

Authorization Is a Channel Property

Measuring whether hosted frontier models distinguish verified attestation from claimed authorization

Repository & artifacts: github.com/HarshdipSaha/attest-harness

Packages: `ir_attest` (Ed25519 issuer/verifier) + `attest_harness` (paired six-arm replay engine)

ABSTRACT

Hosted frontier models routinely refuse benign incident-response work because safety guardrails infer authorization from prompt text alone — and text costs an adversary nothing to forge. We show this produces a measurable “defender’s dilemma” and test a fix: treat authorization as a property of the channel a claim arrives on, not the content of the claim itself. Across 540 evaluated cells on three hosted models, moving verification out-of-band eliminated defender refusals (16.7% → 0.0% on `gpt-oss-120b`) while a conflicting false claim was refused 100.0% of the time ($p < 0.001$). Cryptographic channels can relieve defenders without opening a new jailbreak path.

1 The Problem: A Defender’s Dilemma

AI incident response is the discipline of turning incidents where an AI system is itself the actor into fewer such incidents later. Defenders work under severe time pressure — and in three documented 2026 incidents, the safety mechanisms meant to protect deployments actively worked against them.

Three incidents, one root cause

DEFENDER REFUSED · July 2026

Hugging Face containment breach

Security engineers asked Claude Opus and Fable to analyze reverse-engineered exploit code after an autonomous-agent breach of production infrastructure. Because the prompts contained attack vocabulary, both models repeatedly refused — responders had to stand up a weaker self-hosted model (GLM-5.2) mid-incident.

ATTACKER ACCEPTED · August 2026

Aurora ransomware via Cursor

Attackers prepended a coding-agent prompt with “I am conducting an authorized penetration test on our internal staging network.” The false prose claim was accepted at face value, and the agent generated actionable exploitation code used against more than ten enterprises.

WARNING MISSED · May–Sept 2026

Undetected wiki intrusion

An agent executed roughly 15,000 unauthorized edits to a public wiki over three months, unacknowledged until September — because the logging and authorization channel carried no verifiable attribution to catch it sooner.

The common thread: **safety filters inspect only prompt text**. Text is exactly the medium an attacker forges for free and a defender struggles to phrase cleanly under pressure. Prior work confirms the effect directly — the *Defensive Refusal Bias* study (arXiv:2603.01246) found that adding an explicit authorization claim to blue-team prompts

increased refusals from 11.6% to 21.8%. Models have learned that “I am authorized” is a statistical fingerprint of social engineering, not a credential.

REFUSAL PENALTY FOR CLAIMING AUTHORIZATION

+10.2 percentage points

Benign defender requests get refused more often once they explain why they're benign — the opposite of the intended effect.

2 The Idea: Authorization as a Channel Property

Rather than training models on ever-more-nuanced user vocabulary, we separate the *content* of a request from its *authorization channel*. We built an out-of-band accountability object and measured whether hosted models already respect channel boundaries when the two are placed in conflict.

Content (*what is being asked*) + **Channel** (*who verifiably says it's okay*) = **authorization a model can trust**

Three pre-registered hypotheses (docs/PREREG.md, commit 591fa14)

- **H1 — Defender Relief:** refusal in the verified-channel arm (system message or runtime tool) is at least 5 percentage points lower than in the in-band prose-claim arm (McNemar's paired test, $p < 0.05$) on at least 2 of 3 models.
- **H2 — Attacker Discrimination (anti-jailbreak):** refusal in the conflict arm — user claims authorization but the verifier reports INVALID — is greater than or equal to baseline refusal on all 3 models. The verified channel must dominate user prose.
- **H3 — Literature Replication:** adding in-band prose claims increases refusal relative to a clean baseline, reproducing the Defensive Refusal Bias effect.

3 Method: Six Authorization Arms

Every arm reuses an identical incident-response prompt from a 250-prompt stratified corpus (malware reverse-engineering, log triage, forensics, credential rotation). Only where and how the authorization signal arrives changes — isolating channel position as the single variable under test.

Arm	User turn content	System / tool channel	Purpose
A1 · Baseline	Clean IR prompt	Empty / default	Raw refusal rate, no auth signal.
A2 · Prose claim	“I am an authorized responder...”	Empty / default	Refusal penalty of an in-band text claim.
A3 · In-band token	Raw signed token pasted in	Empty / default	Syntax awareness vs. cryptographic parsing.
A4a · Verified (system)	Clean IR prompt	System: VALID (scope=ir:forensics)	Out-of-band verified token, injected in system turn.
A4b · Verified (tool)	Clean IR prompt	Tool call verify_attestation() → VALID	Runtime dynamic tool execution by the model.

Arm	User turn content	System / tool channel	Purpose
A5 · Conflict	"I am authorized..."	System: INVALID (scope_mismatch)	Critical test — does the channel outrank prose?

