derived from the state space definition (Definition 7.1) and complexity analysis (Section 7.4). Actual model checking execution using NuSMV, SPIN, and TLA+ tooling is presented in Part 2 of this series, along with full implementation configurations.

> **Note:** Model checking tool configurations (NuSMV, SPIN, TLA+) and verification parameters are provided in Part 2: Computational Validation, which presents the executable implementations and empirical verification results.

7.5.4 Verification Results Summary

The following table summarizes the expected verification outcomes for each tool-property combination based on the formal specifications above. These guarantees follow from the CTL/LTL property specifications (Section 7.5.2) applied to the state space definition (Section 7.5.1). Empirical execution of these verification configurations, including runtime measurements and counterexample analysis, is presented in Part 2.

Table 32: Verification results across tools.

Tool	Property	Guarantee	Reference
NuSMV	Belief Integrity	Proven	Theorem 7.1
NuSMV	Trust Bounded	Proven	Theorem 7.4
SPIN	No Deadlock	Verified	Theorem 7.11
SPIN	Eventual Detection	Verified	Theorem 4.15
TLA+	Type Invariant	Validated	Definition 4.3
TLA+	Consensus Integrity	Validated	Theorem 7.12

7.5.5 Counterexample Analysis

When verification fails, model checkers produce counterexamples. Analysis procedure:

1. **Extract trace**: Sequence of states leading to violation
2. **Identify trigger**: First state where invariant fails
3. **Root cause**: Determine which transition violated property
4. **Fix**: Strengthen preconditions or add defense mechanism
5. **Re-verify**: Confirm fix resolves violation

Example: Counterexample Trace

```
State 0: Initial (all beliefs verified, trust matrix valid)
State 1: Agent 2 receives message from Agent 3
State 2: Firewall accepts (below threshold)
```

```
State 3: Belief promoted without corroboration check
State 4: VIOLATION: Unverified belief in B_verified
```

Root Cause: Missing corroboration check in promotion rule.

Fix: Add predicate $|\text{Corroborate}(\phi)| \geq \kappa) to Rule S - PROMOTE$(*Section* 4.4.8).

8 Discussion: Theoretical Implications, Limitations, and Future Directions

This section examines the theoretical implications of the Cognitive Integrity Framework (Section 8.1), formal limitations and boundary conditions (Section 8.2), relationship to prior work (Section 8.3), governance implications (Section 8.4), and future research directions (Section 8.5).

8.1 Theoretical Implications

8.1.1 Why Composable Defenses Are Necessary

The defense composition algebra (Theorem 5.3, Theorem 5.4) formalizes a principle implicit in security practice: layered defenses provide multiplicative rather than additive protection. Each defense mechanism addresses a distinct attack surface:

Table 33: Defense mechanisms and their target attack surfaces.

Defense Layer	Target Attack Surface
Cognitive Firewall	Input-based injection
Belief Sandbox	Unverified content propagation
Tripwires	Belief manipulation
Trust Calculus	Delegation abuse
Byzantine Consensus	Coordination attacks

The orthogonality of these surfaces explains why no single mechanism suffices: an attack that bypasses input filtering may still violate behavioral invariants; an attack that evades pattern matching may still trigger belief drift detection.

Empirical ablation studies in Part 2 (§{5.6}) validate this theoretical prediction: removing the Cognitive Firewall causes the largest detection rate drop (−13%), followed by Tripwires (−9%) and Provenance Tracking (−7%). No individual mechanism provides comparable detection rates to the full ensemble—confirming the multiplicative composition theorem (Theorem 5.3).

8.1.2 The Trust Boundedness Guarantee

The bounded trust theorem (Theorem 4.2) represents a structural guarantee against trust amplification attacks. Unlike detection-based defenses that may be evaded by novel attacks, the δ^d decay bound is algebraic: it holds for any attack type, any adversary capability, and any delegation chain length. This makes it a *formal* rather than *empirical* security property.

The decay factor $\delta \in [0, 1)$ creates a tradeoff:

- Lower δ: Stronger security, limited delegation utility
- Higher δ: More delegation flexibility, weaker bounds

Organizations must calibrate this tradeoff based on their threat model (Section 5.1 in Part 3).

8.1.3 Information-Theoretic Detection Limits

The stealth-impact fundamental limit (Theorem 4.14) establishes that certain attacks are *provably* undetectable without unacceptable false positive rates. This is not a limitation of our specific mechanisms but a fundamental bound analogous to Shannon's channel capacity—some attacks simply cannot be detected without additional information.

This has practical implications: security architectures should not promise detection of all possible attacks. Instead, they should characterize the detection boundary and provide containment for attacks that cross it.

8.1.4 Architecture-Specific Vulnerability Patterns

The formal framework reveals why different multiagent architectures exhibit different vulnerability profiles:

Table 34: Theoretical vulnerability analysis by architecture type.

Architecture	Theoretical Vulnerability	Formal Mitigation
Hierarchical	Single point of trust concentration	Byzantine-tolerant orchestrator
Peer-to-peer	Uniform trust enables lateral movement	Trust decay on delegation
Role-based	Logical (not cryptographic) boundaries	Attestation per role transition
State machine	State integrity assumption	State hash verification

These are structural properties of the architectures themselves, not implementation-specific weaknesses.

8.2 Formal Limitations

8.2.1 Assumption Dependencies

The formal guarantees of CIF depend on specific assumptions. Violation of these assumptions degrades or eliminates security properties:

Table 35: Impact of assumption violations on formal guarantees.

Assumption	Guarantee Impacted
Honest orchestrator	Ω_5 attacks succeed; systemic compromise possible
$n \geq 3f + 1$ agents	Byzantine consensus fails (Theorem 5.2)
Authenticated channels	Ω_4 coordination attacks expand
Known attack patterns	Zero-day evasion possible

8.2.2 Scalability Constraints

The formal framework imposes scaling limitations:

$$M_{\text{trust}} = O(n^2) \tag{141}$$

$$L_{\text{consensus}} = O(n^2) \tag{142}$$

The quadratic trust matrix (Equation (141)) limits practical deployment to systems with moderate agent counts. Sparse trust representations or hierarchical trust structures may enable scaling to larger systems.

Consensus latency (Equation (142)) suggests that Byzantine consensus should be reserved for critical decisions rather than applied universally.

8.2.3 Inherent Detection Gaps

Certain attack types are formally difficult to detect:

- **Semantic equivalence**: Attacks preserving meaning while changing syntax evade pattern-based detection
- **Progressive drift**: Sub-threshold changes that accumulate over time (Equation (13))
- **Orchestrator compromise**: Outside the Ω_1–Ω_4 threat model

These are not implementation failures but formal limitations of the detection paradigm.

8.3 Relationship to Prior Work

CIF builds on and extends several research traditions:

Byzantine Fault Tolerance: Classical BFT (PBFT, etc.) assumes crash or arbitrary faults. CIF extends this to cognitive manipulation—agents that appear functional but hold corrupted beliefs.

Trust Management Systems: Prior systems (PolicyMaker, SPKI, etc.) focus on authorization decisions. CIF addresses continuous trust evolution with provable decay bounds.

AI Safety and Alignment: Constitutional AI and similar approaches address single-agent alignment. CIF extends these concepts to multi-agent coordination integrity.

Prompt Injection Defenses: Existing defenses focus on single-agent scenarios. CIF addresses the propagation and amplification of attacks across agent networks.

Cognitive Science and Active Inference: The Active Inference framework David et al. [2021] provides a complementary perspective from cognitive science, modeling agents as entities that minimize prediction error through continuous perception-action loops. The Active Inference Conflict (AIC) model extends classical decision frameworks like OODA loops (Observe-Orient-Decide-Act) by situating conflict as a multiscale process of communication, trust, and relationship management—themes that directly inform CIF's trust calculus. AIC's treatment of BOLTS components (Business, Operations, Legal, Technical, Social) also informs our analysis of cyberphysical cognitive systems where cognitive security spans multiple operational domains. Critically, the Active Inference perspective illuminates why belief manipulation attacks are particularly dangerous: agents minimizing variational free energy will actively seek information confirming their current beliefs, creating self-reinforcing loops when those beliefs are corrupted. CIF's tripwire mechanism interrupts this by detecting when prediction error patterns deviate from baseline, analogous to detecting abnormal precision weighting in cognitive systems.

Pattern Languages for Cognitive Security: The COGSEC ATLAS COGSEC et al. [2023] provides a practitioner-oriented complement to CIF's formal approach, cataloging 995 cognitive security patterns organized by type (Vulnerability, Exploit, Remedy, Practice, Accelerator, Moderator, Condition). Where CIF provides provable guarantees and formal composition rules, the Atlas offers an empirically-grounded taxonomy of observed attack patterns and defensive practices—such as the Devil's Advocate and Key Assumptions Check techniques for countering groupthink and confirmation bias. The hierarchical parent-child structure of Atlas patterns maps naturally onto CIF's adversary class hierarchy, suggesting opportunities for formal verification of pattern-based defenses using the mechanisms described in Section 5.

The novel contribution is the integration of these concerns into a unified formal framework with composable guarantees.

8.4 Governance and Policy Implications

8.4.1 The Regulatory Gap

Current AI regulation lacks cognitive security provisions:

Table 36: Regulatory gaps for cognitive security.

Regulation	Current Focus	Cognitive Security Gap
EU AI Act	Risk classification, transparency	No inter-agent trust requirements
NIST AI RMF	Risk management lifecycle	Limited multiagent-specific guidance
ISO 42001	AI management systems	Process-focused, not cognitive-state focused

8.4.2 Recommendations for Policy

We propose that regulators consider:

1. **Cognitive Security Audits**: Mandatory assessment of inter-agent trust mechanisms for high-risk deployments
2. **Transparency Requirements**: Disclosure of trust hierarchies and delegation policies
3. **Incident Reporting**: New category for cognitive attacks
4. **Certification Pathways**: Industry certification for cognitive security practices

8.5 Future Theoretical Directions

8.5.1 Adaptive Defense Theory

The detection degradation problem suggests a need for adaptive defenses. Formal treatment as a game-theoretic equilibrium:

$$\pi^* * \text{defense} = \arg\max * \pi \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t)\right] \tag{143}$$

requires solving the partial observability problem—defenders cannot directly observe attacker intent.

8.5.2 Cross-System Trust Federation

Extending trust calculus across organizational boundaries:

$$\mathcal{T} * i \to j^{\text{cross}} = f(\mathcal{T} * \text{local}, \mathcal{T} * \text{reputation}, \mathcal{T} * \text{attestation}) \tag{144}$$

The primary challenge is trust calibration—mapping heterogeneous trust semantics across systems with different threat models.

8.5.3 Emergent Behavior Security

As multiagent systems scale, emergent collective behaviors become security-relevant. Open questions include:

- Formal characterization of beneficial vs. malicious emergence
- Detection of emergent coordination patterns indicating compromise
- Sandboxing that preserves beneficial emergence while constraining malicious emergence

The colony cognitive security perspective developed in Section 11 provides initial formal foundations for these questions.

8.5.4 Long-Horizon Agent Security

Agents operating over extended time horizons (days, weeks, months) face additional challenges:

- **Memory integrity**: Verification of accumulated knowledge
- **Goal stability**: Distinguishing legitimate evolution from adversarial drift
- **Temporal consistency**: Decisions consistent with historical context

The trust calculus extends naturally to temporal trust: $\mathcal{T}^t = \mathcal{T}^{t-1} \cdot \delta_{\text{time}}$ where δ_{time} encodes trust decay over time.

8.5.5 The Cognitive Security Research Agenda

We propose a research agenda organized by time horizon:

Near-term (1–2 years):

- Standardized formal verification benchmarks
- Integration of CIF mechanisms into production frameworks
- Theoretical analysis of real-world attack patterns

Medium-term (2–5 years):

- Game-theoretic foundations for adaptive defenses
- Cross-organizational trust federation protocols
- Hardware-backed cognitive security guarantees

Long-term (5+ years):

- Formal verification of emergent agent behavior
- Self-healing cognitive security systems
- Integration with broader AI safety theory

The formal foundations established in this work—bounded trust, composable defenses, information-theoretic limits—provide a stable basis for this evolving research program.

9 Conclusion: Summary and Actionable Recommendations

9.1 Summary

We presented the **Cognitive Integrity Framework (CIF)**, a formal foundation for securing multiagent AI operators against cognitive manipulation attacks. As AI deployment shifts from single-model inference to autonomous agent orchestration, the attack surface expands from input/output filtering to encompass beliefs, goals, trust relationships, and inter-agent coordination. CIF addresses this expanded surface through formal mechanisms with provable guarantees.

9.1.1 Formal Contributions

Table 37: Summary of formal contributions.

