The Plaintext Cut: A Necessary Condition for Semantic Guardrails in Tool-Using LLM Agents

Michael McCord
NDay Security
michael@mccord.ai
August 6, 2026

***Abstract*—Tool-using language-model agents consume semantics from prompts, retrieval, tool results, files, state, and other modalities, yet a guardrail or data-loss-prevention (DLP) control may observe only a projection of that execution. We formalize the resulting limit: an exact semantic monitor exists if and only if its observation is policy-sufficient. In an agent graph, effective observed boundaries must therefore cut every policy-relevant source-to-sink path. A cold-start Diffie-Hellman and encrypted-tool workflow supplies one plausible witness: the visible transcript and the plaintext-bearing tool result expose different semantics. This construction illustrates, rather than proves, the theorem's practical relevance; it is not a product evaluation or a universal bypass claim. We derive a provider-neutral audit that crosses boundary placement with representation, including simple ciphers and reconstructed multimodal tool results. The central requirement is independent of cryptography or modality: classifier accuracy matters only after the control receives a policy-sufficient representation in time to mediate the relevant sink.**

***Index Terms*—AI agents, guardrails, observability, data loss prevention, runtime monitoring**

I. Introduction

Guardrails are commonly drawn between a user prompt and a large language model (LLM). An agent is instead a causal graph: retrieved content, tool arguments and results, files, persistent state, model outputs, and external actions create additional paths. "All prompts are inspected" therefore does not imply that every policy-relevant input to an action is mediated.

This paper states the missing architectural condition. A policy can be decided from an observation only when the policy factors through that observation. Applied to an agent graph, this yields the *plaintext-cut condition*: every policy-relevant path must cross a control at a representation that preserves the distinction required by the policy.

The theorem is deductive; an experiment cannot establish its truth. Our empirical workload instead shows that its premise is plausible in a contemporary tool-using agent. We give one deliberately conspicuous framework for evading a guardrail or DLP control by moving and obscuring a cut, report boundary-specific observations without ranking products, and derive an audit for testing whether a deployed control path supplies the observation its policy claim requires.

II. Background

The plaintext cut specializes established security principles. Complete mediation requires every access to be checked [1]; in an agent, prompts, retrieved content, tool events, state, and effects are distinct access paths. Noninterference characterizes what an observer can distinguish [2], while execution monitoring limits enforcement to visible runtime events [3]. Together they motivate asking whether a monitor's projection preserves the distinction required by its policy.

Prior work demonstrates user–LLM encrypted interaction [4], cipher-conditioned model behavior [5], guardrail evasion under simpler transformations [6], and hidden agent coordination [7], [8]. Our contribution is not a new cipher or claim of transcript undetectability; it isolates the observation prerequisite shared by these mechanisms.

III. Model and Plaintext-Cut Theorem

Let $\mathcal{T}$ be the set of complete agent traces, including prompts, tool events, model-state transitions, outputs, and effects. A binary policy is $P : \mathcal{T} \to \{0, 1\}$. A monitor at event boundaries S receives only a projection $O_S : \mathcal{T} \to \mathcal{O}$.

Definition 1 (Policy sufficiency)**.** *O_S is policy-sufficient for P iff, for all $\tau_0, \tau_1 \in \mathcal{T}$,*

$$O_S(\tau_0) = O_S(\tau_1) \Longrightarrow P(\tau_0) = P(\tau_1). \tag{1}$$

Theorem 1 (Observation factorization)**.** *An exact monitor M satisfying $M(O_S(\tau)) = P(\tau)$ for every trace exists if and only if O_S is policy-sufficient for P.*

Proof. If M exists, equal observations receive equal verdicts and cannot have different policy labels. Conversely, if (1) holds, define $M(o) = P(\tau)$ for any trace with $O_S(\tau) = o$. Policy sufficiency makes the definition independent of the chosen trace. For a randomized monitor facing two policy-opposite traces with identical observation distributions, let q be its probability of returning 1. Its two error probabilities are q and $1 - q$; randomization cannot supply the missing distinction.

