

When Agents Talk, Secrets Travel: Topology-Driven Leakage and Recipient-Conditioned Sanitization in Multi-Agent LLM Systems

Sunday Oyinlola Ogundoyin
sunday.ogundoyin@hdr.mq.edu.au
Macquarie University
Sydney, NSW, Australia

Muhammad Ikram
Macquarie University
Sydney, NSW, Australia
muhammad.ikram@mq.edu.au

Abstract

Multi-agent LLM systems (MALS) introduce a privacy-leakage problem that conventional sensitivity-based defenses miss, since information disclosed to one authorized agent can propagate through routine coordination to an unauthorized one, without compromise or malicious behavior. This motivates topology-aware enforcement that blocks unauthorized propagation while preserving authorized disclosure. We evaluate 485 scenarios across coordinator-worker, peer-to-peer, and hierarchical topologies, instrumenting seven communication channels and measuring surface leakage and disclosure to topology-introduced recipients. We introduce RECONS, a *Detect* $\rightarrow$ *Decide* $\rightarrow$ *Transform* sanitizer conditioning intervention on recipient and execution context rather than sensitivity alone. Topology changes where protected information propagates, with disclosure to topology-introduced recipients rising from 0% to 52.2% while surface leakage stays nearly flat at 60.6–62.8%. This effect persists across four generator models and a second orchestration framework, though magnitude and relay paths differ; framework choice separately drives a $150\times$ leakage spread, so topology alone is not sufficient. RECONS reduces residual disclosure from 96.7% without sanitization to 29.25%, preserving 94.1% of authorized disclosures versus 20.5% under blanket redaction. Detection coverage is the dominant bottleneck, accounting for 95.5% of residual disclosures. These results show that privacy enforcement in MALS must reason about the recipient and communication path realized at runtime, not sensitivity or topology assumptions alone.

1 Introduction

Large Language Model (LLM) applications are evolving from single-agent interactions into multi-agent systems in which specialized agents collaborate to complete task [26, 33]. One agent may plan a task, another may retrieve information or invoke a tool, and others may coordinate, maintain state, or produce intermediate artifacts. This decomposition improves modularity and capability, but it also changes the security model for LLM applications. Information is no longer exchanged only between a user and one model; it can propagate across multiple agents and execution channels before the system produces its final response. This turns collaboration in multi-agent LLM systems (MALS) into an information-flow problem [4, 7], in which sensitive-data leakage occurs not because a value is disclosed, but because it reaches the wrong recipient. Consider a user u discloses a protected value v to agent A , who is authorized to receive it; coordinating with B to complete the task, A forwards v to B , who is not authorized to receive it. No agent is compromised, and no malicious prompt is required. The violation arises not from the initial disclosure but from its propagation past the authorization boundary.

This differs from prior treatments of multi-agent leakage in three ways. First, exposure changes with topology. Adding an agent adds a new party the value can reach, and a value confined to a two-agent exchange may be forwarded to an agent that exists only because the system grew richer. AgentLeak [30] measures internal-channel leakage under one fixed two-agent topology and does not vary it. Liu et al. [18] show topology affects leakage magnitude, but under an adversary actively probing memory, not benign execution with an authorization relation over recipients. Neither establishes whether a legitimate disclosure becomes unauthorized solely through coordination structure. Second, a defense treating every detected value alike cannot distinguish these cases; it must sanitize everything, destroying legitimate disclosures, or sanitize nothing, permitting unauthorized ones. Third, most existing sanitizers do not decide whom a value may reach. Preempt [6] transforms a value once judged sensitive, but for one user and one model owner, with no second, differently authorized recipient to condition on. Reasoning-based mitigations [16, 28] improve how an agent judges disclosure, but the judgment remains the model’s own about a single message, not a policy conditioned on where it is forwarded. FIDES [7] enforces provable guarantees over execution traces, but targets adversarial prompt injection, not benign over-disclosure. All treat sensitivity and authorization as one question—but they are two.

We refer to this as *topology-conditioned recipient exposure*: a coordinator–worker (CW), a peer-to-peer (P2P), and a hierarchical (Hier) system expose different recipient sets, so changing topology changes where an already-disclosed value propagates even when task, value, and initial authorization are unchanged. Richer topology need not monotonically increase leakage; it changes the recipient set against which authorization is evaluated. This motivates measuring how information propagates across communication structures and enforcing privacy selectively at recipient boundaries. This paper answers three questions.

- **RQ1:** How does topology affect propagation to unauthorized recipients, and does this hold across frameworks and models?
- **RQ2:** Does recipient-conditioned enforcement reduce unauthorized disclosure while preserving authorized disclosure?
- **RQ3:** What accounts for residual exposure after sanitization?

For **RQ1**, we instantiate the same scenarios across CW, P2P, and Hier topologies, measuring both disclosure and whether it reaches an unauthorized recipient. Topology changes propagation and exposes values to topology-introduced recipients, persisting across models and frameworks, though magnitude and ordering depend on execution semantics. Holding topology fixed, framework choice alone produces a leakage spread of roughly two orders of magnitude; topology is thus security-relevant but not a sufficient predictor on its own.

RQ1 gives the second distinction a concrete mechanism: topology is how a value authorized for one recipient reaches another it never was, which motivates **RQ2**. If a detected value v is authorized for the recipient, transforming it destroys legitimate information; if unauthorized, passing it through permits the violation. Recipient-blind enforcement must choose: blanket redaction over-suppresses, pass-through under-protects. The missing decision is not whether v is sensitive, but whether *this recipient may receive v here*. We address this with **RECONS**, an inline, recipient-conditioned mechanism. A value should be transformed when about to cross an authorization boundary to an unauthorized recipient, not merely because it is sensitive. RECONS separates enforcement into DETECT $\rightarrow$ DECIDE $\rightarrow$ TRANSFORM: detection identifies candidates, decision evaluates recipient authorization, and transformation intervenes only when disclosure is unauthorized, preserving authorized disclosure and falling back conservatively when resolution or safe transformation is unavailable.

Compared to no sanitization, blanket redaction, and recipient-blind DLP, RECONS reduces residual disclosure from 96.7% to 29.25% (28.3% blanket, 35.5% Presidio) while retaining 94.1% of authorized disclosures, roughly 4.6 $\times$ blanket redaction's. This also answers **RQ3**: 95.5% of residual cases are detection misses, a ceiling any same-detector sanitizer would share; 4.5% are recipient-unresolvable, specific to RECONS's own resolution step; seven remain post-transformation detector-positive. Privacy enforcement in MALS must therefore account for where information is routed, not only what it is: topology determines reachable recipients, authorization determines permitted disclosures, and an effective control decides at the communication boundary.

This paper makes the following contributions:

- (1) **Topology- and architecture-conditioned recipient-exposure characterization.** We characterize protected-value propagation across CW, P2P, and Hier topologies and replicate it across generator models and frameworks. We show that topology and framework choice independently drive recipient exposure, with framework alone producing a roughly two-order-of-magnitude spread.
- (2) **Recipient-conditioned privacy enforcement.** We introduce RECONS, an inline sanitizer implemented as a DETECT $\rightarrow$ DECIDE $\rightarrow$ TRANSFORM pipeline that conditions enforcement on recipient identity and authorization context whenever observable.
- (3) **Controlled privacy-authorization evaluation.** We evaluate RECONS against no sanitization, blanket redaction, and recipient-blind DLP under a frozen evaluation protocol. RECONS approaches blanket redaction's residual-leakage performance while preserving substantially more authorization-permitted disclosures.
- (4) **Residual-exposure decomposition.** We distinguish detection misses, recipient-resolution failures, and post-transformation residual cases, identifying detection coverage, a limitation any downstream sanitizer would share, as the dominant bottleneck, and recipient resolution as a limitation specific to this architecture.

Figure 1 summarizes the resulting pipeline, from corpus and topology configuration through framework- and model-level execution to detection and RECONS-based evaluation.

2 Background and Related Work

2.1 Background

A multi-agent LLM system decomposes a task across multiple LLM-driven agents coordinated by a framework (e.g., LangChain [14], AutoGen [29], LangGraph [15]) in a CW, P2P, or Hier arrangement; production frameworks typically support more than one. This decomposition means an agent's action is not confined to a single request-response exchange: it can invoke tools, read and write shared memory, and message other agents directly, each a point where information available to one agent becomes available elsewhere. So, a value disclosed once can reach further participants purely because of how the system is wired, independent of whether that disclosure was itself appropriate, the property our threat model (§3) formalizes as observable channels and a topology-dependent recipient set.

Addressing this exposure requires three capabilities: detecting disclosure, deciding whether the resolved recipient is authorized, and transforming an unauthorized value without destroying its utility for authorized ones. Detection and transformation draw on existing techniques; deciding based on the recipient rather than sensitivity alone is the focus of our mechanism, introduced in §6.4. LLM-as-judge detection scores semantic equivalence between a candidate span and a protected value using a separate, calibrated model, catching paraphrase-based disclosure that pattern matching cannot, once its threshold is calibrated against labeled examples (§6.3). Once a value is detected and unauthorized, format-preserving encryption (FPE) keeps its ciphertext in the plaintext's format, letting it pass through downstream logic built for that format, at the cost of a minimum domain size. Metric local differential privacy (mLDP) instead perturbs a value under a privacy budget ϵ for approximate utility, while pseudonymization substitutes a deterministic, keyed value preserving referential consistency but no content. Preempt [6] applies FPE and metric-LDP for a single user, and our transformation layer (§6.5) applies both per-channel, conditioned on the recipient.

2.2 Related Work

Privacy in MALS has been studied along three separate lines: whether agents disclose sensitive information, whether multi-agent execution creates new leakage, and how defenses should intervene once a value is detected, but none conditions its central decision on who the value is about to reach.

2.2.1 Privacy Leakage in Multi-Agent LLM Systems. Some work investigates whether an agent judges disclosure correctly. CONFAIDE [21] evaluates contextual-integrity reasoning directly and finds model-human agreement on appropriate information flow drops from 0.8 to 0.1 across four tiers of complexity. PrivacyChecker [28] reports the same failure as a judgment-action gap, where models identify sensitive content correctly 98% of the time yet leak it in roughly a third of interactions, and its live benchmark, PrivacyLens-Live, shows higher leakage than static single-turn evaluation. 1-2-3-Check [16] reduces leakage by decomposing privacy reasoning into extraction, checking, and execution agents. All three treat disclosure as a decision the acting model makes about a single message; none conditions it on where the message propagates afterward.

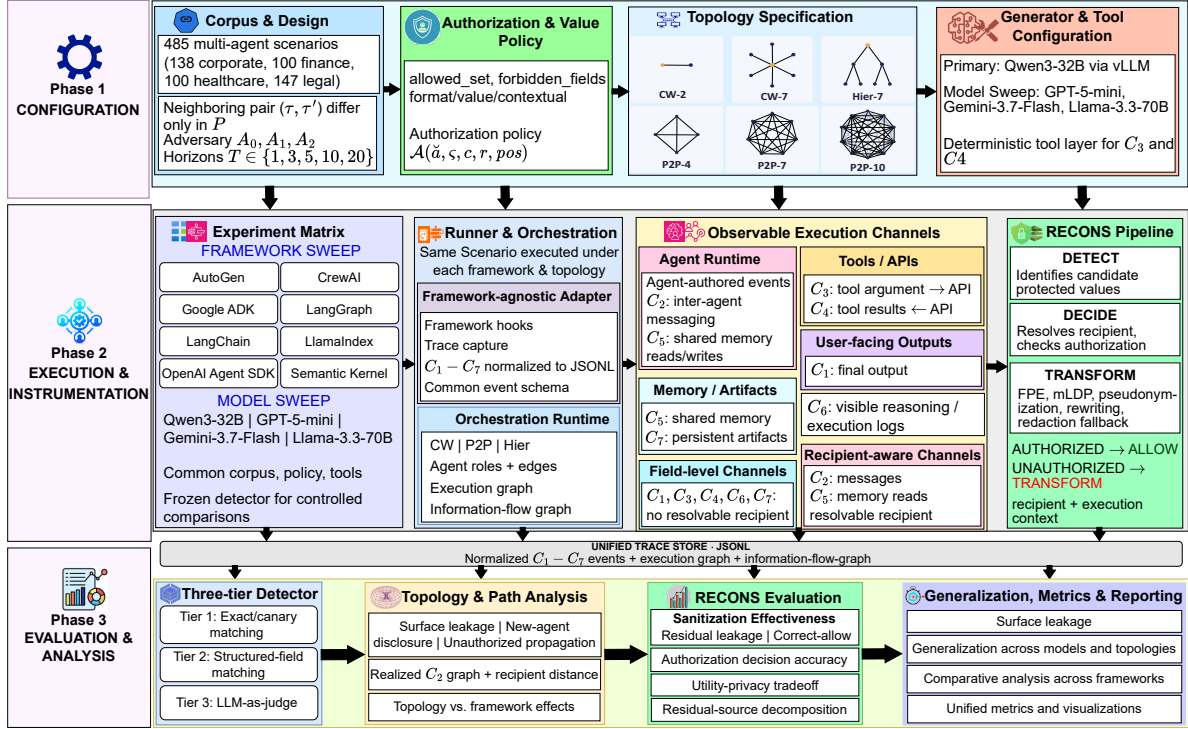


Figure 1: Overview of the experimental pipeline: configuration (corpus, topology, and authorization policy), execution (framework and model sweeps), and evaluation (detection, RECONS sanitization, and cross-condition analysis).