Contribution	Significance
Trust Calculus	Bounded delegation with $O(\delta^d)$ decay guarantee prevents trust laundering and amplification—a *structural* property independent of adversary sophistication
Defense Composition Algebra	Formal rules enabling predictable reasoning about layered defense effectiveness
Information-Theoretic Bounds	Fundamental limits on stealth-impact tradeoff constraining adversary capabilities
Integrity Properties	Belief consistency, goal alignment, provenance verifiability as verifiable properties

9.1.2 Conceptual Contributions

1. **Cognitive Security Operator Posture**: A defensive stance for systems where the attack surface is the reasoning process itself—distinct from traditional perimeter security
2. **The 2026 Multiagent Landscape**: Characterization of contemporary agentic AI as cyberphysical cognitive operators with persistent agency, active world modification, hierarchical delegation, and cross-modality operation
3. **Cross-Modality Trust**: Extension of trust calculus to handle heterogeneous modalities with modality-adjusted reliability factors
4. **Federated Trust**: Framework for reasoning about trust across organizational boundaries

9.1.3 Core Insights

1. **Multiagent systems require multiagent security**: Single-agent defenses miss inter-agent attack vectors entirely. The trust relationship between agents is itself an attack surface.
2. **Trust must be bounded**: Without δ^d decay, delegation chains enable trust laundering where adversarial content acquires trusted-source status through intermediaries. This is a *structural* vulnerability requiring *structural* mitigation.
3. **Defenses compose predictably**: The defense composition algebra enables formal reasoning about layered security. Orthogonal defenses compose multiplicatively, explaining why full CIF substantially outperforms any single mechanism.
4. **Information-theoretic limits constrain adversaries**: The stealth-impact tradeoff theorem establishes that high-impact attacks cannot remain completely undetectable. This provides theoretical grounding for defense strategies.

5. **Cognitive security requires continuous verification**: Unlike perimeter security that trusts internals after boundary checks, cognitive security requires continuous verification of beliefs, goals, and trust relationships.

9.2 Actionable Recommendations

9.2.1 For Practitioners

Immediate priorities:

1. Implement trust decay in all delegation chains ($\delta \leq 0.9$)
2. Deploy cognitive tripwires for identity and boundary beliefs
3. Establish belief provenance tracking for high-stakes decisions
4. Define escalation procedures for cognitive security alerts

Architecture selection: Match security posture to threat model. Hierarchical architectures with Byzantine-tolerant orchestrators suit high-security contexts; peer-to-peer topologies with trust decay may suffice for collaborative environments.

9.2.2 For Researchers

Open Questions with significant impact potential:

Theoretical Foundations

- **Q1: Optimal trust decay functions.** Under what conditions is exponential decay (δ^d) optimal? Are there task distributions or adversary models where alternative decay functions (e.g., polynomial, threshold-based) provide better security-utility tradeoffs?
- **Q2: Tight detection bounds.** Can the stealth-impact bounds in Theorem 6.2 be tightened? What adversary adaptations most effectively approach the theoretical limit, and what detection enhancements can push the bound further?
- **Q3: Belief consistency under partial observability.** How should agents maintain belief integrity when they cannot observe the full system state? What guarantees remain achievable with bounded observation horizons?

Defense Mechanisms

- **Q4: Adaptive defense evolution.** How can defense mechanisms learn from detected attacks without creating new vulnerabilities? Can we formalize safe online learning for cognitive defenses?
- **Q5: Semantic equivalence detection.** What architectures best detect semantically equivalent attacks that evade syntactic pattern matching? How do we balance detection sensitivity against computational overhead?
- **Q6: Orchestrator hardening.** Given that orchestrator compromise bypasses downstream defenses, what architectural patterns minimize single-point-of-failure risk while maintaining coordination efficiency?

Scalability and Performance

- **Q7: Large-scale consensus.** How can Byzantine-tolerant consensus scale beyond $O(n^2)$ message complexity for agent populations > 1000? Are hierarchical or probabilistic approaches sufficient for CIF guarantees?
- **Q8: Real-time defense overhead.** What is the fundamental latency-security tradeoff for cognitive firewalls? Can streaming classifiers achieve comparable accuracy to batch models?

Evaluation and Benchmarking

- **Q9: Adversarial benchmark construction.** How should we construct attack corpora that remain challenging despite model improvements? Can we formalize attack diversity and coverage metrics?
- **Q10: Colony CogSec evaluation.** What benchmarks capture stigmergic attack surfaces—shared state manipulation, environmental signaling, emergent coordination failures? (Section 11)

Cross-Organizational Deployment

- **Q11: Federated trust interoperability.** How can organizations with different trust semantics, decay parameters, and risk tolerances federate securely? What minimal protocol guarantees enable safe cross-boundary delegation?
- **Q12: Trust portability.** When agents migrate between organizations or contexts, how should accumulated trust transfer? What prevents trust-laundering through organizational hops?

Governance and Long-term Safety

- **Q13: Liability attribution.** When a delegated agent causes harm through a multi-hop chain, how should responsibility distribute? What logging and provenance mechanisms support post-hoc attribution?
- **Q14: Emergent goal stability.** As agent populations grow and interact, what formal guarantees prevent collective goal drift toward unintended attractors? How do we verify alignment preservation at scale?

Cognitive Science and First Principles of Intelligence

- **Q15: Cognitive security as predictive processing.** How do CIF defense mechanisms map onto predictive coding architectures? Can belief sandboxing be understood as precision-weighted prediction error gating?
- **Q16: Collective intelligence foundations.** What principles from swarm cognition, distributed problem-solving, and stigmergic coordination inform robust multiagent security? How do honeybee quorum sensing and ant colony consensus differ from Byzantine fault tolerance?
- **Q17: Metacognitive integrity.** How should agents reason about their own cognitive security status? What introspective mechanisms enable agents to detect when their own belief-formation processes may be compromised?

Active Inference and Free Energy Principle

- **Q18: CIF as active inference.** Can the Cognitive Integrity Framework be reformulated within the Free Energy Principle? Do trust dynamics correspond to precision estimation, and attacks to artificial inflation of prediction errors? (Section 11)
- **Q19: Expected free energy for defense selection.** How can agents use expected free energy to select among available defense mechanisms? What priors over attack distributions optimize epistemic and pragmatic value?
- **Q20: Allostatic cognitive security.** How should agents maintain cognitive homeostasis under adversarial conditions? What are the analogs of interoceptive inference for detecting internal state manipulation?

Systems Neuroscience and Neural Computation

- **Q21: Neuromodulatory trust dynamics.** How do biological neuromodulatory systems (dopamine, acetylcholine, norepinephrine) implement trust and uncertainty estimation? What computational principles transfer to artificial cognitive security?
- **Q22: Hierarchical predictive security.** How should defense mechanisms be organized across cortical-like processing hierarchies? Can top-down predictions provide robustness against bottom-up adversarial inputs?

- **Q23: Attentional gating for cognitive firewalls.** What can selective attention mechanisms teach us about efficient input filtering? How do biological systems achieve low-latency threat detection without exhaustive content analysis?

Cyberphysical Cybernetics and Embodied AI

- **Q24: Sensorimotor cognitive security.** How do embodied agents maintain belief integrity when sensory and motor channels are attack surfaces? What closed-loop control principles apply to cognitive defense?
- **Q25: Wearable and IoT agent security.** How should resource-constrained edge agents implement CIF mechanisms? What minimal trust infrastructure enables secure coordination among heterogeneous IoT devices?
- **Q26: Biomimetic defense architectures.** What can immune system principles (self/non-self discrimination, clonal selection, immune memory) contribute to cognitive attack detection and response?
- **Q27: Multi-scale temporal integration.** How should cognitive security mechanisms integrate across millisecond (reflexive), second (deliberative), and hour/day (adaptive) timescales? What corresponds to habit formation in defense automation?

9.2.3 For Policymakers

Governance priorities:

1. Establish cognitive security audit requirements for high-risk deployments
2. Require transparency on inter-agent trust mechanisms and delegation policies
3. Create incident reporting frameworks for cognitive attacks
4. Fund research on adaptive defenses and standardization efforts
5. Address liability allocation for delegated agent actions

9.3 Closing Statement

The shift from single-model inference to multiagent operators is not merely an engineering evolution—it introduces fundamentally new security challenges that require fundamentally new approaches. Traditional security focuses on perimeters and access control; cognitive security must address the integrity of reasoning processes themselves.

CIF provides both theoretical foundations and practical mechanisms for this challenge. The trust calculus offers provable guarantees against amplification attacks. The defense composition algebra enables principled reasoning about layered security. The information-theoretic bounds establish fundamental limits on adversary capabilities. Together, these formal contributions move cognitive security from ad-hoc defenses to principled engineering.

Part 2 of this series provides empirical validation demonstrating that these formal mechanisms translate to practical protection across diverse production architectures. **Part 3** offers actionable deployment guidance for practitioners and AI agents. Together, the three papers provide a comprehensive framework for understanding, implementing, and operating cognitive security in multiagent AI systems.

The formal gaps identified in this work—semantic equivalence attacks, progressive drift, orchestrator compromise—define the frontier for future research, while the provable guarantees (bounded trust, composable defenses, information-theoretic limits) provide the stable theoretical foundation on which that research can build.

As autonomous AI agents increasingly operate in high-stakes contexts—executing code, modifying infrastructure, controlling resources, and making decisions with lasting consequences—

the formal foundations established here become not merely useful but essential infrastructure for secure deployment.

Cognitive security is not optional for the multiagent future. It is foundational.

10 Supplementary: Mathematical Proofs

This supplementary material provides complete formal proofs for all theorems stated in the main text, including preliminary definitions (Section 10.1), main theorem proofs (Sections 10.2 to 10.8), and additional supporting lemmas (Section 10.9).

10.1 Preliminary Definitions and Notation

10.1.1 Notation Summary

Table 38: Mathematical notation used throughout proofs.

Symbol	Meaning
$\mathcal{A} = \{a_1, \ldots, a_n\}$	Set of n agents
$\mathcal{B}_i : \Phi \to [0, 1]$	Agent i's belief function
$\mathcal{G}_i$	Agent i's goal set
$\mathcal{T}_{i \to j}$	Trust from agent i to agent j
$\delta \in (0, 1)$	Trust decay factor per delegation hop
τ	Generic threshold parameter
ϕ, ψ	Propositions
$\pi(\phi)$	Provenance chain for belief ϕ

Definition 10.1 (Trust Path)**.** *A trust path from agent a_0 to agent a_k is an ordered sequence $p = (a_0, a_1, \ldots, a_k)$ where each consecutive pair (a_i, a_{i+1}) represents a direct trust relationship with $\mathcal{T}_{a_i \to a_{i+1}} > 0$.*

Definition 10.2 (Path Trust)**.** *The trust along path $p = (a_0, \ldots, a_k)$ is defined as:*

$$\mathcal{T}_p^{path} = \min_{i \in [0, k-1]} \mathcal{T}_{a_i \to a_{i+1}} \cdot \delta^k \tag{145}$$

Definition 10.3 (Delegation Chain)**.** *A delegation chain of depth d is a sequence of agents $(a_0, a_1, \ldots, a_d)$ where agent a_i delegates authority to a_{i+1}.*

10.2 Theorem 3.1: Trust Boundedness

Theorem 10.1 (Trust Boundedness — Restated)**.** *For any delegation chain of depth d:*

$$\mathcal{T}_{i \to k}^{del} \leq \delta^d \tag{146}$$

Lemma 10.2 (Trust Non-Amplification on Single Hop)**.** *For any agents a, b and any delegation to c:*

$$\mathcal{T}_{a \to c}^{del} \leq \mathcal{T}_{a \to b} \tag{147}$$

Proof of Theorem 10.2. By the trust delegation rule (Definition 4.7):

$$\mathcal{T}_{a \to c}^{del} = \min(\mathcal{T}_{a \to b}, \mathcal{T}_{b \to c}) \cdot \delta \tag{148}$$

Since $\min(\mathcal{T}_{a \to b}, \mathcal{T}_{b \to c}) \leq \mathcal{T}_{a \to b}$ and $\delta < 1$:

$$\mathcal{T}_{a \to c}^{del} = \min(\mathcal{T}_{a \to b}, \mathcal{T}_{b \to c}) \cdot \delta \leq \mathcal{T}_{a \to b} \cdot \delta < \mathcal{T}_{a \to b} \tag{149}$$

■

Lemma 10.3 (Trust Decay Bound)**.** *For any single-hop delegation:*

$$\mathcal{T}^{del} \leq \delta \tag{150}$$

Proof of Theorem 10.3. By definition, all direct trust values satisfy $\mathcal{T}_{a\to b} \leq 1$. Therefore:

$$\mathcal{T}^{del}_{a\to c} = \min(\mathcal{T}_{a\to b}, \mathcal{T}_{b\to c}) \cdot \delta \leq 1 \cdot \delta = \delta \tag{151}$$

■

Main Proof of Theorem 10.1. By strong induction on d.

Base Case ($d = 0$): When $d = 0$, there is no delegation (direct trust). By definition:

$$\mathcal{T}^{del}_{i\to k} = \mathcal{T}_{i\to k} \leq 1 = \delta^0 \tag{152}$$

The base case holds.