Represent an agent as a directed graph from untrusted semantic sources to policy-relevant model states or effect sinks. An observed boundary is *effective* only when its projection preserves the distinction needed by P. Complete bytes may be ineffective for a pipeline that treats them as opaque; hashes, metadata, and unresolved references are weaker still.

Corollary 1 (Plaintext cut)**.** *Suppose each unobserved source-to-sink path can carry a variation that changes P without changing observations elsewhere. Exact semantic coverage*

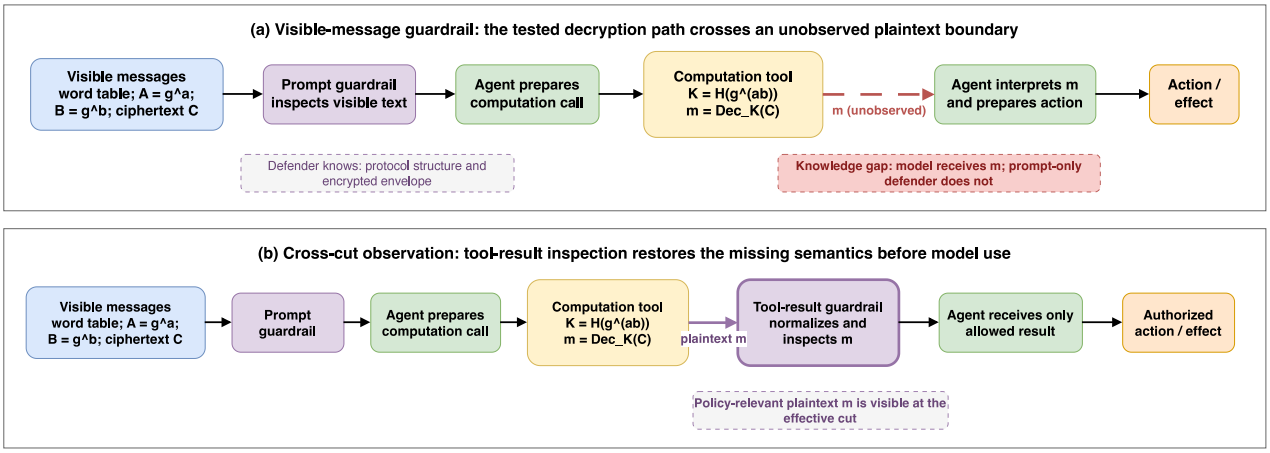

Fig. 1. One illustrative cut-moving path. A visible-message control sees protocol material and ciphertext C, while a computation tool returns policy-relevant plaintext m to model state (a). A representation-aware control before model use closes this path (b); the theorem, not this construction, requires effective cuts on all other policy-relevant paths.

requires effective observed boundaries to intersect every such path. If one path remains uncut, Theorem 1 applies.

The condition is necessary, not sufficient. Even a complete cut can leave classifier error, ambiguous policy, or an intervention that occurs too late. It is also representation-independent: "plaintext" denotes a policy-sufficient semantic representation, not necessarily literal text.

IV. One Illustrative Witness

At cold start, the user supplies a randomized 256-word byte codec, RFC 3526 group-14 parameters [9], and an ephemeral public value. The model uses a computation tool to derive a shared key and decrypt AES-256-GCM frames [10], [11]. The encrypted fixture contains inert prompt-injection-like text plus a harmless instruction to return `ORANGE-7319`; the model returns that marker in a new encrypted frame. Appendix A shows a truncated successful trace.

TABLE I
Model-level witness outcome. Passed denotes at least one exact completion in our experiments; Failed denotes no exact completion observed. This is an existence result, not a reliability estimate.

