Canary-Based Attribution of AI Agents under Paraphrasing Attacks: An Empirical Study of *m*-of-*k* Detection

Abstract. Canary-based attribution protocols can link harmful AI agent behavior to the responsible vendor account. Central to these protocols is the *m*-of-*k* decision rule, which succeeds when at least *m* of *k* injected canaries are recovered from session logs. In adversarial settings where operators rewrite incoming content to suppress attribution signals, the choice of *k* and *m* directly shapes the trade-off between detection recall and false positive rate. We report a controlled experiment in a chat-based setting using 200 synthetic sessions, a 31-item semantic-canary universe, and a rule-based detector. At a zero-FPR operating point (*m* = 2), raising the canary budget from *k* = 2 to *k* = 3 improves post-paraphrase true positive rate from 47.5% to 78.5% (+31.0 pp). An oracle detector confirms that this gap traces to detector sensitivity rather than injection failures. The results suggest that adding a small amount of canary redundancy is more effective than using a stricter detection threshold.

Keywords: AI agent attribution; canary-based tracing; paraphrasing attacks; semantic canaries; *m*-of-*k* decision rule.

---

1. Introduction

Autonomous AI agents are now deployed across a range of consequential settings, where they send messages, invoke APIs, execute code, and retrieve web content on behalf of operators. The accountability gap this creates is well-documented: when an agent causes harm, affected parties can observe the behavior but have no direct path to the operator who configured and deployed it. Chocron et al. [1] formalize this as the *agent attribution* problem and propose a canary-based protocol for the vendor-hosted LLM setting. Because model calls from any agent pass through a vendor API and are logged against an operator account, an authorized authority can inject detectable markers—canaries—into the agent's input stream. If those markers survive the agent's processing pipeline, the vendor can search a bounded window of session logs and recover the originating account.

Two parameters govern whether the protocol performs well in practice. The *canary budget k* is the number of canaries injected per interaction; higher *k* provides redundancy when an adversarial wrapper suppresses some of them, at the cost of additional injection overhead. The *detection threshold m* is the minimum number of recovered canaries required for a session match; relaxing *m* increases recall but risks false positives when canary categories are not sufficiently distinctive.

Chocron et al. evaluate both parameters across communication, web, and cyber agent domains using learned classifiers for semantic detection. What remains unclear is how k and m interact when the semantic detector is imperfect, and whether additional canaries can compensate for detection errors caused by paraphrasing. This paper examines that question directly. We hold the adversary (a paraphrase wrapper) and the detector (keyword rules) fixed, vary *k* and *m*, and measure the resulting TPR–FPR surface. Instead of reproducing the original multi-domain evaluation, we focus on one aspect of the protocol that was not examined in detail.

2. Background

2.1 Agent Attribution and the Vendor Log

Agent attribution is difficult because the relevant information is split between different parties. An authority monitoring harmful behavior sees the agent's external actions—messages, API calls, posted content—but not the vendor session logs that record model inputs and account identifiers. The vendor holds those logs but has no visibility into what the agent does externally. Without a bridging signal, a report that "some agent caused harm around time τ" gives the vendor nothing to search against.

Canary injection provides the missing link. An authority places a canary κ inside content which the agent is expected to read; if the agent passes that content through to the vendor model, κ appears in the session log. The vendor, given κ and an approximate timestamp τ, can restrict its search to sessions active during that window and identify the matching account. No prior registration of agents is needed, and the agent's code need not be modified. This mechanism requires only that the authority control some portion of what the agent ingests.

2.2 Canary Types

A *lexical canary* is a specific token sequence (a UUID, reference code, or synthetic filename, etc.) which is recoverable by exact string match. It is cheap to search for but fragile: any rewrite that changes surface form will destroy it.

A *semantic canary* is a pattern defined at the level of meaning rather than surface form. Typical examples include discourse markers (urgency, authority, emotional register), topical constraints, or conversational style conventions. Detection requires classification rather than string matching, which introduces error, but the flip side is that a paraphrase preserving meaning while changing wording may leave the semantic canary recognizable even after the lexical form is lost.

2.3 The *m*-of-*k* Rule

The authority samples a subset $K \subset N$ of k canaries from a universe N of size n and injects each into the agent's interaction. A session is attributed when at least $m \leq k$ of those canaries are recovered. Setting $m = k$ demands that every canary survive—robust only when the detection pipeline is reliable. Setting $m = 1$ maximizes recall but allows attribution on a single match, which inflates false positives when canary categories occur naturally in unrelated sessions. This paper therefore studies how to choose k and m when detection is noisy and affected by paraphrasing.

3. Experimental Setup

3.1 Dataset

We generate 200 synthetic chat sessions from templates covering both benign contexts (customer support, FAQ assistance) and adversarial ones (scam automation, social engineering). Each session consists of alternating user and agent turns; the authority injects canaries into user-side turns, which the agent must read to continue the conversation. A separate set of 200 sessions from the same templates, generated without canary injection, forms the negative set for FPR measurement.