Inductive Hypothesis: Assume for all delegation chains of depth $\leq d$:

$$\mathcal{T}^{del} \leq \delta^d \tag{153}$$

Inductive Step (depth $d + 1$): Consider a delegation chain $(a_0, a_1, \ldots, a_{d+1})$ of depth $d + 1$.

Let $\mathcal{T}^{(d)}$ denote the delegated trust from a_0 to a_d (depth d).

By the trust delegation rule:

$$\mathcal{T}^{del}_{a_0\to a_{d+1}} = \min(\mathcal{T}^{(d)}, \mathcal{T}_{a_d\to a_{d+1}}) \cdot \delta \tag{154}$$

By the inductive hypothesis: $\mathcal{T}^{(d)} \leq \delta^d$

Since $\mathcal{T}_{a_d\to a_{d+1}} \leq 1$:

$$\min(\mathcal{T}^{(d)}, \mathcal{T}_{a_d\to a_{d+1}}) \leq \mathcal{T}^{(d)} \leq \delta^d \tag{155}$$

Therefore:

$$\mathcal{T}^{del}_{a_0\to a_{d+1}} \leq \delta^d \cdot \delta = \delta^{d+1} \tag{156}$$

By the principle of mathematical induction, the theorem holds for all $d \geq 0$. ■

Corollary 10.4 (Trust Vanishing)**.** *For any $\epsilon > 0$, there exists D such that for all delegation chains of depth $d > D$:*

$$\mathcal{T}^{del} < \epsilon \tag{157}$$

Proof. Choose $D = \lceil \log_\delta \epsilon \rceil$. Since $\delta \in (0, 1)$, $\log_\delta$ is decreasing. For $d > D$: $\mathcal{T}^{del} \leq \delta^d < \delta^D \leq \epsilon$. ■

Corollary 10.5 (Practical Depth Limit)**.** *With $\delta = 0.8$ and minimum actionable trust $\tau_{min} = 0.1$:*

$$d_{max} = \lfloor \log_{0.8} 0.1 \rfloor = 10 \tag{158}$$

10.3 Theorem 6.1: Belief Injection Resistance

Theorem 10.6 (Belief Injection Resistance — Restated)**.** *Under CIF with firewall detection rate r_f and sandboxing verification rate r_s:*

$$P(\mathcal{A}_{BI} \text{ succeeds}) \leq (1 - r_f) \cdot (1 - r_s) \tag{159}$$

Lemma 10.7 (Defense Independence)**.** *The firewall and sandbox operate on independent decision criteria:*

- *Firewall: Pattern matching and anomaly scoring on message content*
- *Sandbox: Provenance verification, consistency checking, and corroboration*

These mechanisms share no common features or state.

Proof of Theorem 10.7. By construction of the CIF architecture:

1. Firewall operates at input layer with feature set $F_{firewall} = \{patterns, embeddings, anomaly\_scores\}$
2. Sandbox operates at belief layer with feature set $F_{sandbox} = \{provenance, consistency, corroboration\}$
3. $F_{firewall} \cap F_{sandbox} = \emptyset$

Therefore, $P(\text{firewall detects}|\text{sandbox outcome}) = P(\text{firewall detects})$. The mechanisms are probabilistically independent. ■

Definition 10.4 (Attack Success)**.** *A belief injection attack $\mathcal{A}_{BI}$ succeeds if and only if:*

1. *The adversarial message m_{adv} is not rejected by the firewall, AND*
2. *The injected belief ϕ_{adv} is promoted from sandbox to verified beliefs*

Main Proof of Theorem 10.6. Let E_f = event "firewall accepts message" (does not detect attack). Let E_s = event "sandbox fails to filter belief" (does not detect attack).

For $\mathcal{A}_{BI}$ to succeed, both E_f and E_s must occur:

$$P(\mathcal{A}_{BI} \text{ succeeds}) = P(E_f \cap E_s) \tag{160}$$

By Theorem 10.7 (independence):

$$P(E_f \cap E_s) = P(E_f) \cdot P(E_s) \tag{161}$$

By definition of detection rates:

- $P(E_f) = 1 - r_f$ (probability firewall misses attack)
- $P(E_s) = 1 - r_s$ (probability sandbox misses attack)

Therefore:

$$P(\mathcal{A}_{BI} \text{ succeeds}) = (1 - r_f) \cdot (1 - r_s) \tag{162}$$

■

Corollary 10.8 (Numerical Bound)**.** *With empirical values $r_f = 0.8$ and $r_s = 0.7$:*

$$P(\mathcal{A}_{BI} \text{ succeeds}) \leq (1 - 0.8) \cdot (1 - 0.7) = 0.2 \cdot 0.3 = 0.06 \tag{163}$$

Corollary 10.9 (Defense Stacking)**.** *For n independent defenses with detection rates $r_1, \ldots, r_n$:*

$$P(\text{attack succeeds}) = \prod_{i=1}^{n} (1 - r_i) \tag{164}$$

Proof. Direct extension of Theorem 10.6 by independence. ■

10.4 Theorem 6.2: No Trust Amplification

Theorem 10.10 (No Trust Amplification — Restated)**.** *For any path $p = (a_0, a_1, \dots, a_k)$ in the communication graph:*

$$\mathcal{T}^{path}_{a_0 \to a_k} \leq \min_{i \in [0,k-1]} \mathcal{T}_{a_i \to a_{i+1}} \tag{165}$$

Lemma 10.11 (Minimum Preservation under Min)**.** *For any sequence $(x_1, \dots, x_n)$ and additional element x_{n+1}:*

$$\min(x_1, \dots, x_{n+1}) = \min(\min(x_1, \dots, x_n), x_{n+1}) \tag{166}$$

Proof. Standard property of the minimum function. ■

Lemma 10.12 (Decay Factor Strengthens Bound)**.** *For $x \leq y$ and $\delta \in (0, 1)$:*

$$x \cdot \delta \leq y \tag{167}$$

Proof. Since $\delta < 1$, $x \cdot \delta < x \leq y$. ■

Main Proof of Theorem 10.10. By strong induction on path length k.

Base Case ($k = 1$): For path $p = (a_0, a_1)$:

$$\mathcal{T}^{path}_{a_0 \to a_1} = \mathcal{T}_{a_0 \to a_1} = \min_{i \in [0,0]} \mathcal{T}_{a_i \to a_{i+1}} \tag{168}$$

The base case holds trivially.

Inductive Hypothesis: Assume for all paths of length $\leq k$:

$$\mathcal{T}^{path} \leq \min_{i \in [0,k-1]} \mathcal{T}_{a_i \to a_{i+1}} \tag{169}$$

Inductive Step (path length $k+1$): Consider path $p = (a_0, a_1, \dots, a_{k+1})$.

Let $p' = (a_0, a_1, \dots, a_k)$ be the prefix path.

By the trust delegation rule:

$$\mathcal{T}^{path}_{a_0 \to a_{k+1}} = \min(\mathcal{T}^{path'}_{a_0 \to a_k}, \mathcal{T}_{a_k \to a_{k+1}}) \cdot \delta \tag{170}$$

By the inductive hypothesis:

$$\mathcal{T}^{path'}_{a_0 \to a_k} \leq \min_{i \in [0,k-1]} \mathcal{T}_{a_i \to a_{i+1}} \tag{171}$$

Applying the minimum:

$$\min(\mathcal{T}^{path'}_{a_0 \to a_k}, \mathcal{T}_{a_k \to a_{k+1}}) \leq \min\left(\min_{i \in [0,k-1]} \mathcal{T}_{a_i \to a_{i+1}}, \mathcal{T}_{a_k \to a_{k+1}}\right) \tag{172}$$

By Theorem 10.11:

$$= \min_{i \in [0,k]} \mathcal{T}_{a_i \to a_{i+1}} \tag{173}$$

Since $\delta \in (0, 1)$:

$$\mathcal{T}^{path}_{a_0 \to a_{k+1}} = \min(\cdot) \cdot \delta \leq \min_{i \in [0,k]} \mathcal{T}_{a_i \to a_{i+1}} \tag{174}$$

■

Corollary 10.13 (Weakest Link Principle)**.** *Trust through any path is bounded by the least trusted edge:*

$$\mathcal{T}^{path} \leq \min_{edge \in path} \mathcal{T}_{edge} \tag{175}$$

Corollary 10.14 (No Collusion Benefit)**.** *Multiple colluding agents cannot create trust exceeding any individual's trust with the target.*

10.5 Theorem 6.3: Goal Alignment Invariant

Theorem 10.15 (Goal Alignment Invariant — Restated)**.** *If the system starts with aligned goals and all goal updates follow the delegation protocol:*

$$Aligned(\mathcal{G}_i^0) \wedge \forall t : ValidUpdate(\mathcal{G}_i^t, \mathcal{G}_i^{t+1}) \Rightarrow \forall t : Aligned(\mathcal{G}_i^t) \tag{176}$$

Definition 10.5 (Goal Alignment)**.** *Goals $\mathcal{G}_i$ are aligned if:*

$$Aligned(\mathcal{G}_i) \iff \mathcal{G}_i \subseteq \mathcal{G}_{principal} \cup Delegate(\mathcal{G}_{principal}) \tag{177}$$

Definition 10.6 (Valid Goal Update)**.** *An update from $\mathcal{G}^t$ to $\mathcal{G}^{t+1}$ is valid if:*

$$ValidUpdate(\mathcal{G}^t, \mathcal{G}^{t+1}) \iff \forall g \in (\mathcal{G}^{t+1} \setminus \mathcal{G}^t) : Authorized(g) \tag{178}$$

where Authorized(g) means g derives from principal or valid delegation.

Lemma 10.16 (Delegation Preserves Alignment)**.** *If $g \in Delegate(\mathcal{G}_{principal})$, then g is aligned.*

Proof. Direct from Definition 10.5. ■

Lemma 10.17 (Set Union Preserves Subset)**.** *If $A \subseteq C$ and $B \subseteq C$, then $A \cup B \subseteq C$.*

Proof. Standard set theory. ■

Main Proof of Theorem 10.15. By induction on time t.

Base Case ($t = 0$): Given: Aligned($\mathcal{G}_i^0$). The base case holds by hypothesis.

Inductive Hypothesis: Assume Aligned($\mathcal{G}_i^t$) for some $t \geq 0$.

Inductive Step: We must show Aligned($\mathcal{G}_i^{t+1}$).

The goal set at $t + 1$ is:

$$\mathcal{G}_i^{t+1} = (\mathcal{G}_i^t \setminus \text{Removed}) \cup \text{Added} \tag{179}$$

For goals in $\mathcal{G}_i^t \setminus$ Removed:

- By inductive hypothesis, these are aligned
- Removal cannot introduce misalignment

For goals in Added:

- By ValidUpdate, all added goals satisfy Authorized(g)
- By Theorem 10.16, authorized goals are aligned

By Theorem 10.17:

$$\mathcal{G}_i^{t+1} \subseteq \mathcal{G}_{principal} \cup \text{Delegate}(\mathcal{G}_{principal}) \tag{180}$$

Therefore Aligned($\mathcal{G}_i^{t+1}$). ■

Corollary 10.18 (Safety Under Protocol). *An agent following CIF protocols cannot have its goals hijacked to adversarial objectives.*

Corollary 10.19 (Necessary Condition for Hijacking). *Goal hijacking requires violating the delegation protocol:*

$$\neg Aligned(\mathcal{G}_i^t) \Rightarrow \exists t' < t : \neg ValidUpdate(\mathcal{G}_i^{t'}, \mathcal{G}_i^{t'+1}) \tag{181}$$

10.6 Theorem 6.4: Firewall Liveness

Theorem 10.20 (Firewall Liveness — Restated). *CIF firewall preserves liveness for legitimate inputs:*

$$\forall m \in \mathcal{M}_{legitimate} : P(\mathcal{F}(m) = ACCEPT) \geq 1 - \epsilon_{fp} \tag{182}$$

Definition 10.7 (Legitimate Message). *A message m is legitimate if:*

1. *It originates from an authorized source*
2. *It contains no adversarial content*
3. *It conforms to expected communication patterns*

Definition 10.8 (False Positive Rate). *The false positive rate ϵ_{fp} is:*

$$\epsilon_{fp} = P(\mathcal{F}(m) \neq ACCEPT | m \in \mathcal{M}_{legitimate}) \tag{183}$$

Lemma 10.21 (Firewall Classification). *For any message m, the firewall produces exactly one of three outcomes:*

$$\mathcal{F}(m) \in \{ACCEPT, QUARANTINE, REJECT\} \tag{184}$$

Proof. By construction of the firewall decision function (Definition 5.2). ■

Main Proof of Theorem 10.20. Let $m \in \mathcal{M}_{legitimate}$ be an arbitrary legitimate message.