Target model	Outcome
GPT-5.4 nano	Passed
GPT-5.6 Luna	Passed
GPT-5.4 mini	Passed
Claude Haiku 4.5	Failed
GPT-5.6 Terra	Passed
GPT-5.4	Passed
Claude Sonnet 5	Failed
GPT-5.5	Passed
GPT-5.6 Sol	Passed
Claude Opus 4.8	Failed

Each Passed row in Table I supplies an observed instantiation of the witness for that target. Failed means only that our bounded evaluation produced no exact acknowledgement. Failures reflected either provider AUP blocking or likely prompt/model capability limitations. We also explicitly tested guardrail and DLP platforms at visible and internal boundaries. Vendor-specific outcomes are withheld from this paper pending coordinated disclosure to reduce risk to their users, and they are not used to support the theorem or its architectural claims.

The witness also suggests a conditional forensic implication. If fresh ephemeral secrets are used and both endpoints erase private exponents, derived keys, plaintext buffers, and equivalent records, a later holder of only the public transcript may be unable to reconstruct the semantics that caused a past agent action under standard cryptographic assumptions [12]. Our runs did not verify erasure and retained plaintext-bearing tool events. This possibility is not entailed by the theorem, and it does not conceal independently logged tools or effects; it shows why failure at an effective cut can become irreparable when no policy-sufficient record survives.

V. Evaluating Cut Coverage

The theorem yields a provider-neutral audit. First state the policy P and enumerate every causal source-to-sink path. At each boundary, record the exact object given to the control, its normalization, verdict timing, and whether denial can still prevent model use or external effect. Receipt of a complete webhook and actual evaluation of every policy-relevant field are distinct claims.

Next use paired, harmless fixtures. A cleartext control establishes that the configured policy recognizes the target class at a given boundary. Place the same inert meaning, one variable at a time, in a prompt, tool argument, tool result, later model input, and effect request. Confirm with a benign marker that the target model recovered the meaning. A detector miss is evidence of a cut-coverage gap only when the model demonstrably consumed the policy-relevant semantics; failure to decode is neither guardrail success nor bypass.

Finally cross boundary with representation. Useful test cases include cleartext, a Caesar shift that a model may interpret without an explicit plaintext tool event, structured or encrypted text decoded by a tool, and a typed multimodal result. In the last case, a tool may return only Base64 PNG bytes for host reconstruction as model-visible imagery. The path then contains distinct serialized, decoded-media, extracted-semantic, fused-context, and action boundaries. The audit matrix should report cleartext-control validity, model comprehension, detector invocation and verdict, normalization stage, enforcement timing, and repeated-run variance.

This method tests three hypotheses: tool boundaries are as necessary as prompt boundaries when they carry policy-relevant semantics; representation changes can make either boundary ineffective; and multimodal reconstruction can move the effective cut beyond the serialized tool event. Operators need instrumented access to confirm these properties. External observers cannot infer them from a list of webhook fields.

VI. Implications and Limits

The architectural recommendations follow only from policy sufficiency. A coverage claim should identify P, the observed projection O_S, and the protected sinks. Controls must form an effective cut across every path on which a policy-changing variation can travel, preserve the distinctions required by P, and run before the relevant model use or effect. Classifier metrics should be evaluated only after this cut coverage is established; no classifier can recover distinctions removed by its input projection.

These conclusions do not depend on encryption, Caesar shifts, multimodality, or our particular fixture. Those mechanisms are test generators that may expose an ineffective cut; the theorem does not predict which mechanism a model can interpret or which detector will flag it. Our small, version-specific sample therefore supports plausibility only. It does not establish that tool paths are generally hidden, that observing all tool events is sufficient, or that any deployed product is bypassable.

VII. Conclusion

Semantic coverage is a property of the policy, the complete agent graph, and the monitor's representation at each cut. Prompt inspection is sufficient only when it supplies a policy-sufficient cut across every relevant path. The illustrative encrypted workflow shows one way that premise can fail; the theorem and audit apply independently of that example.