3.2 Canary Universe

The semantic canary universe N comprises 31 categories drawn from discourse-level patterns: expressed authority, urgency, sadness, anger, politeness, political framing, hedging, and self-disclosure, among others. We chose categories that (i) could be added naturally to conversational text without making the session implausible, and (ii) could be detected using keyword rules. This allows us to study the effect of k and m without introducing differences caused by the detector architecture.

3.3 Injection

For each condition (k, seed = 123), we sample K uniformly from N and rewrite user-side turns to include surface-form realizations of each sampled category. A lexical reference code is also embedded in each session to support the exact-match baseline.

3.4 Paraphrase Wrapper

The paraphrase wrapper rewrites user-side text before it would reach the model API. It is instructed to preserve explicit identifiers—reference codes, named entities, dates—verbatim, but it receives no information about the semantic canary categories. As a result, semantic patterns may be incidentally lost during rewriting while the lexical canary remains intact. This models an operator deploying paraphrasing as a general suppression strategy, without knowledge of which specific canary categories the authority has sampled.

3.5 Detectors

Semantic rule detector. Keyword lists for each category in N are applied to the session text; attribution is declared if at least m of the k sampled categories are matched.

Oracle detector. The oracle reads the ground-truth label set K from session metadata and declares attribution if at least m labels are present. It is unaffected by paraphrasing or keyword mismatches and represents a performance ceiling.

Lexical exact-match baseline. The detector searches for the reference code string. Attribution is binary.

3.6 Metrics

We report session-level TPR (fraction of positive sessions correctly attributed) and FPR (fraction of negative sessions incorrectly attributed) for each (k, m) pair. TPR is measured separately before paraphrasing (injected) and after (paraphrased). All results use seed 123 and are computed over the full 200-session positive and negative sets.

4. Results

4.1 Semantic Rule Detector

Table 1 reports attribution performance across $k \in \{2, 3\}$ and all feasible thresholds.

Table 1. Session-level attribution performance of the semantic rule detector.

k	*m*	TPR (Injected)	TPR (Paraphrased)	FPR
2	1	1.000	0.930	0.140
2	2	1.000	0.475	0.000
3	1	1.000	0.995	0.140
3	2	1.000	0.785	0.000
3	3	1.000	0.320	0.000

Without paraphrasing, every configuration reaches TPR = 1.000, confirming that injection embeds detectable surface-form realizations of each sampled category. Under paraphrasing, TPR falls across all configurations, and the extent of the drop is shaped by both *k* and *m*.

We also noticed that, at *m* = 1, FPR is 0.140 for both *k* values. This rate reflects keyword rules broad enough to match at least one category in 14% of unmodified sessions—an expected consequence of using surface-form heuristics over naturally varied chat text. Raising *m* to 2 eliminates false positives entirely but reduces paraphrased TPR, more sharply at *k* = 2 than at *k* = 3.

4.2 Canary Budget at Zero FPR

The most relevant comparison for the protocol is between the two configurations with zero FPR:

- (*k* = 2, *m* = 2): post-paraphrase TPR = 0.475
- (*k* = 3, *m* = 2): post-paraphrase TPR = 0.785

Adding one canary to the budget improves TPR by 31.0 percentage points while holding FPR at zero. With *k* = 3, the *m* = 2 threshold can be satisfied even when paraphrasing degrades one of the three injected categories—the additional canary provides exactly that margin. With *k* = 2, no such margin exists: a single failed detection causes the session to be missed.

4.3 Effect of Strict Thresholds

Requiring every canary to be recovered (*m* = *k* = 3) drops post-paraphrase TPR to 0.320—far below the *m* = 2 result. When paraphrasing damages even one of three semantic patterns, strict conjunction matching fails the session. The lesson is not

that high thresholds are never appropriate, but that they become viable only when each canary's survival probability is high; under generic paraphrasing with a keyword detector, that condition does not hold.

4.4 Oracle Upper Bound

The oracle achieves TPR = 1.000 at all thresholds on both the injected and paraphrased sets, and FPR = 0.000 on the negative set. This leads to two observations. First, the performance drop seen in the rule detector is not an artifact of injection failures; canary labels are present in every positive session by construction. The detector's shortfall traces to surface-form changes introduced by paraphrasing and to the keyword rules' sensitivity to those changes. Second, the oracle's zero FPR on the negative set rules out label contamination, confirming that the negative sessions were generated without injecting canary metadata.

4.5 Lexical Baseline

The exact-match detector achieves TPR = 1.000 and FPR = 0.000 in both conditions. This is a direct consequence of the wrapper's design, which preserves reference codes verbatim. It should not be read as evidence that lexical canaries survive paraphrasing in general—a wrapper without that constraint would be expected to destroy them. The baseline functions here as a pipeline sanity check: the consistent results confirm that the experimental machinery is operating correctly, and that the semantic degradation we observe reflects the paraphraser's effect on unconstrained text rather than any systematic error upstream.