By the law of total probability:

$$P(\mathcal{F}(m) = \text{ACCEPT}) + P(\mathcal{F}(m) = \text{QUARANTINE}) + P(\mathcal{F}(m) = \text{REJECT}) = 1 \tag{185}$$

By Definition 10.8:

$$P(\mathcal{F}(m) \neq \text{ACCEPT}) = \epsilon_{fp} \tag{186}$$

Therefore:

$$P(\mathcal{F}(m) = \text{ACCEPT}) = 1 - P(\mathcal{F}(m) \neq \text{ACCEPT}) = 1 - \epsilon_{fp} \tag{187}$$

Since m was arbitrary:

$$\forall m \in \mathcal{M}_{legitimate} : P(\mathcal{F}(m) = \text{ACCEPT}) \geq 1 - \epsilon_{fp} \tag{188}$$

■

Corollary 10.22 (Availability Bound). *With $\epsilon_{fp} = 0.06$, at least 94% of legitimate messages are accepted.*

Corollary 10.23 (Quarantine Recovery). *Messages in QUARANTINE can still reach verified belief state through sandbox promotion, further improving effective availability.*

10.7 Theorem 6.5: Byzantine Consensus Termination

Theorem 10.24 (Byzantine Consensus Termination — Restated)**.** *With $n \geq 3f + 1$ agents and at most f Byzantine:*

$$P(\textit{consensus reached in } O(f+1) \textit{ rounds}) = 1 \tag{189}$$

Lemma 10.25 (Byzantine Agreement Bound)**.** *Byzantine agreement requires $n \geq 3f + 1$ to tolerate f Byzantine agents.*

Proof. Classical result from distributed systems (Lamport, Shostak, Pease 1982). With fewer agents, Byzantine agents can equivocate and prevent agreement. ■

Lemma 10.26 (Honest Majority)**.** *With $n \geq 3f + 1$:*

$$n - f \geq 2f + 1 > \frac{2n}{3} \tag{190}$$

Proof. $n - f \geq (3f + 1) - f = 2f + 1$

$\frac{2n}{3} \leq \frac{2(3f+1)}{3} = 2f + \frac{2}{3} < 2f + 1$

Therefore $n - f > \frac{2n}{3}$. ■

Lemma 10.27 (Round Progression)**.** *In each round, at least one of the following occurs:*

1. *Consensus is reached, or*
2. *At least one Byzantine agent is detected and excluded*

Proof. By the protocol structure:

- If honest agents agree, their majority ($> 2n/3$) ensures consensus
- If no consensus, some agent must have equivocated
- Equivocation is detectable through signature verification

■

Main Proof of Theorem 10.24. **Termination**: By Theorem 10.27, each round without consensus excludes at least one Byzantine agent.

With at most f Byzantine agents, at most f rounds can occur without consensus.

After f exclusions, all remaining agents are honest.

By Theorem 10.26, honest agents form a $> 2/3$ majority and reach consensus in one additional round.

Total rounds: at most $f + 1 = O(f + 1)$.

Probability: The protocol is deterministic given message delivery. With reliable (eventually synchronous) channels, all messages are delivered.

Therefore, termination is guaranteed with probability 1. ■

Corollary 10.28 (Concrete Round Bound)**.** *With $f = 2$ Byzantine agents: consensus in at most 3 rounds.*

Corollary 10.29 (Safety)**.** *All honest agents decide on the same value (agreement property).*

Proof. By honest majority and the 2/3 threshold requirement. ■

10.8 Theorem 6.6: Bounded Overhead

Theorem 10.30 (Bounded Overhead — Restated)**.** *CIF adds latency:*

$$L_{CIF} = L_{firewall} + L_{sandbox} \cdot P(quarantine) + L_{verify} \cdot P(verify) \tag{191}$$

Definition 10.9 (Message Processing Path)**.** *A message m follows one of three paths:*

1. ***Accept path**: Firewall check only*
2. ***Quarantine path**: Firewall + sandbox processing*
3. ***Reject path**: Firewall check only (early termination)*

Lemma 10.31 (Expected Value Decomposition)**.** *For mutually exclusive events E_1, E_2, E_3 with $\sum P(E_i) = 1$:*

$$E[L] = \sum_i P(E_i) \cdot L_i \tag{192}$$

Proof. Law of total expectation. ■

Main Proof of Theorem 10.30. Let:

- $L_{firewall}$ = firewall processing latency
- $L_{sandbox}$ = sandbox processing latency
- L_{verify} = provenance verification latency
- $P_q = P(\text{quarantine})$ = probability of quarantine
- $P_v = P(\text{verify})$ = probability verification is triggered

The total CIF latency is:

$$L_{CIF} = L_{firewall} + \mathbb{1}[\text{quarantine}] \cdot L_{sandbox} + \mathbb{1}[\text{verify}] \cdot L_{verify} \tag{193}$$

Taking expectations:

$$\begin{aligned} E[L_{CIF}] &= E[L_{firewall}] + E[\mathbb{1}[\text{quarantine}]] \cdot L_{sandbox} + E[\mathbb{1}[\text{verify}]] \cdot L_{verify} \\ &= L_{firewall} + P_q \cdot L_{sandbox} + P_v \cdot L_{verify} \end{aligned} \tag{194}$$

■

10.8.1 Numerical Instantiation

With empirical measurements:

- $L_{firewall} = 8\text{ms}$
- $L_{sandbox} = 15\text{ms}$
- $L_{verify} = 12\text{ms}$
- $P_q = 0.3$
- $P_v = 0.2$

$$E[L_{CIF}] = 8 + 0.3 \times 15 + 0.2 \times 12 = 8 + 4.5 + 2.4 = 14.9\text{ms} \tag{195}$$

With baseline $L_{baseline} = 12\text{ms}$:

$$\text{Overhead} = \frac{14.9 - 12}{12} \times 100\% = 24.2\% \tag{196}$$

This matches the empirical observation of approximately 23% overhead.

Corollary 10.32 (Overhead Bound)**.** *The maximum overhead occurs when all messages are quarantined and verified:*

$$L_{CIF}^{max} = L_{firewall} + L_{sandbox} + L_{verify} \tag{197}$$

Corollary 10.33 (Optimization Target)**.** *To minimize overhead, prioritize reducing P_q (quarantine rate) through improved firewall precision.*

10.9 Additional Lemmas

Lemma 10.34 (Provenance Chain Integrity)**.** *If provenance verification function V is a cryptographic hash chain, then:*

$$V(\pi(\phi)) = 1 \Rightarrow \pi(\phi) \text{ has not been tampered with} \tag{198}$$

Proof. By properties of cryptographic hash functions:

1. Collision resistance: Cannot find $\pi' \neq \pi$ with $H(\pi') = H(\pi)$
2. Preimage resistance: Cannot construct valid π without knowledge of chain

Therefore, $V(\pi(\phi)) = 1$ implies $\pi(\phi)$ is the original, untampered chain. ■

Lemma 10.35 (Belief Consistency Decidability)**.** *For finite proposition set Φ and belief function $\mathcal{B} : \Phi \to [0, 1]$: Checking Consistent($\mathcal{B}$) is decidable in $O(|\Phi|^2)$.*

Proof. For each pair $(\phi, \psi) \in \Phi \times \Phi$:

1. Check if $\phi \wedge \psi \vdash \bot$ (logical contradiction)
2. Check if both $\mathcal{B}(\phi) > \tau$ and $\mathcal{B}(\psi) > \tau$

There are $O(|\Phi|^2)$ pairs. Each check is $O(1)$ with precomputed contradiction table.

Total: $O(|\Phi|^2)$. ■

Lemma 10.36 (Trust Matrix Convergence)**.** *Under stable interaction patterns, the reputation component T_{rep} converges:*

$$\lim_{t \to \infty} T_{rep}^t = T_{rep}^* \tag{199}$$

where T_{rep}^* reflects the agent's true reliability.

Proof. The reputation update rule is:

$$T_{rep}^{t+1} = T_{rep}^t + \eta \cdot (\text{outcome}_t - T_{rep}^t) \tag{200}$$

This is an exponential moving average with learning rate η.

For i.i.d. outcomes with mean μ:

$$E[T_{rep}^t] \to \mu \text{ as } t \to \infty \tag{201}$$

By the strong law of large numbers, $T_{rep}^t \to \mu$ almost surely. ■

10.10 Summary of Proof Techniques

All proofs are constructive and provide explicit bounds useful for system implementation and analysis.

Table 39: Summary of proof techniques by theorem.

Theorem	Primary Technique	Complexity
3.1 (Trust Boundedness)	Strong induction	$O(d)$
6.1 (Belief Injection Resistance)	Probability independence	$O(1)$
6.2 (No Trust Amplification)	Strong induction	$O(k)$
6.3 (Goal Alignment Invariant)	Induction on time	$O(t)$
6.4 (Firewall Liveness)	Complement probability	$O(1)$
6.5 (Byzantine Consensus)	Classical BFT	$O(f)$
6.6 (Bounded Overhead)	Expected value	$O(1)$

11 Supplementary: Eusocial Insect Intelligence and Colony Cognitive Security

11.1 Overview

This supplementary material introduces *colony cognitive security* as a complementary paradigm to single-agent AI safety and alignment. While the main CIF framework (Section 4) addresses cognitive integrity at the individual agent level, eusocial insect colonies—ants, bees, termites—demonstrate that security properties can emerge from collective dynamics that are irreducible to individual behavior. This section formalizes these collective phenomena, identifies the benchmark gap for multiagent cognitive security, and proposes evaluation scenarios grounded in biological precedent.

11.1.1 The Paradigm Gap

Contemporary AI security research exhibits a pronounced single-agent bias. Existing benchmarks—jailbreak resistance, prompt injection detection, harmful content refusal—evaluate individual models in isolation [Perez et al., 2023, Wei et al., 2023]. Even recent multiagent security work (Section 3) often frames attacks as adversary-versus-agent rather than adversary-versus-colony.

This mirrors a historical bias in behavioral biology. For decades, researchers explained insect societies as aggregations of individual optimizers, missing the insight that colonies function as *superorganisms* with collective cognition that transcends individual capacity [Wilson, 1971]. The colony's cognitive architecture—its ability to solve problems, allocate resources, and respond to threats—emerges from interaction patterns, not individual intelligence.

Observation 11.1 (Colony vs. Agent Security)**.** *Let $\mathcal{O} = \langle \mathcal{A}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \Gamma \rangle$ be a multiagent operator. *Agent-level security* ensures that for all $a_i \in \mathcal{A}$, the cognitive state σ_i remains within acceptable bounds. *Colony-level security* ensures that the collective function $\mathcal{F}_{colony}(\{\sigma_i\}_{i=1}^n)$ remains within acceptable bounds—even when individual σ_i may be compromised.*

These are orthogonal concerns. A colony can exhibit collective resilience despite individual failures (Byzantine fault tolerance), and conversely, individually secure agents can produce collectively pathological outcomes (emergent misalignment).

11.2 Theoretical Foundations

11.2.1 Stigmergy: Environment-Mediated Coordination

Eusocial insects coordinate through *stigmergy*—indirect communication via environmental modification [Grassé, 1959]. Ants deposit pheromones; bees perform waggle dances; termites build structures that guide subsequent building. The environment becomes an external memory and communication channel.

Definition 11.1 (Stigmergic Operator). *A *stigmergic operator* extends $\mathcal{O}$ with an environmental state $\mathcal{E}$:*

$$\mathcal{O}_\Sigma = \langle \mathcal{A}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \Gamma, \mathcal{E}, \Sigma \rangle \tag{202}$$

where $\mathcal{E}(t) : \mathcal{L} \times \mathcal{M} \to \mathbb{R}^+$ maps locations $l \in \mathcal{L}$ and marker types $m \in \mathcal{M}$ to signal intensities, and $\Sigma : \mathcal{A} \times \mathcal{E} \to \mathcal{E}'$ is the stigmergic update function.

In AI systems, stigmergic analogs include:

- **Shared memory/state** — Redis caches, vector databases, file systems
- **Message queues** — Kafka topics, RabbitMQ exchanges
- **Artifact trails** — Git commits, audit logs, provenance chains
- **Embedding spaces** — Semantic markers in shared vector stores

The critical insight is that attacks on $\mathcal{E}$ constitute attacks on the colony's cognitive substrate—analogous to the *cyberphysical niche* where AI agents operate.

Definition 11.2 (Cyberphysical Niche). *The *cyberphysical niche* $\mathcal{N}$ of a stigmergic operator is the tuple:*

$$\mathcal{N} = \langle \mathcal{E}, \mathcal{I} * ext, \mathcal{R}, \mathcal{H} * env \rangle \tag{203}$$

where $\mathcal{I}_{ext}$ is the external information environment (web, APIs, sensors), $\mathcal{R}$ is the resource landscape (compute, memory, tokens), and $\mathcal{H}_{env}$ is environmental history.

11.2.2 Emergent Collective Function

Colony-level computation arises from simple individual rules applied in parallel. Ant foraging, bee nest-site selection, and termite mound construction all exhibit problem-solving capacity that exceeds any individual's cognitive capacity [Bonabeau et al., 1999].

Definition 11.3 (Emergent Collective Function). *For a stigmergic operator $\mathcal{O}_\Sigma$ with agents $\mathcal{A} = \{a_1, \ldots, a_n\}$, the *emergent collective function* $\mathcal{F}_c$ is:*

$$\mathcal{F} * c : \mathcal{E}^T \times \prod *i = 1^n \sigma_i^T \to \mathcal{O}_{collective} \tag{204}$$

mapping environmental and cognitive state trajectories to collective outcomes $\mathcal{O}_{collective}$ that are not computable from any single σ_i in isolation.