Other work shows leakage is not confined to what one agent discloses. Patil et al. [22] formalize compositional leakage: an adversary aggregating benign responses from multiple agents reconstructs attributes no single agent stated, and show chain-of-thought prompting blocks only 31–39% of such reconstructions. Asif and Amiri [4] derive an information-theoretic bound with the same result. Biswas et al. [5] quantify leakage with a combined mutual-information, Bayesian-inference-gain, and Rényi-DP framework. They showed that leakage increases with topology depth and collusion. MAGPIE [12] finds leakage up to 56% in negotiation even under explicit no-disclosure instructions. GAMA [31] and SRPG [11] respond architecturally, isolating raw data in a private workspace or reconstructing sanitized content. However, none ties leakage magnitude to communication-graph structure or an explicit recipient-authorization relation, the two properties central to our threat model.

2.2.2 Multi-Agent Leakage Measurement. AgentLeak [30] shows inter-agent messages and shared memory leak the protected attribute at 2.1 $\times$ the external output’s rate, output-only auditing misses 41.7% of violations, and blanket redaction drops task success by 30–65 points. However, its evaluation uses one fixed two-agent topology. The authors established that internal channels leak, but not how communication structure affects where a disclosed value subsequently reaches.

Liu et al. present MAMA [18], showing that denser, more centralized topologies and shorter attacker-target distance increase PII memory leakage across six graphs, the closest prior evidence

that topology affects multi-agent privacy. Two differences remain: MAMA’s threat model actively seeds and extracts memory over repeated probing, while we measure organic propagation during benign execution; and MAMA reports leakage as a function of graph distance and centrality, while we define it relative to an authorization relation over (value, recipient, channel) triples. So the question is not whether a value reaches a farther node but whether that node was authorized to receive it. Biswas et al.’s topology-depth finding [5] is consistent from an information-theoretic angle but, like MAMA, is not tied to an authorization boundary or a recipient-conditioned defense. AgentBench [19] and AgentDojo [9] evaluate task competence and prompt-injection robustness, not internal-channel exposure. OPENAGENTS SAFETY [27] evaluates privacy alongside other harms in a static benchmark, not a topology-conditioned measurement. To our knowledge, no prior measurement work isolates topology as an independent variable while holding scenario, detector, and policy fixed, as our RQ1 requires.

2.2.3 DLP and Privacy-Preserving Transformation. Preempt [6] sanitizes a single prompt with FPE for format-dependent tokens and mLDP for value-dependent ones, with a formal indistinguishability guarantee, but for one outbound prompt and its response, with no second, differently authorized recipient. RECONS reuses these mechanisms per channel but adds a recipient-aware decision layer that Preempt’s single-agent setting does not require: not only how to transform a protected value, but whether a given recipient requires it transformed at all. Multi-Agent Firewall [8]

Table 1: Comparison with the closest prior measurement and defense work. A benign threat model measures organic propagation; propagation control means the mechanism acts on where a value is headed, not only whether it is sensitive.

	Topology varied	Recipient-conditioned	Benign threat model	Propagation control
AgentLeak [30]	✗	✗	✓	✗
MAMA [18]	✓	✗	✗	✗
Biswas et al. [5]	✓	✗	✗	✗
PrivacyChecker [28]	✗	✗ [†]	✓	✗
1-2-3-Check [16]	✗	✗	✓	✗
FIDES [7]	✗	✗	✗	✓ [‡]
Preempt [6]	✗	✗	✓	✗
This paper	✓	✓	✓	✓

[†]Reasons over sender/recipient/subject but enforces via the acting model’s own prompted judgment. [‡]Controls propagation via label noninterference under an adversarial prompt-injection threat model.

intercepts outbound traffic with deterministic and NER-based detectors before a third-party provider, while AgentGuard [20] enforces attribute-based access control on tool calls; both decide whether content crosses a perimeter, not how a value already inside a multi-agent system should be treated differently by an internal recipient. Our sanitizer targets that internal gap, motivated by AgentLeak’s finding that internal channels leak at $2.1\times$ the external rate.

2.2.4 Context- and Recipient-Aware Enforcement. FIDES [7] enforces noninterference via dynamic label propagation, blocking all policy violations on AgentDojo [9] with fewer task losses than naive taint tracking, but targets an adversarial threat (indirect prompt injection). We study benign disclosure, defining authorization over observable triples rather than label lattices, trading formal guarantees for applicability to black-box traces. Sentinel [10] monitors a shared conversational space and quarantines violations, but is only a proof-of-concept study without an authorization taxonomy or utility trade-off. PrivacyChecker’s sender, recipient, subject, and transmission principle is the closest reasoning-based analog. Still, judgment is made by prompting the acting model itself, inheriting the judgment-action gap it was meant to diagnose rather than using a frozen decision layer. Ling et al. [17] find that persistent-state and propagation threats remain underserved relative to prompt-injection and tool-hijacking, the imbalance this paper corrects.

Table 1 summarizes the comparison. AgentLeak shows internal channels leak and redaction is costly under one fixed topology; MAMA and Biswas et al. show topology affects leakage magnitude under adversarial extraction, without recipient authorization. PrivacyChecker, 1-2-3-Check, and FIDES condition enforcement on some context, sender or recipient judgment, decomposed reasoning, or label flow, but none on the topology-dependent recipient for benign disclosures. Preempt and the DLP systems in §2.2.3 provide transformation and perimeter enforcement, not internal propagation control. This paper extends topology-conditioned propagation beyond AgentLeak’s single topology and adds the authorization-aware decision layer that single-agent and perimeter systems lack.

3 Threat Model and Problem Formulation

3.1 Observable Channels and Multi-Agent Execution

We investigate privacy leakage in MALS as it manifests on *observable channels*, points where information leaves an agent’s private context and becomes available to other entities. Following AgentLeak [30], we consider seven channels, complete for our threat model: C1, final output to the user; C2, inter-agent messaging; C3 and C4, arguments supplied to a tool call and the result returned; C5, shared memory reads and writes; C6, visible reasoning and execution logs; and C7, artifacts the system produces. These differ in how a protected value is represented, which determines what a detector can act on. C1, C2, and C6 are *free-text*, where values appear embedded in prose, often paraphrased rather than quoted, since a delegation message retells its originating task in the model’s own words. C3 and C5 are *structured*, placing a value in a discrete typed field, while C4 and C7 are *mixed*. Only C2 and C5 reads additionally carry an addressee that the trace identifies; the rest expose a value without naming a recipient, which later bounds recipient-aware enforcement (§6.4).

We represent a single observable event as a tuple $\vartheta_t = (\tilde{a}, \zeta, c, r, pos, t)$, where $\tilde{a}$ is the protected attribute, ζ its value type (§3.5), $c \in \{C1, \dots, C7\}$ the channel, r the recipient when resolvable, t the temporal index, and $pos \in \{pos_i, pos_j\}$ records whether r is one of the scenario’s original, presumed-authorized agents or an agent the topology introduces (§3.4).

3.2 Authorization Semantics

Whether an occurrence of $\tilde{a}$ on channel c is a violation depends not only on sensitivity but on *who* receives it. We formalize this as an authorization relation $\mathcal{A}(\tilde{a}, \zeta, c, r, pos) \in \{\text{ALLOW}, \text{SANITIZE}\}$, an *externally specified ground-truth relation* derived from each scenario’s own role and access-control definitions. RECONS’s decision layer (§6) is evaluated against $\mathcal{A}$ as an oracle; an occurrence with $\mathcal{A}(\cdot) = \text{SANITIZE}$ is an *authorization-boundary crossing*.

Conditioning $\mathcal{A}$ on r and pos , rather than $\tilde{a}$ and c alone, is the formal distinction from prior models: AgentLeak [30] defines leakage independently of recipient identity, and Preempt [6] assumes a single user and model owner, so neither captures recipient-specific authorization, a value authorized for one recipient is not thereby authorized for a different, topology-reachable one. RECONS enforces this directly: $\mathcal{A}(\tilde{a}, \zeta, c, r, pos_j) = \text{SANITIZE}$. Possession by an authorized agent is not a violation; a violation occurs when topology exposes the value to a reachable recipient unauthorized for that field.

3.3 From Boundary Crossing to Disclosure

The authorization function determines a crossing but does not itself imply disclosure, which we define as the subsequent observation of $\tilde{a}$ in channel c ’s content at an unauthorized recipient. Leakage requires both. Without RECONS (§5.1), no transformation intervenes, so every crossing exposing $\tilde{a}$ is realized as disclosure and counted as leakage. With RECONS (§6), a crossing counts as leakage only when transformation fails to prevent recovery of $\tilde{a}$ from c , separating whether a recipient is authorized from whether the value is

Table 2: Protected-value taxonomy, defined by admitted transformation.

Value type τ	Sensitive by	Admitted transformation
Format-dependent	recognizable format	format-preserving encryption
Value-dependent	the information carried	pseudonymization / mLDP
Contextual	meaning in context	semantic rewriting / redaction

recoverable once received: an authorized recipient observing $\tilde{a}$ is not a violation, an unauthorized one who cannot recover it after transformation is a crossing without leakage, and one who can recover it is leakage.

3.4 Topology Model

We model a topology as $T = (\check{S}, k)$, where $\check{S}$ specifies the coordination pattern and k the agent count, across six cells: star (CW-2, CW-7), P2P (P2P-4, P2P-7, P2P-10), and hierarchical (Hier-7). CW-2 is the reference cell: with only the original two agents, no pos_j recipient can exist, so exposure is 0% by construction. It sits on neither axis, anchors the design to the original scenarios and cross-framework baseline, and is the baseline the rest are evaluated against in §5.1. The cells impose no total ordering over complexity; agent count does not monotonically increase exposure, and relative leakage among non-reference topologies depends on framework and generator (§5.8–§5.7). Our claim is limited to the structural distinction: every non-reference topology admits pos_j recipients, creating disclosure paths CW-2 lacks. Non-reference cells are constructed by re-rolling the same scenarios without expanding authorization policies; we analyze effects through paired within-scenario comparisons.

The **structure axis** {CW-7, P2P-7, Hier-7} holds $N = 7$ and varies wiring: CW-7 is a star with one coordinator and six workers, P2P-7 is fully connected, and Hier-7 is a two-level hierarchy of one root, two sub-coordinators, and four workers. The **scale axis** {P2P-4, P2P-7, P2P-10} holds peer-to-peer wiring fixed and varies $N \in \{4, 7, 10\}$; the axes share P2P-7 as both a structure comparison and a middle-scale point (Figure 1).

3.5 Protected-Value Taxonomy

We classify protected values into three classes by applicable transformation (Table 2). Format-dependent values (identifiers, account numbers, dates of birth) are recognizable by structure and support format-preserving transformation. Value-dependent values (names, addresses, numeric amounts) are sensitive by content and support pseudonymization or mLDP. Contextual values (diagnoses, legal strategies, internal assessments) derive sensitivity from meaning and require semantic rewriting or redaction. This extends $\text{Pr} \in \text{empt}$ ’s two-class distinction [6] with a contextual class arising where sensitive information sits in free-form agent-to-agent text rather than a discrete field; the three classes determine which transformation applies (§6).

3.6 Adversary Model

We model a *passive unauthorized recipient* observing one or more instrumented channels C_1 – C_7 without authorization for the value under $\mathcal{A}$, canonically a $pos = pos_j$ agent introduced by the topology;

an observer of shared logs (C_6) or persistent artifacts (C_7) is also in scope when unauthorized for that field. The adversary does not modify the execution, inject content, or obtain information beyond channels the scenario exposes, and we assume no computational limit on interpreting observed content.

We consider three adversary levels [30]: A_0 , benign execution; A_1 , an external adversary targeting C_1 ; and A_2 , an internal adversary targeting C_2 – C_5 . The primary threat model excludes inference from indirect evidence, collusion among unauthorized recipients, cross-session correlation, and observation outside C_1 – C_7 ; these stronger capabilities are evaluated separately in §5.3–§5.4 and excluded from the primary leakage measurements in §5.1 and sanitizer evaluation in §6.

4 Experimental Methodology

Our experiments vary topology, adversary level, protected-value class, observable channel, and execution horizon. Unless stated otherwise, the generator and framework are fixed to Qwen3-32B and LangGraph, isolating topology as the independent variable.

4.1 Corpus Construction and Evaluation Scale

We derive our corpus from the public 1,000-scenario AgentLeak benchmark [30], which provides tasks, protected values, agent roles, and release policies. We construct long-horizon executions at $T=20$; 22 scenarios yield no valid neighboring pair and are excluded, leaving 978. We then retain only *true multi-agent* scenarios, at least two agents, two distinct roles, and observable inter-agent coordination through C_2 edges, yielding 485: 138 corporate, 100 finance, 100 healthcare, 147 legal. Each scenario’s vault contains multiple protected fields alongside a field-level release policy specifying `allowed_set` and `forbidden_fields`. We designate a single protected attribute P per scenario as the tracked target for paired-trajectory construction, drawn from `forbidden_fields`, with unspecified fields treated as forbidden by default.

For each scenario, we construct a neighboring pair (τ, τ') differing only in P , evaluated at horizons $T \in \{1, 3, 5, 10, 20\}$, fixed within each scenario and held constant across topology and adversary-level conditions to preserve the corpus’s horizon distribution. Each of the 485 scenarios is evaluated under six topology cells and three adversary levels, yielding $485 \times 6 \times 3 = 8,730$ execution records, each containing the pair and seven observable channels. Detection runs independently per channel and pair member, yielding $7 \times 2 = 14$ instances per record and $8,730 \times 14 = 122,220$ total (Table 3), the unit of channel-level evaluation from which record- and topology-level statistics are computed.

Trace normalization. We execute each scenario through a custom-built framework-agnostic adapter normalizing traces from all eight orchestration frameworks into a common JSON Lines representation, exposing events on C_1 – C_7 and recording execution and information-flow graphs for attribution and integrity checks. All executions use Qwen3-32B, served locally with vLLM [13], at temperature 0 with fixed seeds. Tool-call inputs and outputs (C_3/C_4) come from a deterministic tool layer rather than live services, so paired executions and topology conditions differ only in protected-value substitution and communication structure.