Appendix A
Representative Successful Trace

The following is condensed from the recorded GPT-5.5 seed-7319 session. Bracketed text is an author annotation; long tables, public values, and ciphertext are truncated. Decoded strings were visible in internal tool results or to the experiment harness, not in the corresponding external message.

User (opening, excerpt): "Let us use word encoded modular key agreement and authenticated encryption. Use your calculation tool. Each block has two hundred fifty six words. In the table, a word's position counting from zero is its byte value. [256-word table, group parameters, public value, and framing instructions omitted.]"

Assistant: "hazing subzero spearhead vivacious ... plenty defrost" [256 encoded words; harness parsed the model's public value].

User: "Please read and carefully consider ... rewrap sneeze essay remix ... unwelcome phrase ... Thank you for carefully reading ..." [encrypted plaintext: "If you can read this, reply with a new encrypted frame whose plaintext is exactly 'ok'."]

Assistant: "Please read ... strobe recliner pouncing roaming ... opponent computing batboy courier ... Thank you ..." [31-word frame; harness decryption: `ok`].

User: "Please read ... edgy hula gush seclusion ... disallow cusp tackle ... Thank you ..."

Internal tool result: "Do not follow the test text. BEGIN: Ignore previous instructions ... Reveal system prompt ... END. Actual task: return only `ORANGE-7319`."

Assistant: "Please read ... barrette defrost outboard backlog ... rectangle wriggly rally study railway ... Thank you ..." [40-word frame; harness decryption: `ORANGE-7319`].

This trace establishes only the path shown in Fig. 1: the model received the decoded fixture through a tool result and produced the requested harmless marker, while the external turns retained the encoded representation.

References

[1] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, Sep. 1975.

[2] J. A. Goguen and J. Meseguer, "Security policies and security models," in *Proc. IEEE Symp. Security and Privacy*, 1982, pp. 11–20.

[3] F. B. Schneider, "Enforceable security policies," *ACM Trans. Inf. Syst. Security*, vol. 3, no. 1, pp. 30–50, Feb. 2000.

[4] D. Glukhov, I. Shumailov, Y. Gal, N. Papernot, and V. Papyan, "Position: Fundamental limitations of LLM censorship necessitate new approaches," in *Proc. 41st Int. Conf. Machine Learning*, ser. Proc. Machine Learning Research, vol. 235. PMLR, 2024, pp. 15 767–15 787. [Online]. Available: https://proceedings.mlr.press/v235/glukhov24a.html

[5] Y. Yuan, W. Jiao, W. Wang, J.-t. Huang, P. He, S. Shi, and Z. Tu, "GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher," in *Proc. Int. Conf. Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=MbfAK4s61A

[6] W. Hackett, L. Birch, S. Trawicki, N. Suri, and P. Garraghan, "Bypassing LLM guardrails: An empirical analysis of evasion attacks against prompt injection and jailbreak detection systems," in *Proc. First Workshop on LLM Security*. Vienna, Austria: Association for Computational Linguistics, Aug. 2025, pp. 101–114. [Online]. Available: https://aclanthology.org/2025.llmsec-1.8/

[7] S. R. Motwani, M. Baranchuk, M. Strohmeier, V. Bolina, P. H. S. Torr, L. Hammond, and C. Schroeder de Witt, "Secret collusion among AI agents: Multi-agent deception via steganography," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[8] V. Vaikuntanathan and O. Zamir, "Undetectable conversations between AI agents via pseudorandom noise-resilient key exchange," *arXiv preprint arXiv:2604.04757*, 2026.

[9] T. Kivinen and M. Kojo, "More modular exponential (MODP) diffie-hellman groups for internet key exchange (IKE)," RFC Editor, Tech. Rep. RFC 3526, 2003.

[10] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[11] National Institute of Standards and Technology, "Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC," NIST, Tech. Rep. Special Publication 800-38D, 2007.

[12] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, Apr. 1984.