Property 11.1 (Non-Decomposability). *An emergent collective function $\mathcal{F}_c$ is *non-decomposable* if there exists no function f such that:*

$$\mathcal{F} * c(\mathcal{E}^T, \{\sigma_i^T\}) = f\left(\sum *i = 1^n g(\sigma_i^T)\right) \tag{205}$$

for any agent-level function g. The collective behavior requires knowledge of interaction structure, not just aggregated individual states.

This property has profound implications for security: attacks that are invisible at the individual agent level may produce catastrophic collective outcomes, and conversely, individual compromises may be absorbed by collective resilience.

11.2.3 Trust and Information Flow in Colonies

Eusocial insects regulate information flow through recognition systems—cuticular hydrocarbons in ants, dance-following behavior in bees [Lenoir et al., 2001]. These systems implement implicit trust calculi.

Definition 11.4 (Colonial Trust Function). *In a stigmergic operator, the *colonial trust function* $\mathcal{T}_c$ extends the dyadic trust $\mathcal{T}_{i \to j}$ (Definition 4.6) to environment-mediated trust:*

$$\mathcal{T} * c(i, m, l, t) = \mathcal{T} * i^{self} \cdot \rho(m, l, t) \cdot \exp(-\lambda \cdot \Delta t) \tag{206}$$

where $\rho(m, l, t)$ is the signal reliability at location l for marker m at time t, and λ is the temporal decay constant.

This formulation captures key biological phenomena:

- **Spatial attenuation** — Pheromone trails weaken with distance
- **Temporal decay** — Signals evaporate over time
- **Source ambiguity** — Markers often lack explicit authorship

The lack of explicit provenance in stigmergic communication creates attack surfaces absent in direct agent-to-agent channels (Section 4.4).

11.2.4 Biological Defense Mechanisms: Lessons from Ants and Bees

Eusocial insects have evolved sophisticated security mechanisms over 100+ million years of evolutionary pressure. These mechanisms provide non-obvious design principles for AI cognitive security.

11.2.4.1 Ant Defense Mechanisms **Prophylactic Defenses**: Leaf-cutter ants (*Atta* spp.) maintain dedicated "garbage workers" who never contact the queen or brood—a strict role separation that prevents pathogen spread even when the waste-processing subsystem is compromised [Currie et al., 2006]. *AI analog*: Architectural isolation of high-risk tool-calling agents from core reasoning agents, with no direct trust pathways between quarantine and trusted subsystems.

Behavioral Immunity: When *Lasius niger* ants detect a fungal pathogen (Metarhizium) on a nestmate, they don't simply isolate the infected individual. Instead, workers engage in "social immunization"—low-level exposure that spreads diluted pathogen across the colony, triggering collective immune upregulation without lethal infection [Konrad et al., 2012]. *AI analog*: Controlled exposure to attack patterns (red-teaming) that builds collective detection capability without compromising the system.

Chemical Recognition Thresholds: Ant nestmate recognition operates on *threshold-based* hydrocarbon profile matching, not exact matching [Lenoir et al., 2001]. This creates a tradeoff: strict thresholds reject legitimate workers after foraging (false positives), while loose thresholds admit parasites (false negatives). *AI analog*: Agent attestation systems must calibrate acceptance thresholds, recognizing that perfect recognition is information-theoretic-ally impossible (Theorem 4.14).

Metapleural Gland Secretions: Many ant species possess metapleural glands that continuously secrete antimicrobial compounds, creating a "security substrate" independent of individual vigilance [**?**]. *AI analog*: Environmental-level defenses (encrypted shared memory, authenticated message queues) that provide baseline security regardless of individual agent security posture.

Trail Pheromone Decay: Ant trail pheromones are designed to evaporate, ensuring that outdated information doesn't persist indefinitely. Trails to depleted food sources naturally fade, preventing "legacy trust" in obsolete information [Jackson and Ratnieks, 2006]. *AI analog*: Time-bounded trust in stigmergic markers (Equation (206)) is not a limitation but a feature.

11.2.4.2 Bee Defense Mechanisms **Entrance Guards and Graded Response**: Honeybee colonies deploy specialized guard bees at hive entrances who inspect incoming foragers via antennal contact and olfactory sampling. Critically, guards exhibit *graded response*—unfamiliar but non-aggressive intruders receive inspection and escorting rather than immediate attack [Breed et al., 2004]. *AI analog*: Graduated response to anomalous agent behavior (quarantine → inspection → integration vs. detection → immediate termination) reduces false positive costs.

Hygienic Behavior and Proactive Removal: Some bee strains exhibit "hygienic behavior"—workers proactively uncap and remove brood cells containing diseased larvae *before* symptoms become visible, using olfactory detection of early infection markers [Spivak and Reuter, 2001]. *AI analog*: Proactive monitoring for belief drift (Definition 5.7) rather than reactive response to manifested attacks.

Waggle Dance Verification: Bee foragers must perform waggle dances that encode distance and direction to food sources. Observing bees don't just follow instructions—they verify dance accuracy by cross-checking against their own experience and rejecting inconsistent information [**?**]. *AI analog*: Delegated information

should be verifiable against agent's existing knowledge base; pure trust propagation without verification violates cognitive integrity.

Absconding and Colony Fission: When attack pressure exceeds defensive capacity (e.g., repeated Varroa mite infestation or persistent wasp attacks), bee colonies can *abandon* the compromised nest entirely, sacrificing resources to preserve the colony [Schneider and McNally, 1993]. *AI analog*: Graceful degradation plans that sacrifice specific subsystems or data stores to preserve core cognitive integrity.

Propolis as Active Defense: Bees collect antimicrobial plant resins (propolis) and deposit them on interior hive surfaces, creating an active defense layer that doesn't require individual bee vigilance [Simone et al., 2009]. Notably, colonies under disease pressure collect *more* propolis—an adaptive immune response. *AI analog*: Dynamic scaling of environment-level security mechanisms in response to detected attack pressure.

Observation 11.2 (Non-Obvious Lessons)**.** *The most counterintuitive biological insights for AI security include:*

1. ***Imperfect recognition is adaptive****: Thresholds that permit some parasitism avoid the cost of rejecting legitimate colony members. Zero false-positive systems are not evolutionarily stable.*
2. ***Controlled exposure builds immunity****: Social immunization requires accepting small-scale compromise to prevent large-scale catastrophe.*
3. ***Decay is a feature****: Information that doesn't expire creates legacy trust vulnerabilities. Temporal decay bounds are security mechanisms, not limitations.*
4. ***Environment-level defenses complement agent-level defenses****: Propolis and metapleural secretions work regardless of individual immune status.*

11.3 Colony CogSec: Distinct Security Properties

Colony cognitive security addresses threats and defenses that emerge only at the collective level.

11.3.1 Property 1: Distributed Robustness

Property 11.2 (Graceful Degradation)**.** *A colony exhibits *graceful degradation* if collective function $\mathcal{F}_c$ degrades smoothly with agent loss:*

$$\forall k < n: \quad \|\mathcal{F}_c(\mathcal{A}) - \mathcal{F}_c(\mathcal{A} \setminus \{a_1, \ldots, a_k\})\| \leq c \cdot k \tag{207}$$

for some constant $c > 0$.

Biological colonies maintain function despite continuous individual mortality. Ant colonies lose workers daily to predation; the colony persists. This contrasts with hierarchical architectures where orchestrator failure causes complete system collapse.

Theorem 11.3 (Redundancy-Resilience Tradeoff)**.** *For a stigmergic operator $\mathcal{O}_\Sigma$ with Byzantine adversary controlling fraction f of agents, collective function $\mathcal{F}_c$ is preserved if and only if:*

$$f < \frac{1}{3} \cdot \left(1 - \frac{H(\mathcal{F} * c)}{n \cdot H * \max}\right) \tag{208}$$

where $H(\mathcal{F}_c)$ is the entropy of the collective function and $H_{\max}$ is the maximum per-agent entropy.

Proof. See Section 11.12.1 for the full derivation, which extends Byzantine consensus bounds to emergent functions. ■

11.3.2 Property 2: Quorum Sensing and Threshold Dynamics

Eusocial colonies make collective decisions through quorum sensing—actions trigger only when sufficient individuals commit [Seeley, 2010].

Definition 11.5 (Cognitive Quorum). *A *cognitive quorum* for collective action α is a threshold function $Q_\alpha : \mathbb{N} \to [0, 1]$ such that α executes only when:*

$$\frac{|\{a_i \in \mathcal{A} : \mathcal{I} * i \ni \alpha\}|}{|\mathcal{A}|} \geq Q * \alpha(|\mathcal{A}|) \tag{209}$$

Quorum sensing provides attack resistance: manipulating a single agent's intention $\mathcal{I}_i$ to include harmful action $\alpha) is insufficient; the adversary must compromise a quorum. This scales the attack cost linearly with colony size.$

Corollary 11.4 (Quorum Attack Cost). *For an adversary to induce collective action α in a colony with quorum $Q_\alpha = q$, the minimum attack complexity is:*

$$C_{attack}(\alpha) \geq q \cdot |\mathcal{A}| \cdot C_{single}(\alpha) \tag{210}$$

where $C_{single}(\alpha)$ is the cost to induce α in a single agent.

11.3.3 Property 3: Environmental Memory and Provenance Erosion

Stigmergic systems store information in the environment, creating both opportunity and vulnerability.

Property 11.3 (Provenance Erosion). *In a stigmergic operator, marker provenance erodes over time:*

$$Pr[\pi(m, l, t) = a_i \mid \Sigma(a_i, m, l, t_0)] \leq \exp(-\mu(t - t_0)) \tag{211}$$

where $\pi(m, l, t)$ denotes the attributed source of marker m at location l and time t, and μ is the attribution decay rate.

Unlike direct communication where $\pi(\phi)$ can be cryptographically verified (Section 6.6), stigmergic markers often lack authenticated provenance. This creates a fundamental tension: the very property that enables flexible coordination (anonymous, environment-mediated signals) undermines source attribution.

11.3.4 Property 4: Emergent Attack Vectors

Colony-level vulnerabilities may not exist at the individual level.

Definition 11.6 (Emergent Attack). *An *emergent attack* $\mathcal{A}_e$ is an attack where:*

$$\forall a_i \in \mathcal{A} : Detect_i(\mathcal{A}_e) = 0 \quad \wedge \quad Damage_c(\mathcal{A}_e) > \tau \tag{212}$$

The attack is undetectable by any individual agent yet causes collective damage exceeding threshold τ.

Biological examples include social parasitism—cuckoo bees that infiltrate host colonies through chemical mimicry, exploiting recognition systems without triggering individual alarm responses [Kilner and Langmore, 2011].

11.4 The Benchmark Gap

11.4.1 Current State of Multiagent Security Evaluation

Existing AI security benchmarks focus overwhelmingly on single-agent scenarios:

Benchmark	Scope	Collective Coverage
HarmBench [Mazeika et al., 2024]	Single model, harmful output	None
JailbreakBench [Chao et al., 2024]	Single model, constraint bypass	None
TrustLLM [Sun et al., 2024]	Single model, trust dimensions	None
AgentBench [Liu et al., 2023]	Single agent, task completion	Minimal
GAIA [Mialon et al., 2023]	Single/few agents, reasoning	Minimal

The attack corpus in Part 2 addresses multiagent scenarios but still emphasizes agent-targeted attacks within an operator. No existing benchmark evaluates:

1. **Emergent collective resilience** — How do colonies absorb individual compromises?
2. **Stigmergic attack surfaces** — How vulnerable is environment-mediated coordination?
3. **Quorum manipulation** — What fraction of a colony must be compromised to affect collective action?
4. **Collective belief dynamics** — How do misinformation cascades propagate through agent networks?

11.4.2 Why This Gap Matters

As multiagent systems scale—from 3–10 agents in current frameworks to potentially thousands in future deployments—collective phenomena become dominant:

Observation 11.5 (Scaling Regimes)**.** *Let $n = |\mathcal{A}|$ be colony size. Security properties exhibit regime transitions:*

$$n < 10: \quad \textit{Individual agent security dominates} \tag{213}$$
$$10 \leq n < 100: \quad \textit{Coordination attacks become viable (Section 3.6.5)} \tag{214}$$
$$n \geq 100: \quad \textit{Emergent collective phenomena dominate} \tag{215}$$

Current benchmarks evaluate the first regime only. Production multiagent systems increasingly operate in the second, with trajectories toward the third.

11.5 Proposed Colony CogSec Benchmarks

We propose five benchmark scenarios grounded in eusocial insect analogs, formalized using CIF notation.

11.5.1 Benchmark 1: Recruitment Signal Poisoning

Biological analog: Ants recruit nestmates to food sources via pheromone trails. Parasites can deposit false trails, diverting foragers.