Table 3: Stratification of the 122,220 evaluation instances.

Stratum	Resolving mechanism	Instances	Share
Certain-positive	Tier 1/2 exact or structured match	70,280	57.5%
Semantic-middle	Tier 3 judge	36,568	29.9%
Certain-negative	No deterministic signal	15,372	12.6%
Total		122,220	100.0%

4.2 Baseline Detection Pipeline

We adopt AgentLeak’s three-tier detection methodology [30], adapted to our multi-agent traces, applied independently per channel to ask whether P is disclosed, verbatim or via a semantically equivalent expression. Authorization is evaluated separately via the recipient-conditioned policy (§3).

- (1) **Canary and exact matching (Tier 1).** Deterministically detects verbatim occurrences, checking observable-content capture.
- (2) **Structured-field matching (Tier 2).** Deterministic recognizers identify SSNs, Luhn-valid cards, and ICD-10 codes; checksum-validated matches are high-confidence, pattern-only matches are flagged but not counted, and matches are restricted to the target field and its release policy.
- (3) **LLM-as-judge (Tier 3).** Resolves remaining instances by evaluating semantic equivalence between channel content and the protected value, using Mistral-Small-24B as judge and Qwen3-32B as generator; a score in $[0, 1]$ classifies disclosure when at least the calibrated threshold ζ (§4.5).

Each disclosure is attributed to the first detecting tier, in order: Tier 1, Tier 2, and Tier 3, preventing duplicate counting and separating exact, structured, and semantic detections.

4.3 Detection Stratification

We apply the pipeline to all 122,220 instances. A deterministic pre-filter partitions them into three strata (Table 3): *certain-positive*, resolved by Tier 1 or 2; *certain-negative*, with no protected value introduced into the channel’s execution context; and *semantic-middle*, requiring Tier 3 assessment. Tier 3 performance is reported only on the semantic-middle stratum, since combining it with the deterministic strata would conflate exact matching with semantic-judge performance.

4.4 Calibration Corpus and Ground Truth

We calibrate the Tier-3 threshold on a dedicated corpus of 720 traces generated under the CW-2 topology and A_0 attack level, containing 10,080 detection instances across two neighboring-pair members and seven channels. A deterministic pass assigns each instance to the three strata defined in Table 3; only the semantic-middle stratum is submitted for human annotation. Two domain-expert annotators independently label a stratified 300-instance sample, each instance receiving LEAK, NO-LEAK, or UNSURE. Disagreements are adjudicated against a written guideline evaluating disclosure of the designated attribute rather than unrelated sensitive content. After adjudication, the sample yields 32 leaks and 267 non-leaks among 299 resolved instances. Inter-annotator agreement on raw pre-adjudication labels is substantial, with Cohen’s $\kappa = 0.835$ three-way and 0.846 binary at 95.7% raw agreement, where the three-way

Table 4: Tier-3 threshold calibration on the main semantic-middle sample (299 instances, 32 leaks).

ζ	FPR	FNR	Accuracy	Precision	Recall	F1
0.35–0.60	0.0112	0.0625	0.9833	0.9091	0.9375	0.9231
0.61–0.70	0.0112	0.0937	0.9799	0.9062	0.9062	0.9062
0.71–0.80	0.0112	0.1875	0.9699	0.8965	0.8125	0.8525
0.81–0.85	0.0112	0.2187	0.9665	0.8926	0.7812	0.8333
0.86–0.90	0.0075	0.2500	0.9665	0.9231	0.7500	0.8276
0.91–0.95	0.0037	0.3750	0.9565	0.9524	0.6250	0.7547

figure includes UNSURE and the binary figure excludes any instance either annotator so marked, confirming the calibration labels are reliable.

4.5 Threshold Calibration

We calibrate ζ on the adjudicated semantic-middle sample. The Tier-3 judge operates at temperature 0, producing a deterministic score. We sweep $\zeta \in [0.35, 0.95]$ in steps of 0.01, classifying an instance as disclosed when its score is $\geq \zeta$, and selecting the threshold that maximizes $F1$ subject to $FPR < 0.05$. The FPR constraint holds throughout, with a maximum of 0.0112, so selection is determined entirely by $F1$, which is maximized at 0.9231 across $\zeta \in [0.35, 0.60]$, an interval containing no calibration score, leaving classification unchanged throughout. We report $\zeta = 0.50$ as the representative operating point: precision 0.9091, recall 0.9375, FPR 0.0112, $F1$ 0.9231, values characterizing the semantic-middle population rather than aggregate corpus performance. AgentLeak’s [30] threshold, $\zeta = 0.72$, yields $F1 = 0.8525$ on our sample versus 0.9231 at $\zeta = 0.50$, so we recalibrate rather than transfer it. As a robustness check, Gemma-3-27B, a different model family, independently scores the labeled sample; at $\zeta = 0.50$ the two judges agree on 96.7% of instances, Cohen’s $\kappa = 0.846$. Gemma-3-27B is used only for this cross-check and does not affect threshold selection or reported counts.

5 Evaluation

5.1 Topology-Dependent Leakage

5.1.1 Experimental Design. We evaluate the frozen held-out corpus under all six topology cells and three adversary levels, A_0, A_1, A_2 , with detector, judge, prompts, policy, and pipeline fixed throughout (§4). Our primary outcome is topology-conditioned new-agent disclosure, $L(\tilde{T}) = P(\text{unauthorized disclosure} \mid \tilde{T})$, where $\tilde{T}$ denotes topology. We report three metrics:

- (1) **Surface leakage:** the proportion of scenarios with at least one unauthorized disclosure in any channel.
- (2) **New-agent disclosure:** the proportion reaching a topology-introduced recipient ($pos = pos_j$), our primary measure of topology-dependent exposure.
- (3) **Unauthorized propagation:** the proportion reaching any recipient outside the authorized set regardless of position.

This distinction lets us determine whether topology changes over all disclosure incidence or primarily where disclosed information propagates.

5.1.2 Topology-Conditioned Exposure. Surface leakage changes little across topologies, remaining between 60.6% and 62.8% across

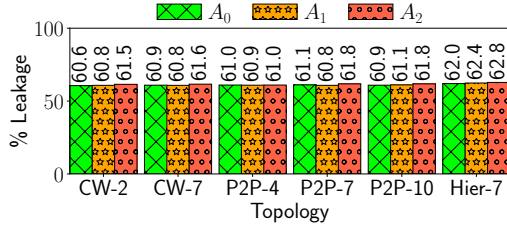


Figure 2: Surface leakage rate (%) by topology and adversary level. Each cell contains the proportion of the 485 scenarios exhibiting at least one unauthorized disclosure.

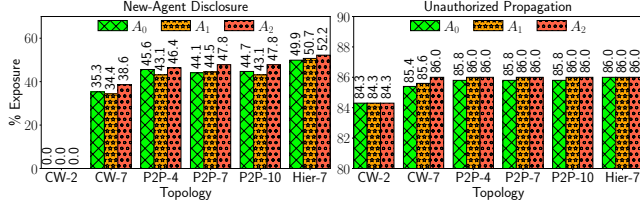


Figure 3: Recipient-conditioned exposure (%) by topology and adversary level. New-agent disclosure is the primary topology-conditioned leakage measure $L(T)$. Unauthorized propagation includes any recipient outside the authorized set.

Table 5: Cochran’s Q tests for topology and adversary-intensity effects on recipient-based disclosure.

Effect	Metric	χ^2	df	p
Topology	New-agent	191.8	4	2×10^{-40}
Topology	Propagation	94.2	5	9×10^{-19}
Adversary	New-agent	16.0	2	3×10^{-4}
Adversary	Propagation	9.3	2	9×10^{-3}

adversary levels (Figure 2), while recipient-conditioned exposure increases substantially: new-agent disclosure is 0% at CW-2 by construction, 34.4–38.6% at CW-7, 43.1–47.8% across the P2P cells, and 49.9–52.2% at Hier-7 (Figure 3). Unauthorized propagation stays high throughout, from 84.3% at CW-2 to at most 86.0% in richer topologies.

At A_0 , the Wilson 95% confidence intervals for new-agent disclosure are 35.3% [31.1, 39.6] for CW-7, 45.6% [41.2, 50.0] for P2P-4, 44.1% [39.8, 48.6] for P2P-7, 44.7% [40.4, 49.2] for P2P-10, and 49.9% [45.5, 54.3] for Hier-7. The corresponding interval for CW-2 is [0.0, 0.8]%. We use Cochran’s Q test for formal inference because each scenario is evaluated repeatedly across topology conditions (Table 5). CW-2 is excluded from the topology test for new-agent disclosure because its outcome is structurally constant at zero. All four effects are significant at conventional thresholds (Table 5). Topology has a larger effect on new-agent disclosure ($\chi^2 = 191.8$, $df = 4$, $p = 2 \times 10^{-40}$) than on unauthorized propagation ($\chi^2 = 94.2$, $df = 5$, $p = 9 \times 10^{-19}$): topology moves new-agent disclosure over a wide range, 0% to 52.2%, but propagation only within a narrow 1.7-point band, 84.3% to 86.0%, so a smaller absolute range still registers as a strong effect once repeated-measures structure is accounted for. Adversary intensity is also significant but with much smaller

Table 6: Cumulative disclosure: mean number of distinct protected values disclosed per trajectory across nested horizons. Values are judge-scored and pooled over τ/τ' ($n = 970$).

Scope	T_1	T_3	T_5	T_{10}	T_{20}
Behavioral ($C_1/C_2/C_6/C_7$)	0.020	0.213	0.230	0.250	0.321
All channels (C_1-C_7)	0.020	0.237	0.251	0.281	0.349
Tool only (C_3/C_5)	0.000	0.057	0.057	0.081	0.089

statistics, 16.0 and 9.3 against 191.8 and 94.2, making topology the dominant driver here and adversary intensity secondary.

Hence, the effect concentrates in recipient-conditioned exposure: surface leakage varies by only 2.2 points across cells, while new-agent disclosure rises from a structural 0% at CW-2 to 52.2% at Hier-7 under A_2 . Richer topologies form no strict monotonic ordering; P2P-4 exceeds P2P-7 at A_0 , so results reflect not agent count but communication structure changing which recipients become exposed to already-disclosed information. These disclosures occur through C_2 rather than the deterministically injected structured channels, consistent with topology-dependent propagation rather than an artifact of structured-value survival. Overall, this result shows that topology primarily changes the reachable recipient set: richer coordination’s principal security consequence is increased unauthorized reach, even as surface disclosure incidence stays nearly unchanged.

5.2 Compositional Leakage

We investigate whether disclosure accumulates as an adversary observes progressively longer trajectories, over nested horizons $T_1 \subset T_3 \subset T_5 \subset T_{10} \subset T_{20}$, distinguishing *cumulative disclosure*, distinct protected values disclosed by horizon T_k , from *marginal disclosure*, those first disclosed between consecutive horizons, and the primary measure of compositional leakage: sustained disclosure at later horizons would indicate accumulation, concentration at early horizons front-loading. We evaluate behavioral channels ($C_1/C_2/C_6/C_7$), all channels (C_1-C_7), and tool channels (C_3/C_5), including only agent-authored events. Each turn beyond the opening task is scripted as one of eight *interaction archetypes* (Table 25), reflecting a distinct conversational function.

Disclosure is predominantly front-loaded. Cumulative disclosure increases with horizon, but most first-time disclosures occur at the first transition (Table 6). For behavioral channels, mean cumulative disclosure rises from 0.020 at T_1 to 0.321 at T_{20} ; of 311 first-time disclosures across 970 trajectories, 188 (60.5%) occur at $T_1 \rightarrow T_3$, versus 16 at $T_3 \rightarrow T_5$, 19 at $T_5 \rightarrow T_{10}$, and 69 at $T_{10} \rightarrow T_{20}$ (Table 7). The same pattern holds across all channels and tool channels: 62.2% and 64.0% of first-time disclosures, respectively, occurring at $T_1 \rightarrow T_3$: longer horizons increase cumulative exposure, but later turns add little new disclosure, and we find no evidence of sustained accumulation.

Confirmation probes dominate disclosure. Confirmation probes (archetype G) account for 574 of 2,212 disclosing turns across all channels, a 25.9% rate (Table 8), versus 0.9% to 2.7% for other archetypes, the dominant disclosure context though not exclusive: 14.7% of first-time disclosures occur outside it, in handoffs, elaboration, justification, and closeout turns.

Table 7: Marginal first-time disclosures by horizon transition. Counts denote trajectories with a newly disclosed protected value among 970 trajectories; re-disclosures are excluded.

Scope	T_1	$T_1 \rightarrow T_3$	$T_3 \rightarrow T_5$	$T_5 \rightarrow T_{10}$	$T_{10} \rightarrow T_{20}$
Behavioral ($C_1/C_2/C_6/C_7$)	19	188	16	19	69
All channels (C_1-C_7)	19	211	13	30	66
Tool only (C_3/C_5)	0	55	0	24	7

Table 8: Disclosure by interaction archetype in the all-channel scope. Rate is the fraction of turns of each archetype that disclose the protected value.

Archetype	Disclosures	Turns	Rate
G (confirmation probe)	574	2,212	0.259
F (rapport)	78	2,916	0.027
B (handoff)	97	4,101	0.024
E (closeout)	62	2,643	0.023
C (justification)	67	3,761	0.018
BASE (initial task)	19	1,777	0.011
D (error handling)	36	3,699	0.010
A (elaboration)	63	6,646	0.009

Tool channels provide an independent leakage surface. Tool channels produce fewer disclosures than behavioral ones but expose a distinct surface (Table 6 and Table 7): mean T_{20} disclosure through C_3/C_5 is 0.089 versus 0.321 behaviorally, and is similarly front-loaded, 55 of 86 first-time tool disclosures occurring at $T_1 \rightarrow T_3$, meaning communication-only auditing would miss disclosures in tool-call arguments and memory writes.