5. Discussion

5.1 Redundancy versus Threshold Tightening

When detection is noisy, redundancy and a strict matching threshold have different effects. Increasing k gives the detector more opportunities to recover the required number of canaries, while increasing m makes the system less tolerant of missed detections. Raising m demands more of that evidence, which is counterproductive when individual detections are unreliable. The results support this pattern. At zero FPR, increasing k from 2 to 3 improves TPR by 31 percentage points, whereas moving from $m = 2$ to $m = 3$ at $k = 3$ costs 46.5 percentage points. This difference suggests that, under the conditions tested here, increasing the number of canaries is more effective than using a stricter threshold.

Among the configurations tested, (k = 3, m = 2) performs best, but the right choice in a production setting will depend on factors we do not model here—the cost of injecting additional canaries, the acceptable false positive rate under vendor-side investigation, and the expected strength of the adversary's wrapper.

5.2 Interpreting the Oracle Gap

At (k = 2, m = 2), the oracle achieves 100% TPR while the rule detector achieves 47.5%. This gap has two contributing sources. One is paraphrasing-induced semantic loss: the canary pattern may genuinely no longer be present in the rewritten text. The other is detector insensitivity: the pattern persists linguistically, but keyword rules fail to recognize its paraphrased surface form. These two sources cannot be separated without human annotation of the paraphrased sessions—annotators would need to judge whether each semantic category is present in meaning even when its keyword realizations are absent. We did not conduct that annotation here. The gap therefore provides an upper bound on the improvement that a better detector could achieve with the same canary budget. Whether a learned classifier would close most of that gap or only part of it is an open question.

5.3 Adversary Scope

The paraphrase wrapper in these experiments does not adapt to the canary taxonomy. An adversary who knows the full universe N could target semantic patterns specifically, selectively removing the categories most likely to have been sampled. The original paper discusses whether utility-bearing semantic canaries can resist targeted suppression, but we do not test this question experimentally. Extending this setup to include an adaptive, taxonomy-aware adversary, and measuring the resulting solve-rate degradation alongside attribution performance, would be a natural next step.

5.4 Scope of Generalization

These results hold for one domain (chat), one detector type (keyword rules), one paraphrase wrapper, and two values of k. Web agent settings introduce preprocessing pipelines that may strip content before any adversary acts; cyber agent settings involve artifacts—filenames, HTTP headers, source files—where lexical canaries are often utility-bearing and therefore harder to remove. The redundancy benefit we identify is likely present in those domains too, but its magnitude will depend on canary placement, the agent's ingestion pipeline, and the detector's accuracy. We therefore leave cross-domain evaluation to future work rather than extrapolating beyond the current experiments.

6. Limitations

The experiments are limited in scale. With 200 sessions in each split, we can reliably detect large differences in TPR, but FPR estimates at the zero level should be treated as upper bounds rather than precise measurements; at this sample size, a true FPR of 0.005 would go undetected with high probability.

The semantic detector is rule-based. Keyword lists underperform learned classifiers in the presence of synonym substitution and register variation—two transformations that paraphrasers apply routinely. A learned classifier may achieve higher TPR than the rule-based detector used here, but the size of the improvement is unknown because we did not run that comparison.

The adversary is not adaptive to the detector or the canary taxonomy. Its paraphrasing degrades semantic canaries as collateral damage rather than by design. A more capable adversary—one that identifies and selectively suppresses the patterns in N—could reduce TPR further, potentially below the levels we observe at $m = 2$.

Finally, the sessions are synthetic and generated from a small number of templates. Natural chat data exhibit greater lexical and stylistic diversity, which could raise FPR (if natural text more frequently resembles canary patterns) or lower TPR (if injected patterns are less distinguishable after paraphrasing).

7. Conclusion

We have examined how canary budget k and detection threshold m interact under a paraphrasing adversary in a chat-based attribution setting. The main finding is that raising k from 2 to 3—while holding $m = 2$ for zero FPR—improves post-paraphrase TPR from 47.5% to 78.5%, a gain of 31.0 percentage points. Imposing stricter thresholds ($m = k$) instead reduces TPR sharply without further reducing FPR beyond what $m = 2$ already achieves. Oracle results confirm that these differences arise in the detection pipeline rather than from injection failures. Taken together, the findings support investing in canary redundancy over threshold tightening when semantic detectors are imperfect and paraphrasing is the primary evasion strategy. Future work should extend this analysis to learned semantic detectors, adaptive adversaries, and multi-domain settings to map out the broader operating characteristics of m-of-k attribution.

References

[1] Ruben Chocron, Doron Jonathan Ben Chayim, Eyal Lenga, Gilad Gressel, Alina Oprea, and Yisroel Mirsky. Who Owns This Agent? Tracing AI Agents Back to Their Owners. *arXiv preprint arXiv:2605.16035v1*, 2026.