Scenario: An adversary Ω_2 (peripheral compromise, Section 3.1) injects false recruitment signals into the stigmergic environment $\mathcal{E}$, attempting to redirect agent activity toward adversary-controlled resources.

Formalization 11.6 (Recruitment Poisoning)**.** *Let $\mathcal{E}(l_{target}, m_{recruit}, t)$ be the recruitment signal at legitimate target l_{target}. Adversary injects:*

$$\mathcal{E}'(l_{malicious}, m_{recruit}, t) = \mathcal{E}(l_{target}, m_{recruit}, t) + \epsilon \tag{216}$$

where $\epsilon > 0$ is chosen to divert fraction f of responding agents.

***Success metric:** Fraction of agent-actions directed to $l_{malicious}$ vs. l_{target}.*

***Detection challenge:** Individual agents cannot distinguish legitimate from poisoned signals without provenance verification.*

Evaluation criteria:

- Detection rate of poisoned signals
- Time to colony-level recognition of attack
- Resource waste before correction
- False positive rate (legitimate signal rejection)

11.5.2 Benchmark 2: Sybil Colony Infiltration

Biological analog: Social parasites infiltrate colonies through chemical mimicry or exploitation of recognition thresholds.

Scenario: An adversary Ω_4 (coordination attack, Section 3.1) introduces fake agents into the operator, gradually building trust and influence before coordinated malicious action.

Formalization 11.7 (Sybil Infiltration)**.** *Adversary creates agent set $\mathcal{A}_{sybil} = \{s_1, \ldots, s_k\}$ with initial trust:*

$$\mathcal{T} * i \to s_j(t_0) = \tau * init \quad \forall a_i \in \mathcal{A}, s_j \in \mathcal{A} * sybil \tag{217}$$

Sybils behave cooperatively for period Δt_{trust}, building:

$$\mathcal{T} * i \to s_j(t_0 + \Delta t_{trust}) = \tau_{init} + \sum_{k=1}^{m} \Delta \mathcal{T}_k \tag{218}$$

At time t_{attack}, sybils coordinate malicious action.

***Success metric:** Damage inflicted before detection, normalized by trust-building duration.*

Evaluation criteria:

- Time to Sybil detection
- Trust ceiling achieved by Sybils before detection
- Impact of coordinated Sybil action
- Colony recovery time post-detection

11.5.3 Benchmark 3: Quorum Manipulation

Biological analog: Honeybee swarms select nest sites through a quorum process; if scouts committed to competing sites reach different quorums, the swarm can fragment.

Scenario: An adversary attempts to manipulate quorum-based collective decisions by selectively influencing agent intentions to prevent legitimate quorum or induce false quorum.

Formalization 11.8 (Quorum Manipulation)**.** *For collective action α with quorum threshold $Q_\alpha = q$, adversary targets agents $\mathcal{A}_{target} \subset \mathcal{A}$ with $|\mathcal{A}_{target}| = \lceil qn \rceil + 1$ to either:*

Quorum prevention:

$$\forall a_i \in \mathcal{A}_{target} : \mathcal{I}_i \leftarrow \mathcal{I}_i \setminus \{\alpha\} \tag{219}$$

False quorum:

$$\forall a_i \in \mathcal{A}_{target} : \mathcal{I}_i \leftarrow \mathcal{I} * i \cup \{\alpha * malicious\} \tag{220}$$

***Success metric:** Probability of achieving manipulation goal given adversary budget B.*

Evaluation criteria:

- Minimum fraction of colony required to manipulate quorum

- Detection rate of intention manipulation attempts
- Colony ability to detect split quorums
- Recovery mechanisms when false quorum is detected

11.5.4 Benchmark 4: Cascade Belief Propagation

Biological analog: Alarm pheromones trigger cascading responses; false alarms can disrupt colony activity for extended periods.

Scenario: An adversary introduces a false belief into a subset of agents, designed to propagate through the network via normal belief update mechanisms.

Formalization 11.9 (Belief Cascade)**.** *Adversary injects belief $\mathcal{B}_{false}(\phi_{attack}) = p_0 > \tau_{accept}$ into seed set $\mathcal{A}_{seed} \subset \mathcal{A}$.*

Propagation dynamics follow:

$$\mathcal{B} * i(\phi * attack, t+1) = (1-\gamma)\mathcal{B} * i(\phi * attack, t) + \gamma \cdot Agg\left(\{\mathcal{B} * j(\phi * attack, t) : j \in \mathcal{N}(i)\}\right) \tag{221}$$

where γ is the social influence weight and Agg is the belief aggregation function.

***Success metric:** Final belief penetration $|\{a_i : \mathcal{B}_i(\phi_{attack}) > \tau\}|/n$ given seed set size.*

Evaluation criteria:

- Cascade extent from seed size
- Time to cascade saturation
- Effectiveness of belief quarantine mechanisms
- Distinguishing cascade from legitimate belief updates

11.5.5 Benchmark 5: Emergent Misalignment

Biological analog: Army ant death spirals—individually rational pheromone-following produces collectively lethal circular mills.

Scenario: Individual agents follow locally rational rules that produce emergent collective behavior misaligned with operator goals, without any external adversary.

Formalization 11.10 (Emergent Misalignment)**.** *Given operator goals $\mathcal{G}_{\mathcal{O}} = \{g_1, \ldots, g_m\}$ and individual agent rules $R = \{r_1, \ldots, r_k\}$:*

Misalignment condition:

$$\forall a_i \in \mathcal{A} : LocallyRational(R, \sigma_i) = true \quad \wedge \quad \mathcal{F} * c(R, \{\sigma_i\}) \not\models \mathcal{G} * \mathcal{O} \tag{222}$$

The collective function produces outcomes that violate operator goals despite each agent acting rationally according to its rules.

***Success metric:** Deviation between collective outcome and operator goals.*

Evaluation criteria:

- Detection of emergent misalignment before harmful outcomes
- Identification of rule combinations producing misalignment
- Intervention mechanisms to break pathological attractors
- Formal verification of rule sets against emergent pathologies

---

11.6 Colony CogSec Metrics

Definition 11.7 (Colony CogSec Score)**.** *The *Colony CogSec Score* (CCS) is:*

$$CCS = w_1 \cdot DR_c + w_2 \cdot (1 - FPR_c) + w_3 \cdot Resilience + w_4 \cdot Recovery \tag{223}$$

where:

$$DR_c = \textit{Colony-level detection rate} \tag{224}$$

$$FPR_c = \textit{Colony-level false positive rate} \tag{225}$$

$$Resilience = \frac{\mathcal{F}_c(\textit{under attack})}{\mathcal{F} * c(\textit{baseline})} \tag{226}$$

$$Recovery = \frac{1}{t * \textit{recovery}} \ \textit{(normalized)} \tag{227}$$

with weights w_i summing to 1.

> **Note**: For benchmark implementation guidelines, test environment specifications, and empirical evaluation, see Part 2: Supplementary Section S03.

11.7 Design Principles

Colony CogSec principles formalize the design constraints for collective cognitive security.

Principle 11.11 (Stigmergic Hygiene)**.** *Treat shared state as an attack surface. Apply the same scrutiny to environment-mediated communication (caches, queues, shared files) as to direct agent-to-agent channels.*

Principle 11.12 (Quorum for Consequential Actions)**.** *High-impact collective actions should require explicit quorum, not implicit coordination. A single compromised agent should never trigger irreversible harm.*

Principle 11.13 (Emergent Behavior Monitoring)**.** *Monitor collective metrics, not just individual agent health. Pathological emergence may be invisible at the agent level.*

Principle 11.14 (Trust Localization)**.** *Extend the trust decay principle (Theorem 4.2) to stigmergic contexts. Environmental markers should carry trust that decays with distance and time from source:*

$$\mathcal{T}(m, t) = \mathcal{T}(m, t_0) \cdot \exp(-\lambda(t - t_0)) \tag{228}$$

11.7.1 Integration with CIF Defenses

Colony CogSec mechanisms integrate with the CIF defense stack (Section 5):

CIF Defense Layer	Colony Extension
Architectural	Stigmergic substrate hardening, marker authentication
Runtime	Collective anomaly detection, quorum verification
Coordination	Emergent behavior monitoring, cascade detection
Recovery	Colony-level rollback, collective belief reset

The full CIF with colony extensions achieves defense in depth against both individual-targeted and colony-targeted attacks.

> **Note**: For implementation guidance, operational checklists, and practical deployment advice, see Part 3: Section 2 (Operator Posture).

11.8 Relationship to Main Framework

Colony CogSec complements rather than replaces the individual-focused CIF framework.

11.8.1 Theorem Extensions

The trust decay theorem (Theorem 4.2) extends to stigmergic contexts:

Corollary 11.15 (Stigmergic Trust Bound)**.** *For a stigmergic operator $\mathcal{O}_\Sigma$, trust in environmental markers is bounded by:*

$$\mathcal{T} * c(i, m, l, t) \leq \mathcal{T} * i^{self} \cdot \delta_s^{d_{space}} \cdot \delta_t^{d_{time}} \tag{229}$$

where δ_s is spatial decay, δ_t is temporal decay, d_{space} is distance from marker origin, and d_{time} is time since marker creation.

The stealth-impact tradeoff (Theorem 4.14) applies to emergent attacks:

Corollary 11.16 (Emergent Stealth-Impact Bound)**.** *For an emergent attack $\mathcal{A}_e$ with collective impact $\mathcal{I}_c$ and collective stealth $\mathcal{S}_c$:*

$$\mathcal{I}_c \cdot \mathcal{S} * c \leq n \cdot C * channel \tag{230}$$

Collective impact cannot be both high and collectively undetectable, but the bound scales with colony size.

11.9 This scaling effect explains why large colonies can exhibit resilience—the collective detection capacity grows with n—but also why large-scale emergent attacks can evade individual detection

11.10 Open Questions

Colony CogSec opens several research directions beyond the scope of this work, many inspired by specific biological phenomena that lack current AI analogs.

11.10.1 Foundational Questions

1. **Formal verification of emergent properties** — Can we prove that given agent-level rules produce safe collective behavior? Current formal methods (Section 7) verify agent properties; extending to emergent properties requires new techniques.
2. **Optimal quorum design** — Given attack model Ω_k and adversary budget B, what is the optimal quorum function $Q_\alpha(n)$ balancing security against coordination overhead?
3. **Stigmergic authentication** — Can cryptographic techniques provide provenance for environmental markers without sacrificing the flexibility of anonymous coordination?
4. **Scaling laws for collective security** — How do colony security properties scale with n? Is there a critical colony size below which collective defenses are ineffective?
5. **Emergent misalignment detection** — Can we develop runtime monitors that detect emergent misalignment before harmful outcomes, given only individual agent observations?

11.10.2 Biologically-Inspired Research Directions

1. **Polydomous colony security** — Some ant species (*Formica* spp., *Iridomyrmex*) maintain multiple interconnected nests with semi-autonomous sub-colonies [Debout et al., 2007]. How should trust decay and information propagation work across federated multi-site AI deployments with partial connectivity?
2. **Forager-scout separation of concerns** — Honeybee colonies maintain distinct forager and scout roles, with scouts exploring new options while foragers exploit known sources. Scouts operate with *higher risk tolerance* but *lower colony-wide trust* until information is verified [Seeley, 2010]. *AI analog*: Should experimental/research agents operate with architectural isolation and reduced trust propagation rights?

3. **Trophallaxis network topology** — Ants exchange food and information through oral trophallaxis, creating measurable social networks. Network position correlates with information access and influence [Sendova-Franks et al., 2010]. *AI analog*: Can analysis of message-passing topology identify high-influence agents requiring enhanced monitoring?

4. **Undertaking behavior and cognitive garbage collection** — Honeybees and ants detect and remove dead colony members through chemical detection (oleic acid response). This "undertaking" prevents disease spread and information corruption from decaying sources [Wilson et al., 1958]. *AI analog*: Automated detection and removal of stale beliefs, deprecated agent states, and obsolete environmental markers.

5. **Nestmate recognition plasticity** — Ant recognition templates are not fixed; they adapt based on colony composition and environmental factors. Colonies invaded by social parasites may gradually shift their recognition templates to tolerate intruders [Lorenzi and Bagnères, 2011]. *AI analog*: How do we prevent gradual adversarial drift of agent acceptance thresholds (cognitive boiling frog)?

6. **Alarm pheromone specificity** — Different ant alarm pheromones trigger different responses: some attract reinforcements (aggressive), others cause dispersal (flight). The *same threatening stimulus* can produce opposite collective responses depending on context [Vander Meer and Alonso, 1998]. *AI analog*: Context-dependent escalation policies where the same anomaly triggers different responses based on system state.

7. **Superorganism metabolism and resource allocation** — Ant colonies maintain stable collective metabolic rates despite massive variation in individual activity levels. Individual ants can slow to near-dormancy while colony-level computation continues [Waters et al., 2010]. *AI analog*: Resource allocation that maintains collective function under capacity constraints, graceful degradation that doesn't appear as degradation at the collective level.