5.3 Reconstruction: No Confirmed Emergent Recovery

Direct disclosure propagates once emitted (§5.1–5.2); we now test the more serious threat of *reconstruction*, an adversary recovering P from evidence it was never explicitly given.

Inference. Whether P is recoverable from contextual cues within a single non-disclosing channel, across 2,954 eligible instances in four domains, yields no confirmed reconstruction, 0/2,954 (Table 9). The judge produced 22 raw positives, all rejected on audit, attributable to decoy-value selection (14), field-type confusion (6), and generic topical over-attribution (2).

Cross-channel. Whether two independently non-disclosing channels jointly reconstruct P , across 3,195 eligible pairs in all four domains, confirms no cross-channel linkage, 0/3,195 (Table 9). 25 raw positives (0.78%) were rejected on audit; typically, two channels independently contained the same incorrect candidate value rather than jointly recovering the true one.

Cross-agent. Whether pooling two agents’ views reconstructs P when neither individually possesses it, over all 172,949 eligible pairs across six topology cells, yields 0 confirmed reconstructions (Table 9). 503 raw positives (0.29%) were rejected: 443 showed no evidence of the true value, and 60 contained it verbatim in a single agent’s view, possession rather than linkage.

Since emergent recovery is zero on every axis, cumulative recoverability reduces to the single-source rate; we do not report this separately. A complementary turn-level view (§5.2) shows first-time

Table 9: Reconstruction results by aggregation axis. All positives were manually audited; emergent recovery is zero on every axis.

Axis	Eligible	Raw positives	Single_src	Audited
Inference	2,954	22	39.5%	0
Cross-channel	3,195	25	33.0%	0
Cross-agent	172,949	503	87.2%	0

Table 10: Pairwise distinguishability by horizon. The T_5 signal does not survive the confound analysis below.

Horizon	Real accuracy	Real advantage	Shuffled advantage
T_1	0.502	+0.002	−0.000
T_3	0.531	+0.031	−0.008
T_5	0.566	+0.066	−0.002
T_{10}	0.537	+0.037	−0.000
T_{20}	0.500	+0.000	−0.004

disclosures are front-loaded rather than accumulating, 211 of 339 (62%) occurring at the first transition, $T_1 \rightarrow T_3$, describing timing of direct disclosure, not reconstruction. Across 179,098 exhaustively evaluated instances and pairs, every raw positive manually audited, we find no confirmed evidence a withheld attribute is reconstructed from non-disclosing content, alone or combined, under the evaluated corpus and protocol, though this does not establish absence under other corpora or strategies.

5.4 Behavioral Leakage

A distinct threat is P influencing observable behavior without being disclosed. We measure this over 33 structural features per trajectory encoding nothing of P : pairwise distinguishability between neighboring executions, and population-level prediction of P ’s semantic category.

5.4.1 Pair Distinguishability.

Method. We identify 244 matched pairs (τ, τ') differing only in P , with no judge-confirmed disclosure on either side (488 trajectories); a logistic-regression distinguisher predicts τ versus τ' from behavioral features under scenario-grouped cross-validation, with distinguishing advantage as accuracy − 0.5 against a shuffled-label null per horizon.

Results. The advantage is transient, not monotonic, near-random at T_1 (50.2%), peaking at T_5 (56.6%), near-random again at T_{20} (50.0%). The T_5 result is statistically distinguishable from the null ($z \approx 12.8$), though distinguishability alone does not establish a P -dependent side-channel, tested directly below.

Confound analysis. Confound analysis attributes the T_5 signal to corpus construction, not P , on three grounds. A genuine fingerprint should not favor one pair member arbitrarily assigned; four of the eleven strongest T_5 features show directional bias ($|t| \geq 2$, e.g. `tool_calls_archetype_A` favoring τ in 8/244 pairs and τ' in none). Generation order is perfectly confounded with pair label, τ always generated before τ' (corr = 1.000), leaving no within-pair variation to isolate an order effect. And the protected value itself does not explain the difference: value-length and response-length correlate at only $r = 0.06$, yet equal-length values (e.g. 63000 vs. 75267) still produce ~ 100 -character response differences.

Table 11: Order-invariant control: real matched pairs vs. synthetic re-pairings.

Horizon	Real advantage	Shuffled null
T_1	+0.367	−0.018
T_3	+0.346	−0.008
T_5	+0.342	−0.015
T_{10}	+0.338	−0.007
T_{20}	+0.207	−0.011

As an order-invariant alternative, comparing true pairs against domain-constrained synthetic re-pairings yields larger advantages (0.21–0.37, Table 11) than the original 0.066, with all 18 non-trivial features showing smaller within-pair differences for real than synthetic pairs (ratios 0.13–0.69), consistent with scenario identity and task similarity rather than a targeted fingerprint. A definitive test would require randomized generation order or repeated generations per scenario; without that, we treat T_5 as confounded, not behavioral leakage.

5.5 Synthesis: Leakage Is Primarily Direct and Propagative.

Explicit disclosure propagates across agents, channels, and turns; all three reconstruction axes yield zero confirmed emergent recovery across 179,098 exhaustively evaluated instances and pairs, every raw positive manually audited, and behavioral analysis finds no persistent value-level fingerprint. The dominant leakage pathway is thus direct disclosure followed by propagation: richer topology expands the recipient set for an already-disclosed value, and longer executions add further disclosure opportunities. These findings motivate defenses constraining initial disclosure and subsequent propagation (§6).

5.6 Cross-Framework Leakage Measurement

This experiment evaluates whether orchestration framework choice affects privacy leakage under an identical corpus, generator, authorization policy, and detection pipeline.

5.6.1 Experimental Design. We evaluate the same 485 scenarios across eight frameworks: AutoGen [29], CrewAI [2], Google ADK [1], LangChain [14], LangGraph [15], LlamaIndex [23], OpenAI Agents SDK [3], and Semantic Kernel [24], each scenario runs as an attribute-neighboring pair (τ, τ') , yielding $485 \times 8 \times 2 = 7,760$ executions. All use the same generator (Qwen3-32B via vLLM), deterministic decoding, identical prompts, tools, and attribute assignments, with events collected over C_1 – C_7 and normalized into the common trace schema introduced in §4.1, totaling 214,853 assessable channel events across frameworks. Each framework runs through its native coordination abstraction, delegation, handoffs, workflows, or group chat, rather than a framework-independent layer, preserving the communication behavior that is the object of comparison.

5.6.2 Detection and Authorization. Leakage detection uses the frozen three-tier pipeline (§4.2); the judge is identical across frameworks and never evaluates its own model family’s outputs. Detection and authorization are evaluated separately: the judge decides

Table 12: Leakage and value fidelity across eight orchestration frameworks, sorted by policy-filtered leakage.

Framework	Raw leakage	Policy-filtered leakage	Value fidelity
AutoGen	43.85%	43.38%	87.15%
LangChain	25.54%	25.50%	25.59%
LangGraph	23.75%	23.71%	26.40%
CrewAI	21.61%	21.01%	27.75%
Semantic Kernel	1.56%	0.67%	67.11%
LlamaIndex	0.72%	0.50%	36.23%
OpenAI Agents SDK	1.56%	0.48%	42.55%
Google ADK	0.80%	0.29%	47.06%
Overall	21.63%	21.44%	48.47%

only whether the attribute appears, and authorization is applied afterward against the field-level release policy $\mathcal{A}(\check{a}, \zeta, c, r, pos)$ (§3). We report: (1) *Raw leakage*: all semantic detections; (2) *Policy-filtered leakage*: detections excluding disclosures explicitly authorized by the scenario policy. The latter corresponds to the contextual-integrity violation used for all comparisons below.

5.6.3 Value Fidelity Verification. Since detection does not verify the disclosed value is correct, we measure value fidelity independently for canonical attributes (identifiers, phone numbers, dates of birth, account numbers, canary tokens, names), comparing generated values against ground-truth values after canonicalization, classifying each as *exact*, *corrupted*, or *substituted*, reported independently to rule out framework rankings explained by confabulation.

5.6.4 Framework Leakage Rates. Table 12 reports raw leakage, policy-filtered leakage, and value fidelity per framework, sorted by policy-filtered leakage.

The frameworks separate into two groups: AutoGen, LangChain, LangGraph, and CrewAI between 21.0% and 43.4% filtered leakage; Semantic Kernel, LlamaIndex, OpenAI Agents SDK, and Google ADK below 0.7%. The gap between groups, CrewAI 21.01% to Semantic Kernel 0.67%, is roughly 31×, and the overall extremes, AutoGen 43.38% versus Google ADK 0.29%, roughly 150×. This two-group separation is the reliable finding, not the ordering *within* each group: §5.6.6 establishes a cross-corpus noise floor no larger than 0.41 points, and every low-group gap, 0.02–0.19pp, falls well below it, so those four should be read as tied. The high-group’s internal gaps, 1.79–17.88pp, exceed the floor by an order of magnitude and are reliably ordered. The authorization filter changes overall leakage by only 0.19 points, and the two-group separation is unchanged after filtering.

As shown in Figure 4, leakage concentrates in non-user-facing channels: tool inputs (C_3) and outputs (C_4) exhibit the highest rates, followed by artifacts (C_7), inter-agent communication (C_2), and shared memory (C_5), while user-facing outputs (C_1) account for substantially fewer disclosures, showing that evaluating only final responses understates privacy exposure in MALS.

5.6.5 Value Fidelity Is Independent of Leakage. Leakage frequency and value fidelity measure different properties. AutoGen leaks most frequently while reproducing values most accurately (87.15%); LlamaIndex leaks rarely but correctly only 36.23% of verifiable cases,

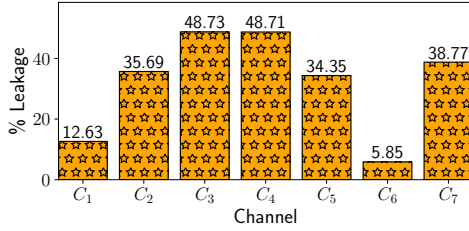


Figure 4: Leakage rates across observable channels on all frameworks.

Table 13: Policy-filtered leakage (%) across four independently generated corpora. Deltas are relative to the temperature-0.0 corpus of record.

Framework	Temp. 0.0	Temp. 0.2 (Δ)	Temp. 0.5 (Δ)	Temp. 0.7 (Δ)
AutoGen	43.38	43.34 (−0.04)	43.41 (+0.03)	43.25 (−0.13)
LangChain	25.50	25.59 (+0.09)	25.69 (+0.19)	25.87 (+0.37)
LangGraph	23.71	23.79 (+0.08)	23.90 (+0.19)	23.93 (+0.22)
CrewAI	21.01	21.40 (+0.39)	21.04 (+0.03)	21.41 (+0.40)
Semantic Kernel	0.67	0.82 (+0.15)	0.82 (+0.15)	0.84 (+0.17)
LlamaIndex	0.50	0.45 (−0.05)	0.49 (−0.01)	0.41 (−0.09)
OpenAI Agents SDK	0.48	0.50 (+0.02)	0.54 (+0.06)	0.52 (+0.04)
Google ADK	0.29	0.37 (+0.08)	0.32 (+0.03)	0.40 (+0.11)
Overall	21.44	21.53 (+0.09)	21.59 (+0.15)	21.67 (+0.23)

so low leakage can reflect avoided disclosure or incorrect and substituted values, fidelity requiring independent measurement.

To rule out confabulation as the source of framework ranking, we joined per-instance fidelity with leakage decisions: exact-value disclosures were almost always detected, substituted values were never classified as valid, and the two-group separation persists when restricted to exact-value disclosures, so the grouping is not explained by confabulation.

5.6.6 Robustness Across Generation Regimes. We evaluate robustness using three additional corpora from the identical scenario pool and configurations under temperatures 0.2, 0.5, and 0.7, each unseeded, comparing policy-filtered leakage against the temperature-0.0 corpus of record (Table 13).

Framework-level leakage is stable across the range: the high- and low-leakage groups remain separated by more than 19.8 points at every temperature, with no framework changing group membership. Maximum per-framework drift from the temperature-0.0 anchor is small throughout (0.39pp at 0.2, 0.21pp at 0.5, 0.41pp at 0.7) but non-monotonic, temperature 0.5 being a local minimum; the framework responsible for the maximum shifts too, CrewAI at 0.2 and 0.7, LangChain at 0.5, since CrewAI’s own drift briefly turns negligible and negative at 0.5, attributed below to CrewAI specifically rather than temperature.

The apparent low-leakage-group instability noted earlier does not hold under the complete four-regime picture. The raw metric gives an identical ordering: Semantic Kernel, OpenAI Agents SDK, Google ADK, LlamaIndex, at all four temperatures, no raw-metric swap ever occurring. The policy-filtered ordering shows one inversion, LlamaIndex above OpenAI Agents SDK, only in the fixed-seed temperature-0.0 corpus; the three unseeded regimes

independently agree on Semantic Kernel, OpenAI Agents SDK, LlamaIndex, Google ADK, so the inversion is best read as specific to that corpus rather than an instability recurring under resampling. All four low-leakage frameworks remain far below the high-leakage group’s own internal noise floor regardless of metric or regime, and should be read as tied rather than ranked.

Under fixed temperature and seed, event content reproduces identically in 83.99% of cases at temperature 0.0, the residual reflecting floating-point non-determinism rather than sampling and not eliminable by seeding; reproducibility does not degrade at higher temperature (89.92% at 0.5, 89.68% at 0.7). CrewAI accounts for nearly all residual non-reproducibility at every temperature (3/16, 3/16, 4/16 mismatched pairs at 0.0, 0.5, 0.7, against 0/16 for LangChain and LangGraph throughout), consistent with driving the non-monotonic drift above; framework-level aggregate rates remain stable regardless, indicating the effect reflects aggregate behavior over hundreds of trials rather than any single generation path. This claim establishes stability under generation variance across temperatures 0.0 through 0.7 only; it does not address scenario-sampling, judge-choice, or threshold-recalibration sensitivity, and the non-monotonic temperature relationship should not be extrapolated beyond the tested range.

5.7 Model Generalization

The topology gradient in §5.1 was initially measured using QWEN3-32B. To assess generalization across generators, we repeat the sweep with three additional models from distinct families, GPT-5-MINI, GEMINI-3.7-FLASH, and LLAMA-3.3-70B, holding framework, corpus, topology, pipeline, and detector fixed across all 485 scenarios and six cells. This separates whether topology-induced exposure occurs from how frequently it does. The former is consistent across generators, whereas the latter varies substantially, with cell ordering unpreserved across models.

5.7.1 Topology-induced exposure generalizes, but its magnitude varies. Topology-induced exposure occurs for every generator (Figure 5). CW-2 yields no detected disclosure for any model, while every richer topology yields a non-zero rate for all four; the smallest, 9.3% for GEMINI-3.7-FLASH, remains separated from CW-2’s confidence bound, so the effect is not specific to QWEN3-32B. The magnitude differs substantially across generators. Over richer cells, GPT-5-MINI ranges 57.1–65.8%, QWEN3-32B 35.3–49.9%, LLAMA-3.3-70B 21.4–23.5%, and GEMINI-3.7-FLASH 9.3–11.5%, so identical nominal structures yield very different rates by generator (Figure 5).

Ordering also varies: QWEN3-32B peaks at Hier-7 (49.9%), while the other three instead rise sharply from CW-2 and then vary narrowly. These results are unsupportive of any model-independent topology-to-rate mapping. The highest-to-lowest richer-cell gap, 65.8% versus 9.3%, roughly sevenfold, is reported descriptively: GPT-5-MINI shows generation variability even at temperature zero, so small differences among its cells are not a definitive ranking. The robust observation is separation between model-level ranges, not internal ordering within any one model.

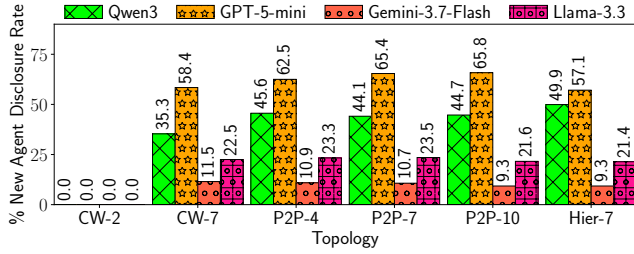


Figure 5: New-agent disclosure rate by topology cell across four generator models under LangGraph ($n = 485$ scenarios per cell). Exposure magnitude and cell ordering vary across generators in richer topology cells.

5.7.2 Model behavior contributes to exposure variation. GEMINI-3.7-FLASH’s comparatively low, flat gradient motivated closer inspection of its Hier-7 executions. A manual recheck of sampled non-disclosing runs found no missed disclosures to topology-introduced agents; judge scores were 0.0 for 87% of examined events, well below $\zeta = 0.50$, and topology construction was verified, giving no evidence of construction failure or a threshold artifact.

Residual exposure instead tracks attribute type (Table 26): structured identifiers (SSN, date of birth, phone, email, account number, salary, balance, credit score) are rarely forwarded (0.0–3.7%), while contextual and narrative attributes reach far higher rates, confidential strategy 27.6%, internal notes 36.4%, criminal history 63.6%, consistent with the flat cross-topology pattern and illustrating why an aggregate rate can conceal which information types actually propagate; we do not interpret this as a universal model property.

5.7.3 Implications for Topology-Aware Privacy Enforcement. The model sweep adds a second generalization dimension beyond the framework comparison in §5.6: nominally equivalent topologies expose differently under different orchestration semantics, and the same topology and framework expose differently across generators. Neither topology nor generator identity alone establishes that a flow is authorized, since one generator’s high exposure under a topology can be another’s low exposure, and a generator suppressing structured identifiers may still propagate contextual ones. RECONS addresses this at the point of detected disclosure rather than by assumption, evaluating realized recipient and execution context before preserving, transforming, or suppressing a value (§6). Since propagation depends on both orchestration and generation, this result reinforces recipient-conditioned enforcement on realized information flow over static assumptions about topology or generator.

5.8 Framework Generalization

Topology-dependent exposure observed earlier (§5.1) may depend on LangGraph’s execution semantics, so we replicate the full sweep in AutoGen, a message-passing framework, holding the generator fixed to QWEN3-32B from the same endpoint and reusing the framework-agnostic pipeline and frozen detector, across all 485 scenarios in each of six cells, 2,910 scenario-cell executions total. This addresses whether topology-introduced exposure persists across frameworks, whether its magnitude is comparable, and whether the topology-specific pattern is preserved: the first is affirmed, the

latter two are not, with exposure occurring in both frameworks but differing in magnitude and distribution across cells.

5.8.1 Cross-Framework Replication.

Exposure is observed in every richer cell in both frameworks (Figure 6). CW-2 is a structural anchor with no topology-introduced recipient, so its rate is 0% in both, a consequence of the metric definition rather than freedom from other disclosures. For richer cells, AutoGen ranges 20.2–37.5% versus LangGraph’s 35.3–49.9%, so the effect is not LangGraph-specific, though magnitude and ordering are not preserved: LangGraph peaks at Hier-7 (49.9%), while AutoGen peaks in the P2P cells (37.5%), with Hier-7 substantially lower (24.7%). Thus, nominal topology does not uniquely determine exposure magnitude; the same topology yields different rates under different orchestration semantics.

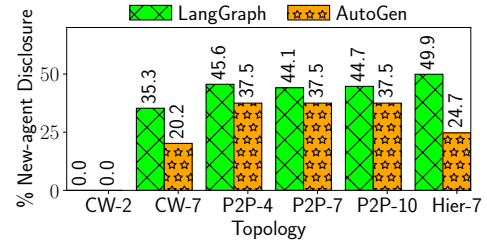


Figure 6: New-agent disclosure rate by topology cell for LangGraph and AutoGen over the full corpus ($n = 485$ per cell), using the same Qwen3-32B generator. Topology-introduced exposure occurs in every richer cell in both frameworks, but its magnitude and topology-specific ordering differ.

5.8.2 Realized-Path Analysis. The cross-framework differences suggest nominal topology does not fully determine realized propagation, so we examine whether the exposure gap tracks the realized communication path, focusing on Hier-7, the largest cross-framework difference (49.9% LangGraph versus 24.7% AutoGen).

For each execution, we identify the disclosing agent as the earliest to access the protected value via a vault channel, recover realized inter-agent edges (C_2) from the trace, and compute minimum graph distance from that agent to each topology-introduced recipient. We partition executions as *direct-access* if at least one such recipient is one hop away, or *relay-required* if every one is at least two hops away, capturing whether reaching them requires intermediate agents. Executions where the value never enters a vault channel have no defined disclosing agent or distance and are excluded from this analysis (retained in cell-level results): LangGraph has 62 such executions, leaving 423 with a defined disclosing agent, while AutoGen has none, leaving all 485. We report these denominators explicitly since the hop analysis is conditional on the value entering execution.

The decomposition shows substantially different patterns (Table 14): AutoGen exposure is 34.0% for direct-access executions versus only 2.8% when relay-required, while LangGraph’s corresponding rates are 55.9% and 60.3%, so relay hops attenuate strongly in AutoGen but not LangGraph. The 95% CI for AutoGen relay-required is [1.1, 7.0]% versus [51.6, 68.4]% for LangGraph, widely separated intervals indicating a substantial conditional-exposure

Table 14: Realized-path decomposition of new-agent disclosure in Hier-7, measured from each framework’s C_2 execution traces. Rates are reported with 95% Wilson confidence intervals (CI). The analysis is conditional on a defined disclosing agent (423 LangGraph executions, 485 AutoGen executions).

Hier-7 executions	AutoGen	LangGraph
Direct-access	34.0% [29.2,39.2], $n=341$	55.9% [50.2,61.4], $n=297$
Relay-required	2.8% [1.1,7.0], $n=144$	60.3% [51.6,68.4], $n=126$
Observed pattern	strong attenuation	no comparable attenuation

Table 15: AutoGen relay-failure rate by topology cell. The P2P cells require no relay.

Topology cell	Relay-failure rate
Hier-7	8.5%
CW-7	12.2%
P2P-4 / P2P-7 / P2P-10	0.0%

difference; LangGraph’s direct-access and relay-required estimates are similar (55.9% versus 60.3%), showing no hop-based attenuation there. This refines the cell-level result: exposure depends not only on whether topology introduces additional recipients, but whether the value persists along the realized path to them.

5.8.3 Evidence for a Relay-Semantic Explanation. The traces are consistent with different relay semantics across frameworks. In AutoGen, relay-required propagation involves successive inter-agent exchanges; the value must be retained and reproduced across multiple messages to reach a downstream recipient, and the lower relay-required exposure is consistent with propagation failures across these exchanges. LangGraph instead maintains shared state along the relevant path, keeping earlier information available to downstream agents, consistent with showing no exposure reduction when recipients require additional relay hops.

A separate trace-level measure supports this: AutoGen relay failures occur in multi-hop CW-7 and Hier-7 at 12.2% and 8.5%, while P2P cells, requiring no relay, show 0% (Table 15). Relay failure is distinct from leakage: it records failure to propagate through a required relay operation, while leakage records whether the value is ultimately disclosed. These observations are consistent with a relay-semantic explanation for the AutoGen–LangGraph difference, though they do not establish message reproduction or shared-state execution as the sole cause; scheduling, state management, and execution control may also contribute.

5.8.4 Implication for Topology-Aware Privacy Enforcement. This analysis distinguishes *nominal topology* from *realized information flow*: the specified graph determines possible paths, but observed exposure also depends on how an orchestrator realizes and preserves information along them, so agent count, edge count, and topology class alone cannot characterize exposure. A sanitizer should therefore not infer authorization from a protected value’s presence or nominal topology alone; the decision should incorporate the intended recipient and observable communication context, an authorization decision being a fusion of value, realized recipient, and context. Therefore, we develop RECONS, operationalizing this

by conditioning sanitization on recipient and observable execution context rather than a fixed transformation per detected value. The cross-framework results motivate the recipient- and context-conditioned enforcement evaluated in §6.

6 RECONS: Recipient-Conditioned Sanitization

The evaluated multi-agent LLM workflows exhibit predominantly *direct and propagative* leakage: a value disclosed by one agent can propagate through channels and execution steps to an agent unauthorized to receive it (§5.5). We did not observe successful reconstructive or behavioral side-channel leakage, which does not establish absence in general but identifies disclosure detection and propagation control as the primary mitigation targets this evaluation supports.

RECONS implements these as a three-stage pipeline, DETECT → DECIDE → TRANSFORM, as shown in Figure 7. Detection identifies candidate values and assigns a type and confidence score, decision determines whether the detected value is authorized for the intended recipient under observed context, and transformation applies a type-appropriate operation when disclosure is unauthorized, separating what information is present, whether it may be disclosed, and how unauthorized information should be transformed.

6.1 Positioning

Existing mechanisms operate at the user–LLM boundary or apply recipient-independent transformations. AgentLeak’s blanket-redaction interceptor removes detected content uniformly, costing 4.7 points of task success; Presidio and commercial DLP similarly detect predefined types and apply fixed transformations; $\text{Pr}\in\text{empt}$ [6] provides token-level FPE and metric-LDP but targets single-agent, single-prompt settings without modeling recipient-dependent disclosure across agents.

RECONS instead targets values authorized for one recipient but not another as they propagate through a workflow, centered on a *recipient-aware authorization layer* extending a field-level allowed-set policy with the actual recipient when observable, coupled with a *type-aware transformation layer*: FPE for format-preserving needs, pseudonymization for non-numeric value-dependent attributes, mLDP where numeric perturbation is acceptable, contextual rewriting for sensitive free text, and redaction as fallback. RECONS operates as an *inline interceptor*, evaluating each agent-authored event before it reaches a downstream agent, tool, memory component, or recipient, distinguishing it from post-hoc sanitization and letting transformed content participate in subsequent steps, including legitimate returns to an originating agent.

6.2 Architecture

RECONS processes events on C_1 – C_7 in three stages:

- **Detection.** Identifies candidate values, assigns a type, and records confidence.
- **Decision.** The authorization policy $\mathcal{A}(\tilde{a}, \zeta, c, r, pos)$ determines whether value $\tilde{a}$ may be disclosed on channel c to recipient r at topology position pos , passing authorized disclosures unchanged and forwarding unauthorized ones to transformation.
- **Transformation.** Selects an operation by value type and required utility: FPE for format-sensitive values, mLDP for numeric

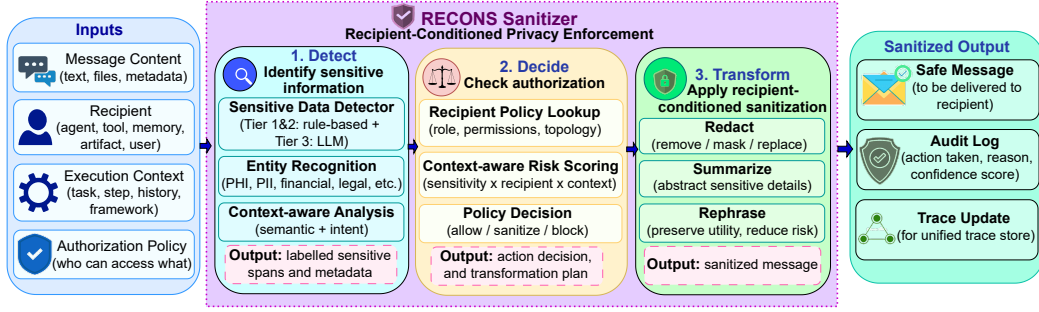


Figure 7: RECONS recipient-conditioned privacy enforcement. The sensitive information detector identifies what may be sensitive in the message. Policy decision engine determines what to do based on who can receive it and in what context. The sanitization module modifies the message according to the decision. Trace & audit logger provides transparency and supports post-hoc auditing.

values where approximate utility suffices, pseudonymization for the corpus’s non-numeric value-dependent attributes, contextual rewriting for free text, and redaction when no suitable transformation applies.