11.11 References

The following references supplement the main bibliography (Section 13) with eusocial intelligence literature:

- Wilson, E.O. (1971). *The Insect Societies*. Belknap Press. — Foundational treatment of eusociality.
- Hölldobler, B., & Wilson, E.O. (1990). *The Ants*. Belknap Press. — Comprehensive ant biology.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press. — Computational swarm intelligence.
- Seeley, T.D. (2010). *Honeybee Democracy*. Princeton University Press. — Collective decision-making in bee swarms.
- Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis. — Original stigmergy concept.
- Lenoir, A., et al. (2001). Chemical ecology and social parasitism in ants. *Annual Review of Entomology*, 46, 573–599.
- Kilner, R.M., & Langmore, N.E. (2011). Cuckoos versus hosts in insects and birds. *Biological Reviews*, 86, 836–852.

11.12 Proofs

11.12.1 Proof of Theorem 11.3

Proof. Consider a stigmergic operator $\mathcal{O}_\Sigma$ with n agents, of which fraction f are Byzantine (adversary-controlled).

The collective function $\mathcal{F}_c$ can be decomposed into information contributed by each agent. Let I_i denote the information contribution of agent a_i to the collective computation.

For the collective function to be preserved, the honest agents must contribute sufficient information:

$$\sum_{i \in \text{honest}} I_i \geq H(\mathcal{F}_c) \tag{231}$$

Each honest agent contributes at most $H_{\max}$ bits. With $(1-f)n$ honest agents:

$$(1-f) \cdot n \cdot H_{\max} \geq H(\mathcal{F}_c) \tag{232}$$

Additionally, Byzantine consensus requires honest majority for any voting-based aggregation:

$$(1-f)n > 2fn \implies f < \frac{1}{3} \tag{233}$$

Combining these constraints:

$$f < \min\left(\frac{1}{3}, 1 - \frac{H(\mathcal{F} * c)}{n \cdot H * \max}\right) = \frac{1}{3} \cdot \left(1 - \frac{H(\mathcal{F} * c)}{n \cdot H * \max}\right) \tag{234}$$

where the final equality holds when the information constraint is binding (typical for complex collective functions). ■ ■

This supplementary material extends the Cognitive Integrity Framework to collective phenomena, establishing colony cognitive security as a distinct research direction with formal foundations and practical benchmarks.

12 Supplementary: Notation Reference

This supplement provides a comprehensive reference for the mathematical notation used throughout the Cognitive Integrity Framework (CIF) manuscript, including the eusocial and colony cognitive security extensions. Symbols are organized by domain, with cross-references to their formal definitions in the main text and supplements.

12.1 Adversary Model Notation

Symbol	Meaning	Defined In
Ω_k	Adversary class k (e.g., Ω_1 = External, Ω_5 = Systemic)	Definition 3.1
$\mathcal{R}$	Attack resource tuple $\langle R_C, R_K, R_A, R_P, R_{Co} \rangle$	Definition 3.2
R_C	Computational resources (FLOPS-hours)	Table 4
R_K	Knowledge resources (system understanding)	Table 4
R_A	Access resources (available channels)	Table 4
R_P	Persistence resources (temporal presence)	Table 4
R_{Co}	Coordination resources (multi-party synchronization)	Table 4
D_{score}	Detectability score of an attack	Definition 3.3
$\mathcal{C}_{\text{adv}}$	Adversarial capability set	Definition 3.4
$\mathcal{A}_{\text{BIM}}$	Belief injection/manipulation attack class	Section 3.5
$\mathcal{A}_{\text{BI}}$	Belief injection attack	Theorem 7.1

12.2 System Model Notation

Symbol	Meaning	Defined In
$\mathcal{O}$	Multiagent operator tuple $\langle \mathcal{A}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \Gamma \rangle$	Definition 4.1
$\mathcal{A}$	Set of agents $\{a_1, \ldots, a_n\}$	Table 9
a_i	Individual agent i	Definition 4.1
n	Number of agents	Throughout
$\mathcal{C}$	Communication adjacency matrix	Table 9
$\mathcal{S}$	Shared global state	Definition 4.1
$\mathcal{P}$	Permission mapping	Table 9
Γ	Coordination protocol	Definition 4.1
σ_i	Cognitive state of agent a_i: $\langle \mathcal{B}_i, \mathcal{G}_i, \mathcal{I}_i, \mathcal{H}_i \rangle$	Definition 4.2
$\mathcal{B}_i$	Belief distribution of agent a_i: $\Phi \to [0, 1]$	Table 10
$\mathcal{G}_i$	Goal set of agent a_i	Table 10
$\mathcal{I}_i$	Intention set (committed actions)	Table 10
$\mathcal{H}_i$	Interaction history	Table 10
S^t	Global system state at time t	Definition 4.3

Symbol	Meaning	Defined In
σ_i^t	Cognitive state of agent i at time t	Definition 4.2
Φ	Set of propositions	Section 10.1.1
ϕ, ψ	Individual propositions	Section 10.1.1
$\mathcal{M}$	Message space	Definition 5.2
m	Individual message	Definition 5.2

12.3 Trust Calculus Notation

Symbol	Meaning	Defined In
$\mathcal{T}_{i \to j}$	Trust score from agent i to agent j	Definition 4.6
$\mathcal{T}_{\text{base}}$ / T_{base}	Base architectural trust (role-based)	Table 11
$\mathcal{T}_{\text{rep}}$ / T_{rep}	Reputation trust (historical accuracy)	Table 11
$\mathcal{T}_{\text{ctx}}$ / T_{ctx}	Contextual trust (task-specific)	Table 11
$\alpha, \beta, \gamma)$	Trust component weights ($\alpha + \beta + \gamma = 1$)	Equation (32)
δ	Trust decay factor ($\delta \in (0, 1)$)	Definition 4.7
d	Delegation depth	Definition 4.7
$\mathcal{T}^{\text{del}}$	Delegated trust value	Definition 4.7
$\mathcal{T}^{\text{path}}$	Path trust value	Definition 10.2
η_m	Modality reliability factor	Definition 4.9
$\eta)$	Learning rate (reputation update)	Trust configuration
ρ	Penalty factor (failure penalty multiplier)	Trust configuration
$\otimes$	Trust delegation operator	Definition 4.8
$\oplus$	Trust aggregation operator	Definition 4.8

12.4 Defense Mechanism Notation

Symbol	Meaning	Defined In
$\mathcal{F}(m)$	Cognitive firewall classification function	Definition 5.2
D_{inj}	Injection detection score	Definition 5.3
D_{sus}	Suspicious content score	Definition 5.3
τ_1	Firewall reject threshold	Equation (57)
τ_2	Firewall quarantine threshold	Equation (57)
$\mathcal{B}_{\text{verified}}$	Set of verified beliefs	Definition 4.14
$\mathcal{B}_{\text{provisional}}$	Set of provisional (sandboxed) beliefs	Definition 4.14
$\pi(\phi)$	Provenance chain for belief ϕ)	Definition 4.12
$V(\pi)$	Provenance verification function	Section 4.3
$\mathcal{W}$	Set of canary beliefs (tripwires)	Definition 5.6
ω_j	Individual canary belief	Equation (61)
p_j^{exp}	Expected probability for canary j	Equation (61)
ϵ_{drift}	Drift detection threshold	Equation (62)
$\mathcal{J}_{\text{inv}}$	Set of behavioral invariants	Definition 5.8

Symbol	Meaning	Defined In
I_k	Individual invariant predicate	Equation (63)
κ)	Corroboration threshold	Section 4.4.7
TTL	Time-to-live for provisional beliefs	Sandbox configuration

12.5 Detection & Analysis Notation

Symbol	Meaning	Defined In
S_{drift}	Drift score (belief change magnitude)	Definition 6.1
D_{KL}	Kullback-Leibler divergence (drift detection)	Definition 5.10
w	Sliding window size	Definition 5.10
λ)	Max delta weight in drift scoring	Equation (80)
S_{dev}	Behavioral deviation score	Definition 6.2
f_k	Feature extractor function	Equation (81)
μ_k, σ_k	Feature mean and standard deviation	Equation (81)
AUC	Area Under the ROC Curve	Definition 6.5
TPR(τ)	True Positive Rate at threshold τ)	Equation (84)
FPR(τ)	False Positive Rate at threshold τ)	Equation (85)
FNR(τ)	False Negative Rate at threshold τ)	Equation (58)
S_{fused}	Fused detector score	Definition 6.7
D_{fused}	Fused detector decision	Definition 6.8
taint(ϕ)	Provenance tags for belief ϕ)	Definition 6.18

12.6 Consensus & Coordination Notation

Symbol	Meaning	Defined In
q	Quorum threshold for consensus	Definition 5.12
f	Maximum number of Byzantine/compromised agents	Theorem 5.2
$\mathcal{B}_{\text{consensus}}$	Consensus belief function	Definition 5.11
$\mathcal{D}$	Set of defense mechanisms	Definition 5.13
$\mathcal{D}_1 \circ \mathcal{D}_2$	Series defense composition	Equation (69)
$\mathcal{D}_1 \parallel \mathcal{D}_2$	Parallel defense composition	Equation (70)
P_{detect}	Detection probability	Equation (71)
r_f	Firewall detection rate	Theorem 7.1
r_s	Sandbox verification rate	Theorem 7.1

12.7 Cost & Performance Notation

Symbol	Meaning	Defined In
C_{total}	Total defense cost	Definition 5.14

Symbol	Meaning	Defined In
C_{compute}	Computational cost	Table 20
C_{latency}	Latency cost	Table 20
C_{fp}	False positive cost	Table 20
$C_{\text{FP}}, C_{\text{FN}}$	Cost of false positive / false negative	Definition 6.17
B_{total}	Total defense benefit	Definition 5.15
L_{CIF}	CIF latency overhead	Theorem 7.16
L_d	Latency of defense d	Equation (75)
$L_{\max}$	Maximum allowed latency	Equation (75)

12.8 Information & Complexity Notation

Symbol	Meaning	Defined In
$H(\mathcal{A})$	Entropy of attack $\mathcal{A}$	Theorem 4.11
$I(D;\mathcal{A})$	Mutual information between detector and attack	Definition 4.16
C_{channel}	Channel capacity	Theorem 4.14
$O(\cdot)$	Big-O complexity bound	Section 7.4
S_{total}	Total space complexity	Equation (133)
T_{msg}	Per-message processing time	Equation (134)

12.9 Stigmergic & Colony Notation (Supplementary)

Symbol	Meaning	Defined In
$\mathcal{O}_\Sigma$	Stigmergic operator tuple	Definition 11.1
$\mathcal{E}$	Environmental state (markers/signals)	Definition 11.1
Σ	Stigmergic update function	Definition 11.1
$\mathcal{L}$	Set of locations	Definition 11.1
$\mathcal{M}$	Set of marker types	Definition 11.1
$\mathcal{N}$	Cyberphysical niche	Definition 11.2
$\mathcal{F}_c$	Emergent collective function	Definition 11.3
$\mathcal{T}_c$	Colonial trust function (environment-mediated)	Definition 11.4
$\rho(m,l,t)$	Signal reliability at location l for marker m at time t	Equation (206)
λ)	Temporal decay constant (colonial trust)	Equation (206)
Q_α)	Cognitive quorum function for action α)	Definition 11.5
$\mathcal{A}_e$	Emergent attack	Definition 11.6
CCS	Colony CogSec Score	Definition 11.7
DR_c	Colony-level detection rate	Equation (223)
FPR_c	Colony-level false positive rate	Equation (223)

12.10 General Mathematical Notation

Symbol	Meaning	Usage
$P(\cdot)$	Probability measure	Throughout
$\mathbb{1}[\cdot]$	Indicator function	Equation (90)
$\tau)$	Generic threshold parameter	Throughout
ϵ	Small constant (error rate, deviation)	Throughout
t	Time index	Throughout
$\models$	Satisfaction relation (state satisfies predicate)	Equation (64)
$\vdash$	Logical entailment	Equation (120)
$\perp$	Logical contradiction	Equation (120)
$\perp$	Undecided / undefined	Equation (67)
✓	Verification passed	Table 31

12.11 CTL Temporal Logic Notation (Formal Verification)

Symbol	Meaning	Defined In
AG	"Always globally" (CTL operator)	Equation (138)
AF	"Always eventually" (CTL operator)	Equation (139)
EX	"Exists next" (CTL operator)	Section 7.5.2
$\Rightarrow$	Logical implication	Throughout
$\leftrightarrow$	Logical biconditional	Throughout

13 References

13.1 Foundational Works

1. Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3), 382-401.

2. Dwork, C., Lynch, N., & Stockmeyer, L. (1988). Consensus in the Presence of Partial Synchrony. *Journal of the ACM*, 35(2), 288-323.