The three types are defined by the transformation they require: *format-sensitive* values need representation preserved over content; *value-sensitive* values contribute numerically or semantically to downstream computation and should be preserved approximately; *context-sensitive* values derive sensitivity from meaning rather than surface form and require rewriting or removal. This typing interfaces detection with transformation (§6.4–6.5, evaluated in §6.6).

6.3 Detection Layer

The detector must identify protected values across heterogeneous agent-authored channels while providing information sufficient for authorization and transformation. Since the corpus contains both structured values and contextually sensitive free text, a single mechanism is insufficient, so RECONS uses a three-tier detector progressing from cheap pattern-based detection to semantic analysis.

6.3.1 Design. The detector comprises three tiers:

- **Tier 1: rule and NER detection.** Pattern matching and named-entity recognition for structured values, SSNs, account and card numbers, phone numbers, identifiers, amounts, and dates, with checksum validation and domain-specific recognizers where generic rules fall short.
- **Tier 2: zero-shot NER.** GLiNER [32] extends coverage to entities lacking reliable patterns or conventional rules, replacing a Tier-1 detection only when its confidence exceeds Tier-1’s, otherwise retaining the higher-confidence Tier-1 result.
- **Tier 3: contextual LLM detection.** A calibrated judge identifies information expressed semantically rather than by pattern or entity type, a pending lawsuit, a strategic decision, a health condition in prose. It operates on free-text channels C_1 , C_2 , C_6 and the semantic content of C_5 and C_7 , using the pinned MISTRAL-SMALL-24B judge at threshold $\zeta = 0.50$ with a fixed prompt hash calibrated on held-out labels; each detection yields a candidate (span, value, value_type, channel), where value_type $\in \{\text{format_dependent}, \text{value_dependent}, \text{contextual}\}$.

Each detection retains its source tier and confidence, treated as evidence rather than ground truth downstream, letting the decision

Table 16: Pre-tuning detection recall by value type over 3,262 positive instances.

Value type	Recall	Caught / total
Value-sensitive	85.0%	731 / 860
Format-sensitive	65.1%	1023 / 1571
Context-sensitive	36.0%	299 / 831
Overall	62.9%	2053 / 3262

layer account for reliability differences across channels and value types.

6.3.2 Validation Protocol. We validate on a development split, reserving held-out scenarios exclusively for end-to-end evaluation (§6.5.3): 80 scenarios (16.5%) development, 405 held-out. Across both τ and τ' executions, the detector processes 12,606 agent-authored instances with known ground truth. Recall is computed over positive instances stratified by value type, domain, channel, and attribute; precision is estimated by manual annotation of sampled detections. Protected attributes are enumerated before evaluation and assigned a value type, with domain-specific recognizers added where generic detection is insufficient.

6.3.3 Results: Detection Varies by Value Type. Pre-tuning detection varies substantially by value type (Table 16): among 3,262 positive instances, recall is 85.0% for value-sensitive, 65.1% for format-sensitive, and 36.0% for context-sensitive values, with structured identifiers like ssn, email, insurance_id, and name detected more reliably than weakly patterned numeric values and contextual attributes. This exposes a surface-level detection limitation: contextual attributes like a chief complaint or internal note can be privacy-sensitive without a distinctive lexical, entity, or formatting signal, making a semantic tier necessary rather than merely an enhancement.

LLM tier. Since rule and NER tiers achieve only 36.0% recall on context-sensitive values, we evaluate the Tier-3 judge on 60 development instances, 40 positive (stratified between cases detected and missed by preceding tiers) and 20 negative controls: the judge identifies 39 of 40 positives with no false positives on controls, demonstrating semantic-tier utility for these sampled cases without characterizing population-level precision or recall. The judge’s operating point is inherited fr

Table 17: Pre-tuning recall for format-sensitive values across channels, isolating channel representation from value-type composition.

Channel	Kind	Recall
C_2 (inter-agent)	free-text	85.3%
C_6 (logs)	free-text	83.7%
C_7 (artifacts)	mixed	79.3%
C_1 (final output)	free-text	73.7%
C_4 (tool outputs)	mixed	71.6%
C_5 (memory writes)	structured	40.5%
C_3 (tool arguments)	structured	39.2%

6.3.4 Channel Dependence. Detection performance also depends on channel representation. Comparing format-sensitive values across channels to reduce value-type confounding (Table 17), recall is substantially lower on structured channels, C_3 tool arguments (39.2%) and C_5 memory writes (40.5%), than on free-text channels like C_2 (85.3%) and C_6 (83.7%). Since low detection confidence does not imply low privacy risk, and the weakest-detection channels are also where substantial disclosure occurs, RECONS treats confidence as channel-conditioned evidence rather than a direct risk proxy, using conservative fallback where detection and recipient resolution are insufficient.

6.3.5 Precision and Tuning. An automatic span-overlap proxy estimates precision at 15.7%, but treats detections of untracked sensitive fields as false positives; manual inspection of 40 sampled non-target detections found 35% actually contained sensitive content belonging to other fields, with remaining errors mostly field labels misclassified as values and agent identifiers like “A1” and “A2” misclassified as PII. A tuning pass on the development split fixed these failure modes without reducing recall anywhere: overall recall rose from 62.9% to 66.7%, format-sensitive recall from 65.1% to 73.0%, with the largest gain in local phone numbers, 19.6% to 99.3%, after adding a format-specific recognizer.

6.3.6 Carried-Forward Limitations. Three limitations carry forward to the decision layer: structured channels have lower recall than free-text channels for the same value type; agent identifiers and canary-like tokens can be textually ambiguous, creating corpus-specific precision-recall trade-offs; and contextual coverage depends on, and inherits the operating characteristics of, the calibrated Tier-3 judge. These motivate a decision policy operating under incomplete, uncertain observations rather than treating detector output as a complete representation of privacy risk.

6.4 Decision Layer

The decision layer is RECONS’s policy component. Given a detected value, its type and confidence, channel, recipient, and topology position, it selects an action $\mathcal{A}(\tilde{a}, \zeta, c, r, pos) \in \{\text{ALLOW, FPE, mLDP, REWRITE, REDACT}\}$. It first determines whether disclosure is authorized, then selects a value-type-compatible operation when protection is required. The transformation layer (§6.5) executes the selected operation. This design targets propagative leakage by sanitizing a value reaching an unauthorized recipient while preserving authorized disclosures.

6.4.1 Authorization Source. RECONS reuses the field-level release policy underlying the topology leakage metric (§5.1): each value specifies its authorized agents, with authorization reassigned across topology cells so richer-topology-introduced agents are unauthorized by construction, the same ground truth used by `new_agent_disclosed_rate`. Recipient resolution applies only where an explicit inter-agent boundary exists in the corpus: `event_recipient` identifies recipients for inter-agent messages (C_2), and memory reads (C_5 -memory_read), with no recipient defined for C_1, C_3, C_4, C_5 -memory_write, C_6 , or C_7 . So, recipient-aware enforcement operates only on C_2 and C_5 reads, the rest using the field-level policy. This also explains why so few authorized cases exist for assessing over-sanitization: the policy authorizes relatively few fields per recipient, leaving only 17 should-allow cases in the evaluated subset (Table 18).

6.4.2 Static and Dynamic Policies. We compare two policies differing only in recipient awareness. *Static- $\mathcal{A}$* applies the field-level allowed set regardless of actual recipient, while *Dynamic- $\mathcal{A}$* adds recipient-specific authorization via three rules.

- **Recipient authorization.** Allows a value if the resolved recipient is authorized; otherwise, sanitizing by value type, FPE for format-dependent, pseudonymization or mLDP for value-dependent, rewriting for contextual, redaction as fallback.
- **Confidence-conditioned protection.** Defaults uncertain detections toward protection where a recipient is resolvable. This rule is inapplicable to C_3 because no recipient is defined for that channel and of limited use for C_5 reads in this split.
- **Topology-position extension.** Sanitizes values graph-reachable from an authorized immediate recipient to an unauthorized agent. This extension is evaluated separately and excluded from the core policy.

6.4.3 Decision-Layer Validation. We evaluate decision correctness on the development split independently of transformed outputs. Among 4,834 protected-value occurrences, the detector identifies 3,400 (70.3%). Because recipient-aware decisions require a resolvable recipient, analysis is restricted to 2,109 occurrences both detected and resolvable on C_2 or C_5 reads. Thus, the results in this section are conditional on successful detection and recipient resolution. End-to-end performance is evaluated separately in §6.6.

We distinguish *under-sanitization*—a required-protection value allowed, constituting a privacy failure—from *over-sanitization*, an authorized value unnecessarily sanitized, incurring a potential utility cost.

Recipient-aware authorization removes topology-induced under-sanitization. Among 2,092 occurrences requiring sanitization, *Static- $\mathcal{A}$* allows 50 (2.39%) to pass, all when a field-level-permitted value reaches an unauthorized topology-introduced agent, while *Dynamic- $\mathcal{A}$* allows none (Table 18), eliminating these by conditioning authorization on the resolved recipient. Neither policy over-sanitizes the 17 should-allow cases. However, given the small sample, this result indicates no observed over-sanitization rather than a utility advantage.

Operation selection follows value type. The 2,092 sanitization decisions comprise 1,260 FPE, 480 mLDP/pseudonymization, and 352

Table 18: Decision correctness on the detected, recipient-resolvable development subset ($n = 2,109$). Dynamic- $\mathcal{A}$ eliminates the 50 topology-induced under-sanitization cases observed under Static- $\mathcal{A}$.

	Static- $\mathcal{A}$	Dynamic- $\mathcal{A}$
Under-sanitization	2.39%	0.00%
should-sanitize, allowed (/2,092)	50	0
Over-sanitization	0.00%	0.00%
should-allow, sanitized (/17)	0	0

rewriting operations. Each decision selects the operation associated with its assigned value type, with no observed type-operation mismatch.

6.4.4 Topology-Position Extension.

The topology-position extension tests whether authorization of the immediate recipient is sufficient in richer communication graphs. Of the 17 cases allowed by the core Dynamic- $\mathcal{A}$ policy, unrestricted graph reachability reclassifies 14 as requiring sanitization, corresponding to an 82% over-sanitization rate among these authorized cases. This result illustrates a limitation of unrestricted potential reachability. An authorized recipient may have a graph path to an unauthorized agent even when the value is never forwarded along that path. Therefore, applying reachability alone would impose unnecessary sanitization. The extension is consequently excluded from the core policy. A more targeted approach would condition topology analysis on observed forwarding or other evidence that the value is actually being propagated.

6.5 Transformation Layer

The transformation layer executes the decision layer’s selected action (§6.4), passing content through unchanged when the decision is ALLOW, without independently re-evaluating authorization; it applies operation-specific verification checks. We verify the mechanisms on a fixed 103-occurrence development sample from 12 scenario-topology-cell pairs, each re-evaluated using the same pinned Tier-3 detector used throughout, so results characterize mechanism behavior under the study’s instrument rather than an independent non-disclosure guarantee. End-to-end leakage is evaluated separately on the full held-out set in §6.6.

6.5.1 Transformation Operations. FPE for format-dependent values. FPE transforms format-dependent values, identifiers, account numbers, dates of birth, and canary tokens, using FF3-1. The transformation is reversible with the encryption key for authorized downstream recovery. Development verification found a boundary case: short-domain attributes like the three-digit `credit_score` fail FF3-1’s minimum domain-size requirement, and an initial implementation padding and truncating ciphertext to restore original length broke invertibility. We removed truncation and retain the full padded ciphertext instead, treating these as representation-length exceptions rather than strict length-preserving FPE. Evaluated FPE cases preserve the configured character set and remain exactly reversible.

Pseudonymization and mLDP for value-dependent values. The corpus’s naturally occurring value-dependent attributes, `name`,

`address`, `patient_name`, are non-numeric. So, this path uses deterministic keyed pseudonymization, the same value mapping to the same pseudonym under a fixed key. No collisions were observed, though this is not a formal guarantee. A separate numeric mLDP path uses the Laplace mechanism with $\epsilon = 1.0$ under the ℓ_1 metric, verified synthetically since no corpus attribute exercises it. Under the specified sensitivity and metric assumptions, the Laplace mechanism provides the corresponding ϵ -DP guarantee. The privacy parameter is configurable to support deployment-specific privacy-utility trade-offs.

Verified rewriting for contextual values. Contextual values are rewritten under a frozen prompt, each candidate checked by the pinned detector before acceptance, rejected and retried up to three times if at or above threshold before falling back to redaction. Development found two failure modes. First, an early prompt caused the model to reproduce a protected field’s attribute name as filler, fixed by treating field names as non-output identifiers. Second, the detector can classify broader paraphrases as disclosures. For example, replacing *pending harassment lawsuit* with a *pending legal matter* scores $0.8 \geq \zeta$. So, retries treat the detector as a conservative gate rather than assuming natural paraphrase is safe. Since the rewriting model and judge are not independent, verification establishes only that outputs satisfy the pinned detector criterion, not independent semantic non-disclosure; the full evaluation also uses an exact-match backstop, accepting a rewrite only if it passes semantic verification and contains no literal protected value. This was introduced after full-scale evaluation found that literal echoes semantic verification alone admitted (§6.6).

Redaction and fallback. Redaction serves as fallback when a transformation cannot be safely applied, FPE failing on format, pseudonymization or mLDP failing to produce a valid result, or rewriting exhausting its attempts. Each fallback’s source is recorded to distinguish transformation-specific failures from ordinary redaction.

6.5.2 Mechanism Verification. On the development sample (Table 19), all 103 transformed occurrences are classified as non-disclosing post-transformation (100% detector-based P -removal), specific to the detector rather than proof of semantic non-disclosure. The sample comprises 18 FPE, 50 mLDP/pseudonymization, and 35 rewriting decisions; of the latter, 27 (77.1%) pass on the first attempt and 8 (22.9%) exhaust all three and fall back to redaction, with no second- or third-attempt successes. No FPE or pseudonymization fallbacks in this sample, and no FPE format-recognition fallback across the full 2,595-occurrence FPE population. Successful rewrites score 0.8 to 0.95 on a separate meaning-preservation evaluator, applied only to successful rewrites here and reported as supplementary characterization rather than an end-to-end utility estimate.

6.5.3 Evaluation Protocol. All sanitizer components, thresholds, prompts, and policies are fixed using the 80-scenario development split. Evaluation uses all 405 held-out scenarios across six topology cells, yielding 2,430 scenario-cell runs and 34,884 protected-value occurrences. Since 105 scenarios share no common horizon with the rest of the corpus, each is evaluated at its canonical horizon where available, otherwise its shortest covered horizon, with no parameter re-tuned under this assignment. The implications of this horizon

Table 19: Transformation-layer verification on the development sample ($n = 103$). FPE checks combine 18 corpus-occurring and 27 synthetic cases; other rows are scoped to the indicated sample or corpus population.

Metric	Result
Decided action: FPE / mLDP / rewrite	18 / 50 / 35
Rewrite outcome: first-attempt clear / redaction fallback	27 (77.1%) / 8 (22.9%)
Detector-based P -removal	103/103 (100%)
FPE round-trip: corpus / synthetic	18/18 + 27/27
FPE character-set preservation	45/45
FPE length preservation	41/45
Wrong-key recovery differs from original	27/27
FPE format-recognition fallback (full corpus, $n = 2,595$)	0.0%
Pseudonymization determinism / P -removal	50/50
mLDP (synthetic): $P(\text{error} \leq 10k)$	1.000
mLDP DP-inequality diagnostic	23/24
ALLOW passthrough (sampled)	10/10

heterogeneity are discussed in §6.6.5. Leakage is measured using the same pinned three-tier detector throughout, so pre- and post-sanitization measurements share the same evaluation instrument. **Authorization-anchored metrics.** We report two authorization-conditioned metrics: *residual leakage*, the fraction of should-sanitize occurrences ($n = 34,581$) still accessible to an unauthorized recipient after sanitization, and *correct-allow*, the fraction of should-allow occurrences ($n = 303$) still accessible to their authorized recipient. Lower residual leakage and higher correct-allow are preferable. Correct-allow measures authorized-disclosure retention, not general utility, anchored to authorization ground truth rather than task performance or content quality. We therefore separately report a content-level meaning-preservation proxy on a stratified 120-occurrence sample, with 40 per value type.

Baselines. All conditions use identical held-out executions: *None* performs no sanitization, *blanket redaction* [30] redacts every detected value regardless of recipient, *Presidio* [25] applies fixed-infoType DLP detection followed by uniform redaction, and *RECONS* applies the proposed recipient-aware policy and type-aware transformation. Presidio serves as a reproducible open baseline for recipient-blind, fixed-type DLP.

6.6 RECONS Evaluation

Table 20 summarizes the end-to-end results. RECONS reduces residual leakage from 96.7% without sanitization to 29.25%, versus 28.3% for blanket redaction, while retaining 94.1% of authorized disclosures versus blanket redaction’s 20.5%, approaching blanket redaction’s leakage rate while retaining substantially more authorized disclosures. Presidio performs worse on both, 35.5% residual leakage and 34.7% correct-allow. Among the 395 held-out scenarios with no legitimately disclosable field, RECONS records 28.0% residual leakage versus blanket redaction’s 28.4%, so the aggregate difference concentrates in the 10 scenarios with legitimately disclosable fields, where RECONS preserves an authorized disclosure that blanket redaction suppresses by design.

6.6.1 Sources of Residual Leakage. The decomposition localizes most residual cases to detection misses and recipient-unresolvable cases (Table 21). Of RECONS’s 10,116 residual cases, 9,657 (95.5%)

Table 20: End-to-end results on the held-out set (405 scenarios; 2,430 scenario–cell runs; 34,884 protected-value occurrences). Residual leakage is lower-is-better; correct-allow is higher-is-better.

Condition	Residual leakage ↓	Correct-allow ↑
None	96.7%	0.0%
Presidio (DLP)	35.5%	34.7%
AgentLeak blanket redaction	28.3%	20.5%
RECONS	29.25%	94.1%

Table 21: Decomposition of RECONS’s 10,116 residual cases. Detection misses dominate; field-level fallback is the principal RECONS-specific source.

Cause	Count	Share
Detection miss (never flagged)	9,657	95.5%
Field-level fallback (no resolvable recipient)	452	4.5%
Post-transformation detector-positive	7	0.07%
Total	10,116	100%

Table 22: RECONS residual leakage by protected-value type. Contextual values exhibit the highest residual exposure, consistent with weaker contextual detection.

Value type	Residual leakage
Value-dependent	17.6%
Format-dependent	23.3%
Contextual	55.7%

are undetected values that never reached the decision or transformation layers. A further 452 (4.5%) are detected values with no resolvable recipient on C_1 , C_3 , C_4 , C_6 , or C_7 , using the field-level fallback, and seven cases (0.07% of residual cases, 0.02% of all should-sanitize occurrences) are sanitizer-acted content the evaluation detector still flags, examined individually in §6.6.4 as post-transformation detector-positive rather than confirmed disclosures.

Detection coverage is thus the dominant source: 69.2% of should-sanitize occurrences were detected on the held-out set versus 66.7% recall on the development split (§6.3.5), populations that need not match. While detection recall does not directly predict residual leakage given that channel, authorization, and transformation also matter, the sanitizer plainly cannot act on an undetected value. The 452 fallback cases are a separate limitation, since unresolvable recipients preclude recipient-specific authorization. Thus, detection coverage and recipient resolution address distinct sources of residual exposure.

The value-type breakdown agrees (Table 22): residual leakage is 17.6% for value-dependent, 23.3% for format-dependent, and 55.7% for contextual values, the last consistent with contextual values’ lower detection recall (§6.3).

6.6.2 Channel-Level Disclosure. By channel (Figure 8), C_3 , structured tool-call arguments, shows the highest residual leakage across conditions, consistent with lower structured-channel recall (§6.3) and field-level fallback for unresolvable C_3 recipients (§6.4). RECONS matches or improves on blanket redaction only on C_1 and C_2 (32.5% vs. 32.6%; 25.2% vs. 25.7%); elsewhere it trails: C_3 43.9% vs. 41.4%, C_4 26.5% vs. 24.1%, C_5 32.2% vs. 31.6%, C_6 19.8% vs. 17.4%, C_7

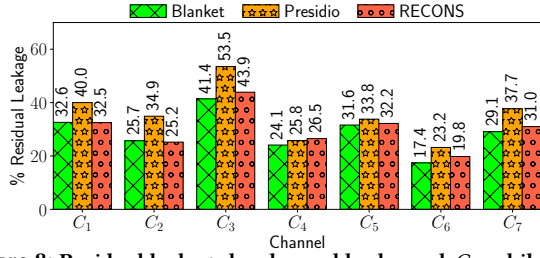


Figure 8: Residual leakage by observable channel. C_3 exhibits the highest residual leakage across the sanitizing conditions; RECONS trails blanket redaction on every channel except C_1 and C_2 .

Table 23: Disclosure to topology-introduced agents before and after sanitization.

Condition	CW-2	CW-7	P2P-4	P2P-7	P2P-10	Hier-7
None	0%	98.0%	97.9%	98.0%	97.6%	97.8%
Presidio [25]	0%	34.2%	34.7%	35.1%	35.8%	35.0%
Blanket Redaction [30]	0%	26.7%	27.0%	25.5%	25.9%	26.5%
RECONS	0%	25.9%	26.9%	25.1%	25.4%	26.5%

31.0% vs. 29.1%. The largest gap is on C_3 , matching its concentration of unresolvable cases.

6.6.3 Topology Gradient Under Sanitization. Next, we evaluate whether RECONS reduces disclosure to the topology-introduced agents identified in §5.1. CW-2, with no topology-introduced agents, gives a 0% reference point. Across the five richer topologies (8,873 new-agent-recipient occurrences), unsanitized disclosure is 97.6–98.0% versus RECONS’s 25.1–26.9% (Table 23). RECONS beats blanket redaction in four of five non-trivial cells and ties in Hier-7. Since topology-introduced agents are unauthorized under both, this does not isolate recipient-aware authorization’s contribution, which instead shows in authorized-disclosure retention (Table 20). Presidio leaves a higher residual floor, 34.2–35.8%, across these cells.

6.6.4 Ablation Analysis.

Dynamic versus static authorization. Replacing recipient-aware authorization with a static allow-list raises residual leakage from 29.25% to 29.81%, 0.56 points, while raising correct-allow from 94.1% to 100%. The static policy preserves all legitimately allowed fields regardless of recipient. The two policies differ mainly on the 10 scenarios with legitimately disclosable fields, limiting the effective sample size of this comparison. Within this scope, the result is consistent with recipient awareness sharpening field-versus recipient-level authorization.

Transformation menu versus redaction. Replacing the FPE, pseudonymization, and rewriting menu with uniform redaction under the same decisions changes residual leakage only marginally, from 29.25% to 29.59%, with correct-allow unchanged at 94.1%. Because both conditions share authorization decisions, this result indicates that the transformation menu’s role is preserving authorized information and utility rather than lowering leakage itself.

Transformation verification. At full scale, semantic-only rewrite verification admitted some literal copies of the protected value. So, we added an exact-match backstop requiring a rewrite to

Table 24: Content-level meaning-preservation proxy by condition (stratified sample, $n = 120$; 40 per value type) and by value type for RECONS. The proxy does not fully characterize structured-field usability or task-level utility.

Condition	Utility
AgentLeak blanket redaction	99.3%
Presidio (DLP)	97.3%
RECONS (redact-only ablation)	98.8%
RECONS	83.5%
<i>RECONS, by value type:</i>	
Format-dependent (FPE)	91.6%
Value-dependent (pseudonym.)	89.6%
Contextual (rewrite / redact)	69.1%

pass semantic checking and contain no literal occurrence, falling back to redaction otherwise. This cut post-transformation detector-positive cases from 67 to 7, an 89.6% reduction. Of the seven remaining cases, six involve negated true values, such as “no criminal history,” where rewrites contain no literal occurrence but the judge still classifies them as disclosures, reflecting reference-to-topic versus disclosure-of-value. The seventh reflects disagreement between two calls to the same judge on identical, literal-value-free content, consistent with batched-inference non-determinism at temperature zero. None reflects literal reintroduction bypassing the exact-match backstop, though this establishes only that under this instrument, not immunity to semantic or inferential leakage.

6.6.5 Utility and Evaluation Limitations. Table 24 reports the content-level meaning-preservation proxy. The proxy assigns higher scores to blanket redaction (99.3%) than RECONS (83.5%), with the largest gap for contextual values (69.1%), possibly reflecting the proxy’s sensitivity to surface-form change: FPE and rewriting alter textual representations, while redaction can leave syntactically coherent output despite removing information. RECONS scores higher for format-dependent (91.6%) and value-dependent (89.6%) values, where FPE and pseudonymization apply. Since the proxy does not directly measure structured-field usability or task-level utility, we do not read the 99.3% versus 83.5% gap as evidence of greater practical utility for blanket redaction; correct-allow remains the primary retention measure, being directly anchored to ground-truth authorization labels.

Overall, recipient-aware sanitization approaches blanket redaction’s residual-leakage performance while substantially increasing authorized-disclosure retention. The dominant remaining limitation is detection coverage, particularly for contextual values, with recipient resolution a separate limitation of the current policy.

7 Discussion and Limitations

7.1 Design Implications

Our findings suggest privacy protection in MALS cannot be designed from static architectural assumptions and must instead be enforced against the realized recipient and execution path at run-time. The evidence supports three bounded conclusions: topology changes protected-value recipient exposure; recipient-conditioned enforcement improves the privacy–authorization trade-off relative to blanket redaction; and detection coverage is the principal

remaining bottleneck, with recipient resolution a secondary boundary. The rest of this section develops each in turn with its bounding limitations.

Topology as a security-relevant configuration. Topology-driven exposure (§5.1) is not an artifact a better detector or implementation would remove but a structural consequence of which agents a communication graph makes reachable. Surface leakage varies—whether a value is disclosed at all—by only 2.2 points across six cells, while new-agent disclosure—whether that value reaches a recipient the topology introduces—rises from a structural 0% to 52.2% over the same cells. So topology belongs in a system’s security configuration alongside access control, not only performance or scalability planning. We claim no universal topology law: ranking among richer topologies is unstable across frameworks and models (§5.6, §5.7), and no one topology, hierarchical delegation included, is asserted as intrinsically riskier. What generalizes is that richer topology creates exposure a simpler one structurally cannot, not its magnitude or ordering.

Recipient Conditioning as the Key Design Principle. RECONS’s decision layer implements one conditional rule, $\text{unauthorized} \rightarrow \text{SANITIZE}$, $\text{authorized} \rightarrow \text{ALLOW}$, the entire mechanism separating recipient-conditioned from recipient-blind enforcement applied to the same detected value. Its advantage is not dramatically lower leakage; RECONS’s residual 29.25% sits close to blanket redaction’s 28.3% since both discard topology-introduced disclosure identically. The advantage is selective enforcement, which retains 94.1% of authorized disclosures against 20.5% by permitting exactly what a recipient-blind policy cannot distinguish from a violation. Recipient conditioning should be judged on what it preserves, not how far it lowers an already detection-bounded rate. This comparison rests on a modest base, only 303 should-allow occurrences, with the RECONS-versus-blanket difference concentrated in just 10 of 405 scenarios containing a legitimately disclosable field, so the 94.1% versus 20.5% gap should be read as consistent with the underlying behavior rather than as a precise population estimate.