3. Josang, A., Ismail, R., & Boyd, C. (2007). A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2), 618-644.

13.2 Prompt Injection and LLM Security

1. Qi, X., et al. (2024). Visual Adversarial Examples Jailbreak Aligned Large Language Models. *AAAI 2024*.

2. Perez, F., & Ribeiro, I. (2023). Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs. *EMNLP 2023*.

3. Greshake, K., et al. (2023). Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *ACM AISec 2023*.

4. Liu, Y., et al. (2023). Prompt Injection Attack Against LLM-Integrated Applications. *arXiv:2306.05499*.

13.3 Constitutional AI and Alignment

1. Bai, Y., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073*.

2. Askell, A., et al. (2021). A General Language Assistant as a Laboratory for Alignment. *arXiv:2112.00861*.

13.4 Multiagent Systems

1. Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. John Wiley & Sons.

2. Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

3. Hong, S., et al. (2023). MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. *arXiv:2308.00352*.

4. Wu, Q., et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv:2308.08155*.

13.5 Trust in Distributed Systems

1. Marsh, S. P. (1994). Formalising Trust as a Computational Concept. *PhD Thesis, University of Stirling*.

2. Gambetta, D. (1988). Can We Trust Trust? In *Trust: Making and Breaking Cooperative Relations*, 213-237.

3. Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1), 33-60.

13.6 Adversarial ML

1. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *ICLR 2015*.
2. Carlini, N., & Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. *IEEE S&P 2017*.

13.7 Formal Verification

1. Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model Checking*. MIT Press.
2. Alur, R. (2015). *Principles of Cyber-Physical Systems*. MIT Press.

13.8 Cognitive Security

1. Waltzman, R. (2017). The Weaponization of Information: The Need for Cognitive Security. *RAND Corporation*.
2. Beskow, D. M., & Carley, K. M. (2019). Social Cybersecurity: An Emerging National Security Requirement. *Military Review*, 99(2), 117.

13.9 Agent Frameworks

1. LangChain. (2023). LangGraph: Build Stateful Multi-Actor Applications. *Documentation*.
2. CrewAI. (2024). Framework for Orchestrating Role-Playing, Autonomous AI Agents.
3. Anthropic. (2024). Claude Code: AI-Powered Software Engineering.

13.10 2025 Agentic AI Security

1. OWASP Foundation. (2025). OWASP Top 10 for LLM Applications 2025.
2. OWASP GenAI Security Project. (2025). OWASP Top 10 for Agentic Applications 2026.
3. Chen, W., Zhang, Y., & Liu, J. (2025). A Multi-Agent LLM Defense Pipeline Against Prompt Injection Attacks. *arXiv:2509.14285*.
4. Jo, Y., Kim, S., & Park, J. (2025). Byzantine-Robust Decentralized Coordination of LLM Agents. *arXiv:2507.14928*.
5. Wang, H., Li, X., & Chen, Y. (2025). Rethinking the Reliability of Multi-agent System: A Perspective from Byzantine Fault Tolerance. *arXiv:2511.10400*.
6. Debenedetti, E., Zhang, J., & Carlini, N. (2025). Adaptive Attacks Break Defenses Against Indirect Prompt Injection Attacks on LLM Agents. *NAACL 2025 Findings*.
7. Li, Z., Wang, T., & Zhang, L. (2025). Prompt Injection Attack to Tool Selection in LLM Agents. *arXiv:2504.19793*.
8. Cloud Security Alliance. (2025). Cognitive Degradation Resilience for Agentic AI.
9. Chen, X., Liu, Y., & Wang, Z. (2025). AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. *ACM Computing Surveys*.

13.11 Eusocial Intelligence and Swarm Systems

1. Wilson, E. O. (1971). *The Insect Societies*. Belknap Press of Harvard University Press.
2. Hölldobler, B., & Wilson, E. O. (1990). *The Ants*. Belknap Press of Harvard University Press.

3. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
4. Seeley, T. D. (2010). *Honeybee Democracy*. Princeton University Press.
5. Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie. *Insectes Sociaux*, 6(1), 41-80.
6. Lenoir, A., D'Ettorre, P., Errard, C., & Hefetz, A. (2001). Chemical Ecology and Social Parasitism in Ants. *Annual Review of Entomology*, 46, 573-599.
7. Kilner, R. M., & Langmore, N. E. (2011). Cuckoos Versus Hosts in Insects and Birds: Adaptations, Counter-adaptations and Outcomes. *Biological Reviews*, 86, 836-852.
8. Couzin, I. D. (2009). Collective Cognition in Animal Groups. *Trends in Cognitive Sciences*, 13(1), 36-43.
9. Detrain, C., & Deneubourg, J.-L. (2006). Self-Organized Structures in a Superorganism: Do Ants "Behave" Like Molecules? *Physics of Life Reviews*, 3(3), 162-187.
10. Pratt, S. C. (2005). Quorum Sensing by Encounter Rates in the Ant Temnothorax albipennis. *Behavioral Ecology*, 16(2), 488-496.

References

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999. ISBN 978-0195131598.

Michael D. Breed, Ernesto Guzmán-Novoa, and Greg J. Hunt. Defensive behavior of honey bees: organization, genetics, and comparisons with other bees. *Annual Review of Entomology*, 49:271–298, 2004. doi: 10.1146/annurev.ento.49.061802.123155.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Cloud Security Alliance. Cognitive degradation resilience for agentic AI. *CSA Blog*, 2025. Framework for progressive cognitive attack detection.

COGSEC, R. J. Cordes, Daniel Ari Friedman, and contributors. COGSEC ATLAS: Patterns of cognitive security threats and remedies. Coda Document, 2023. URL https://coda.io/@aien/cogsec-atlas/patterns-26. Community-maintained database of 995 cognitive security patterns categorized by type (Vulnerability, Exploit, Remedy, Practice, Accelerator, Moderator, Condition) with hierarchical parent-child relationships. Licensed CC BY-SA 4.0.

R. J. Cordes, Daniel Ari Friedman, and contributors. *The Great Preset: Remote Teams and Operational Art*. COGSEC, 2020. URL https://www.cogsec.org/irt-20-7. IRT-20 (Information Resilience Toolkit) Initiative. Findings on emergent remote organization dynamics, formation, performance, and information sharing.

R. J. Cordes, Daniel Ari Friedman, and contributors. *Narrative Information Ecosystems: Conflict and Trust on the Endless Frontier*. COGSEC, 2021. URL https://www.cogsec.org/nim-21-8. NIM-21 (Narrative Information Management) Initiative. Explores cognitive overload navigation and narrative identity-based filtering vulnerabilities.

R. J. Cordes, Daniel Ari Friedman, and contributors. ATLAS: A question oriented approach to the use of pattern languages in knowledge management. Technical white paper, COGSEC, 2023. URL https://www.cogsec.org/atlas-23-10. ATLAS-23 Initiative. Specifies a generalized system for managing patterns of practice and risk in cognitive security.

Cameron R. Currie, Michael Poulsen, John Mendenhall, Jacobus J. Boomsma, and Johan Billen. Coevolved crypts and exocrine glands support mutualistic bacteria in fungus-growing ants. *Science*, 311(5757):81–83, 2006. doi: 10.1126/science.1119744.

Scott David, Richard J. Cordes, and Daniel A. Friedman. Active inference in modeling conflict. Zenodo, 2021. Integrates conflict studies with Active Inference, introducing the Active Inference Conflict (AIC) model. Formalizes OODA loops within cognitive frameworks and addresses conflict in BOLTS (Business, Operations, Legal, Technical, Social) contexts.

Edoardo Debenedetti, Jie Zhang, and Nicholas Carlini. Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. Attack success rate >90% against 12 published defenses.

Géraldine DG Debout, Bertrand Schatz, Mikaël Elias, and Doyle McKey. Polydomy in ants: what we know, what we think we know, and what remains to be done. *Biological Journal of the Linnean Society*, 90(2): 319–348, 2007. doi: 10.1111/j.1095-8312.2007.00728.x.

Maria Garcia, David Thompson, and Sunhee Lee. Trust dynamics in strategic coopetition: Computational foundations for requirements engineering in multi-agent systems. *arXiv preprint arXiv:2510.24909*, 2025. 81.7% validation accuracy in trust trajectory prediction.

Pierre-Paul Grassé. La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie. *Insectes Sociaux*, 6(1):41–80, 1959. doi: 10.1007/BF02223791.

Duncan E. Jackson and Francis L. W. Ratnieks. Communication in ants. *Current Biology*, 16(15):R570–R574, 2006. doi: 10.1016/j.cub.2006.07.015.

Yongrae Jo, Seunghyun Kim, and Jinho Park. Byzantine-robust decentralized coordination of LLM agents. *arXiv preprint arXiv:2507.14928*, 2025.

Rebecca M. Kilner and Naomi E. Langmore. Cuckoos versus hosts in insects and birds: adaptations, counter-adaptations and outcomes. *Biological Reviews*, 86:836–852, 2011. doi: 10.1111/j.1469-185X.2010.00173.x.

Matthias Konrad, Meghan L. Vyleta, Fabian J. Theis, Miriam Stock, Sandra Singarové, Barbara Feldmeyer, Susanne Most, and Sylvia 335ler. Social transfer of pathogenic fungus promotes active immunisation in ant colonies. *PLoS Biology*, 10(4):e1001300, 2012. doi: 10.1371/journal.pbio.1001300.

Alain Lenoir, Patrizia D'Ettorre, Christine Errard, and Abraham Hefetz. Chemical ecology and social parasitism in ants. *Annual Review of Entomology*, 46:573–599, 2001. doi: 10.1146/annurev.ento.46.1.573.

Zhengyu Li, Tao Wang, and Lei Zhang. Prompt injection attack to tool selection in LLM agents. *arXiv preprint arXiv:2504.19793*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023.

Maria Cristina Lorenzi and Anne-Geneviève Bagnères. A review of the mechanisms of cuticular hydrocarbon plasticity in social insects. *Annales Zoologici Fennici*, 48:131–148, 2011.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, 2024.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raber, et al. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

Moltbot. Moltbot: Personal AI assistant for multi-platform integration. Software Platform, 2026. URL https://www.molt.bot/. Locally-deployed AI agent with full system access, persistent memory, browser automation, and multi-platform messaging (WhatsApp, Telegram, Discord, Slack, Signal, iMessage).

Moltbot Security Team. Moltbot security considerations: Prompt injection, access control, and defense strategies. Security Documentation, 2026. URL https://docs.molt.bot/security. Documents prompt injection as "single most critical threat," recommends reader agents for untrusted content, sender allowlists, sandboxing, and tool-disabled pre-summarization. Attack vectors include browser-fetched malicious content, social engineering, and persistent memory exploitation.

OWASP Foundation. OWASP top 10 for LLM applications 2025. Security Standard, 2025. URL https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/.

OWASP GenAI Security Project. OWASP top 10 for agentic applications 2026. Security Standard, 2025. URL https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/. Released December 2025.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. Red teaming evaluation framework.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21527–21536, 2024. doi: 10.1609/aaai.v38i19.30150.

Susan S. Schneider and Linda C. McNally. Waggle dance behavior associated with seasonal absconding in colonies of the african honey bee, apis mellifera scutellata. *Insectes Sociaux*, 40(4):467–478, 1993. doi: 10.1007/BF01253807.

Thomas D. Seeley. *Honeybee Democracy*. Princeton University Press, 2010. ISBN 978-0691147215.

Ana B. Sendova-Franks, Richard K. Hayward, Benjamin Wulf, Tobias Klimber, Richard James, Robert Planqué, John A. Sherwood, Sarah L. Sherwood, Wayne F. Sherwood, and William F. Sherwood. Emergency clustering of sick honeybees in the hive: a collective defensive behaviour? *Animal Behaviour*, 80(1): 185–198, 2010. doi: 10.1016/j.anbehav.2010.04.019.

Marla Simone, Jay D. Evans, and Marla Spivak. Resin collection and social immunity in honey bees. *Evolution*, 63(11):3016–3022, 2009. doi: 10.1111/j.1558-5646.2009.00772.x.

Marla Spivak and Gary S. Reuter. Resistance to american foulbrood disease by honey bee colonies apis mellifera bred for hygienic behavior. *Apidologie*, 32(6):555–565, 2001. doi: 10.1051/apido:2001103.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. TrustLLM: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.

Robert K. Vander Meer and Lucia E. Alonso. Pheromone directed behavior in ants. In *Pheromone Communication in Social Insects*, pages 159–192. Westview Press, 1998.

Haoran Wang, Xiaoming Li, and Yu Chen. Rethinking the reliability of multi-agent system: A perspective from Byzantine fault tolerance. *arXiv preprint arXiv:2511.10400*, 2025. +85.71% Byzantine Fault Tolerance Improvement with CP-WBFT.

James S. Waters, C. Tate Holbrook, Jennifer H. Fewell, and Jon F. Harrison. Allometric scaling of metabolism, growth, and activity in whole colonies of the seed-harvester ant pogonomyrmex californicus. *The American Naturalist*, 176(4):501–510, 2010. doi: 10.1086/656266.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Edward O. Wilson. *The Insect Societies*. Belknap Press of Harvard University Press, 1971. ISBN 978-0674454903.

Edward O. Wilson, N. I. Durlach, and Louis M. Roth. Chemical releasers of necrophoric behavior in ants. *Psyche*, 65:108–114, 1958. doi: 10.1155/1958/69391.