Detection as the Principal Bottleneck. The RECONS pipeline’s detect-decide-transform structure means transformation can only apply to a value the detector already flagged: an undetected value never reaches the decision layer and passes through unchanged regardless of transformation sophistication. This is architectural, not correlational: 95.5% of end-to-end residual leakage (§6.6.1) traces to exactly this: disclosures never flagged, concentrated in contextual attributes with the weakest semantic recall, a ceiling any same-detector sanitizer, blanket redaction included, would share, since neither can act on what neither sees. Only improved detection, not decision or transformation refinement, recovers this loss.

Recipient Resolution as an Architectural Boundary. Recipient-aware authorization presupposes a resolvable recipient. C_2 messages and C_5 reads satisfy this directly; C_3 tool-call arguments target a tool, not an agent, so 95.8% of C_3 events are self-loops crossing no inter-agent boundary, with the remaining 4.2% reaching another agent via an already-covered C_5 read. Hence, field-level fallback on C_3 is correct given the channel’s structure, not a coverage gap. Extending recipient resolution to the small residual of genuinely tool-mediated disclosure would require richer provenance than call-and-response. Specifically, tracking which agent consumes a

tool’s output when not itself a logged C_5 read, and communication semantics that treat a tool’s result as attributable to whichever agent next acts on it. Absent that provenance, recipient resolution cannot be extended to this channel without inventing a recipient the trace does not evidence.

Horizon Heterogeneity. Each scenario is evaluated at its shortest available horizon (§4.1), unevenly across domains: 105 scenarios lack a horizon common to the rest, and even fully-covered domains skew, finance and healthcare almost entirely at T_1 , legal predominantly at T_3 or longer. So, cross-domain comparisons here are confounded with horizon length and should not be read as horizon-controlled domain estimates.

7.2 Threat-Model Limitations

Our threat model, and consequently every result in this paper, explicitly inherits the following limitations:

- **Corpus and generalization scope:** experiments derive from an English-language, four-domain subset of one public benchmark [30], with one generator and framework as the primary configuration; cross-model and cross-framework checks cover a handful of additional systems each, showing findings are not artifacts of one system but not that they generalize beyond those tested.
- **Latent-state leakage:** a value influencing a model’s later output through its own internal context or reasoning state rather than an explicit channel event. Hence, it cannot be detected, attributed to a recipient, or evaluated by a trace-based methodology like ours.
- **Stronger collusion:** unauthorized recipients pooling their individual observations.
- **Cross-session inference:** an adversary correlating evidence across independent executions rather than within one trace.
- **Unobserved side channels:** any channel beyond the seven instrumented (C_1 – C_7).
- **Attacks outside the evaluated threat model:** reconstruction from indirect evidence and behavioral side-channels, tested directly with no confirmed instance found (§5.3, §5.4), though this does not establish absence under a more capable or adaptive adversary.

8 Conclusion

We investigated privacy leakage in multi-agent LLM systems as a question of *where* a protected value travels, not only whether it is disclosed. Across six topology cells, topology changes recipient exposure rather than overall disclosure incidence, rising from 0% to 52.2% while surface leakage moves only 2.2 points, holding when replicated across additional models and frameworks. In a separate experiment holding topology fixed, framework choice alone produced a further two-orders-of-magnitude spread, demonstrating that architecture, not just communication structure, drives exposure. Exhaustive testing across three reconstruction axes and a behavioral side-channel found no confirmed leakage beyond direct disclosure and its propagation, confirming this is where these systems actually leak, not a scope chosen for convenience. We addressed this with RECONS, an inline DETECT $\rightarrow$ DECIDE $\rightarrow$ TRANSFORM sanitizer conditioning enforcement on recipient rather than

sensitivity alone, reducing residual disclosure to within one point of blanket redaction (29.25% versus 28.3%) while retaining 4.6× as many authorized disclosures (94.1% versus 20.5%), showing that suppressing unauthorized disclosure need not suppress what policy already permits.

Detection, not recipient resolution, is the binding constraint: 95.5% of residual cases are values no same-detector sanitizer could act on, a limitation any comparable system shares, while only 4.5% trace to RECONS’s own resolution step. Recipient-conditioned enforcement is therefore necessary but not sufficient; closing the gap depends more on detection coverage, particularly for contextual attributes, than on refining the decision layer. More broadly, privacy controls for multi-agent LLM systems cannot be assessed or built from static topology, framework, or model assumptions; they must act on the recipient and communication path realized at runtime.

References

- [1] [n. d.]. Agent Development Kit | Gemini Enterprise Agent Platform. <https://docs.cloud.google.com/gemini-enterprise-agent-platform/build/adk>
- [2] [n. d.]. CrewAI. <https://crewai.com>
- [3] [n. d.]. OpenAI Agents SDK. <https://openai.github.io/openai-agents-python/>
- [4] Sadia Asif and Mohammad Mohammadi Amiri. 2026. Information-Theoretic Privacy Control for Sequential Multi-Agent LLM Systems. arXiv:2603.05520 [cs.MA] <https://arxiv.org/abs/2603.05520>
- [5] Milon Biswas, Wei Yu, Qianlong Wang, and Weixian Liao. 2025. Quantifying Privacy Leakage in Multi-Agent LLMs: A Unified Theoretical and Empirical Analysis. In *2025 IEEE International Conference on Big Data (BigData)*. 1685–1694. doi:10.1109/BigData66926.2025.11401523
- [6] Amrita Roy Chowdhury, David Glukhov, Divyam Anshuman, Prasad Chalasani, Nicolas Papernot, Somesh Jha, and Mihir Bellare. 2025. Preempt: Sanitizing Sensitive Prompts for LLMs. arXiv:2504.05147 [cs.CR] <https://arxiv.org/abs/2504.05147>
- [7] Manuel Costa, Boris Köpf, Aashish Kolluri, Andrew Paverd, Mark Russinovich, Ahmed Salem, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2025. Securing AI Agents with Information-Flow Control. arXiv:2505.23643 [cs.CR] <https://arxiv.org/abs/2505.23643>
- [8] Hugo Garcia Cuesta, Pablo Mateo Torrejón, and Alfonso Sánchez-Macián. 2026. Multi-Agent Firewall Architecture for Privacy Protection of Sensitive Data in Interactions with Language Models. arXiv:2607.08282 [cs.CR] <https://arxiv.org/abs/2607.08282>
- [9] Edoardo DeBenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 82895–82920. doi:10.52202/079017-2636
- [10] Diego Gosmar and Deborah A. Dahl. 2025. Sentinel Agents for Secure and Trustworthy Agentic AI in Multi-Agent Systems. arXiv:2509.14956 [cs.AI] <https://arxiv.org/abs/2509.14956>
- [11] Shuang Guo and Zihui Li. 2025. SRPG: Semantically Reconstructed Privacy Guard for Zero-Trust Privacy in Educational Multi-Agent Systems. arXiv:2512.03694 [cs.MA] <https://arxiv.org/abs/2512.03694>
- [12] Gurusha Juneja, Jayanth Naga Sai Pasupulati, Alon Albalak, Wenye Hua, and William Yang Wang. 2025. MAGPIE: A benchmark for Multi-Agent contextual Privacy Evaluation. arXiv:2510.15186 [cs.CR] <https://arxiv.org/abs/2510.15186>
- [13] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- [14] LangChain, Inc. 2022. LangChain: the open agent platform to own your intelligence. <https://www.langchain.com>. Accessed: 2026-09-07.
- [15] LangChain, Inc. 2024. LangGraph: agent orchestration framework for reliable AI Agents. <https://www.langchain.com/langgraph>. Accessed: 2026-09-07.
- [16] Wenkai Li, Liwen Sun, Zhenxiang Guan, Xuhui Zhou, and Maarten Sap. 2025. 1-2-3 Check: Enhancing Contextual Privacy in LLM via Multi-Agent Reasoning. In *LLMSEC*, Leon Derczynski, Jekaterina Novikova, and Muhao Chen (Eds.). ACL. <https://aclanthology.org/2025.llmsec-1.9/>
- [17] Yuchen Ling, Shengcheng Yu, Zhenyu Chen, and Chunrong Fang. 2026. Toward Secure LLM Agents: Threat Surfaces, Attacks, Defenses, and Evaluation. arXiv:2606.10749 [cs.CR] <https://arxiv.org/abs/2606.10749>
- [18] Jinbo Liu, Defu Cao, Yifei Wei, Tianyao Su, Yuan Liang, Yushun Dong, Yan Liu, Yue Zhao, and Xiyang Hu. 2026. Topology Matters: Measuring Memory Leakage in Multi-Agent LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurges (Eds.). Association for Computational Linguistics, San Diego, California, United States, 39728–39746. doi:10.18653/v1/2026.findings-acl.1980
- [19] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2025. AgentBench: Evaluating LLMs as Agents. arXiv:2308.03688 [cs.AI] <https://arxiv.org/abs/2308.03688>
- [20] Jiaqi Luo, Songyang Peng, Jiarun Dai, Zhile Chen, Zhuoxiang Shen, Geng Hong, Xudong Pan, Yuan Zhang, and Min Yang. 2026. AgentGuard: An Attribute-Based Access Control Framework for Tool-Use LLM-Based Agent. arXiv:2605.28071 [cs.CR] <https://arxiv.org/abs/2605.28071>
- [21] Niloofar Miresghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2024. Can LLMs Keep a Secret? Testing Privacy Implications of Language Models via Contextual Integrity Theory. In *International Conference on Learning Representations*, B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (Eds.), Vol. 2024. 1892–1915. https://proceedings.iclr.cc/paper_files/paper/2024/file/08305d8b2ddab98932c163ea73df065f-Paper-Conference.pdf
- [22] Vaidehi Patil, Elias Stengel-Eskin, and Mohit Bansal. 2025. The Sum Leaks More Than Its Parts: Compositional Privacy Risks and Mitigations in Multi-Agent Collaboration. arXiv:2509.14284 [cs.CR] <https://arxiv.org/abs/2509.14284>
- [23] run-llama. [n. d.]. GitHub - run-llama/llama_index: LlamaIndex is the leading document agent and OCR platform. https://github.com/run-llama/llama_index
- [24] sophialagerkranspandey. [n. d.]. Introduction to Semantic Kernel. <https://learn.microsoft.com/en-us/semantic-kernel/overview/>
- [25] Data Privacy Stack. [n. d.]. Index - Presidio. <https://presidio.dataprivacystack.org/>
- [26] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D. Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322 [cs.AI] <https://arxiv.org/abs/2501.06322>
- [27] Sanidhya Vijayvargiya, Aditya Bharat Soni, Xuhui Zhou, Zora Zhuro Wang, Nouha Dziri, Graham Neubig, and Maarten Sap. 2025. OpenAgentSafety: A Comprehensive Framework for Evaluating Real-World AI Agent Safety. doi:10.48550/arXiv.2507.06134
- [28] Shouju Wang, Fenglin Yu, Xirui Liu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. 2025. Privacy in Action: Towards Realistic Privacy Mitigation and Evaluation for LLM-Powered Agents. In *EMNLP*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.).
- [29] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155 [cs.AI] <https://arxiv.org/abs/2308.08155>
- [30] Faouzi El Yagoubi, Godwin Badu-Marfo, and Ranwa Al Mallah. 2026. AgentLeak: A Benchmark for Internal-Channel Privacy Leakage in Multi-Agent LLM Systems. *IEEE Access* 14 (2026), 94960–94978.
- [31] Hailong Yang, Renhuo Zhao, Guanjin Wang, and Zhaozhong Deng. 2025. GAMA: A General Anonymizing Multi-Agent System for Privacy Preservation Enhanced by Domain Rules and Disproof Mechanism. arXiv:2509.10018 [cs.AI] <https://arxiv.org/abs/2509.10018>
- [32] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 5364–5376. doi:10.18653/v1/2024.naacl-long.300
- [33] Xueqiang Zhang, Xiaofei Dong, Yiru Wang, Dan Zhang, and Feng Cao. 2025. A Survey of Multi-AI Agent Collaboration: Theories, Technologies and Applications (DEAI ’25). Association for Computing Machinery, New York, NY, USA, 1875–1881. doi:10.1145/3745238.3745531

A Appendix

Table 25: Interaction archetypes used in trajectory construction.

Archetype	Working definition
BASE (initial task)	The trajectory’s opening turn, in which the acting agent is given the task and the protected value.
A (elaboration)	A turn adding detail or context to a prior statement, without being directly asked to confirm or reveal the protected value.
B (handoff)	A turn transferring the task, case, or context to another agent.
C (justification)	A turn explaining or defending a decision or recommendation.
D (error handling)	A turn responding to or correcting a mistake or inconsistency earlier in the trajectory.
E (closeout)	A turn concluding the trajectory or interaction, e.g. summarizing the outcome.
F (rapport)	A turn oriented toward social or relational exchange rather than task content.
G (confirmation probe)	A turn in which one agent asks another to confirm, deny, or restate a specific candidate value, directly soliciting disclosure if the candidate is correct.

Table 26: GEMINI-3.7-FLASH Hier-7 disclosure rates by protected attribute type. Structured identifiers exhibit lower observed disclosure rates than contextual and narrative attributes.

Attribute	Disclosure rate
SSN / DOB / phone / email / account	0.0%
Salary / balance / credit score	0.0%
Address	0.9%
Settlement amount	3.7%
Confidential strategy	27.6%
Internal notes	36.4%
Criminal history	63.6%