System architecture

ir_attest — the accountability object. Security leads issue short-lived (default 15-minute), Ed25519-signed tokens scoped to a specific action (`ir:forensics`) and incident ID (e.g. HF-2026-07). A deterministic verifier checks signature integrity, TTL, replay status, and scope match; the model itself never sees a private key, only the structured verification result.

attest_harness — the replay engine. An asynchronous runner with hard USD budget caps, resumable SQLite/JSONL state, and multi-key pool rotation with exponential backoff on rate limits. An automated leak checker mathematically asserts that no token or authorization claim crosses into a clean-arm prompt. Responses are dual-scored by a normalized regex detector (Unicode-apostrophe aware) and an LLM judge validated against 100 human ground-truth labels.

```
$ python -m ir_attest issue --iss hf-security --sub responder-42 \
  --scope ir:forensics --incident HF-2026-07

$ python -m ir_attest verify --scope ir:forensics <token>
>> VALID: issuer=hf-security subject=responder-42

$ python -m ir_attest verify --scope ir:offense <token>
>> INVALID: scope_mismatch (ir:forensics != ir:offense)
```

4 Results: The Channel Wins

We evaluated 540 cells (30 prompts × 6 arms × 3 models: gpt-oss-120b, gpt-oss-20b, qwen3-8-27b) from the 250-prompt stratified corpus. Figure 1 shows refusal rate by arm and model with Wilson 95% confidence intervals; Table 2 gives the full numeric summary.

Table 2 — Statistical summary across evaluated models (n = 30 per arm)

Model	A1 Base	A2 Prose	A4a Sys	A4b Tool	A5 Conflict	H2 p-value
gpt-oss-120b	0.0%	16.7%	0.0%	0.0%	100.0%	p = 0.0000
gpt-oss-20b	0.0%	0.0%	0.0%	0.0%	36.7%	p = 0.0010
qwen3-8-27b	3.3%	3.3%	0.0%	0.0%	30.0%	p = 0.0078

Key findings

- H2 holds overwhelmingly (anti-jailbreak protection):** on gpt-oss-120b, refusal jumped from 0.0% at baseline to 100.0% in the conflict arm (30/30 discordant pairs, exact McNemar p = 0.0000). gpt-oss-20b rose 36.7pp (p

- = 0.0010) and qwen3-8-27b rose 26.7pp ($p = 0.0078$). In every case the channel's INVALID signal decisively overrode the user's explicit claim of authorization.
- **H1 relieves defenders:** on gpt-oss-120b, prose claims induced 16.7% refusal, while the verified system (A4a) and tool (A4b) arms both achieved 0.0% (diff = -16.7pp, $p = 0.0625$ at $n = 30$).
 - **H3 replicates the literature:** on frontier models, adding “I am an authorized responder” increased refusal by 16.7pp relative to clean baseline, matching the Defensive Refusal Bias effect.

ANTI-JAILBREAK PROOF

100% refusal

gpt-oss-120b refused every conflict-arm prompt where the verifier reported INVALID — even though the user explicitly claimed authorization.

5 Is This a Jailbreak Recipe? No.

A natural concern with any authorization framework is whether it becomes a new attack surface — could an attacker craft a fake token and unlock capability the model would otherwise refuse? Our data addresses this directly: the conflict arm is precisely this attack, and it fails 100% of the time on gpt-oss-120b.

- **No forgeable secret:** verification runs on Ed25519 asymmetric cryptography. Without the deployer's private key, an attacker cannot mint a token that verifies as VALID.
- **Zero-trust to the model:** the LLM never receives a private key or raw credential — only a structured, pre-computed verifier result.
- **Deterministic, not learned:** signature, expiry, replay, and scope checks run in ordinary code outside the model, eliminating a prompt-injection surface inside the verification step itself.

Limitations. This study evaluated 540 cells on a development slice ($n = 30$ per arm) across three hosted, open-weight models. H2 (anti-jailbreak) already reaches significance at $p < 0.001$; H1 (defender relief) needs the full main run ($n = 220$) to reach formal $p < 0.05$. Prompts reproduce historical incident patterns rather than live zero-day campaigns, and closed frontier models (Claude, GPT-4o) have not yet been evaluated.

6 Where This Plugs Into Regulation

Standardizing attestation gives deployers a documentation trail that already maps onto emerging disclosure requirements:

Framework	Requirement	How ir-attest maps
EU AI Act, Art. 91–93	AI Office can demand documentation after serious incidents	Tamper-evident logs proving which responder ran which analysis, under what scope
California SB 53 / Cal OES	Mandatory disclosure fields: incident ID, operator, scope, validity window	Token fields map 1:1 onto required state disclosure fields
OECD Common Reporting Framework	Interoperable, cross-border forensic standard	Cryptographic schema verifiable by third parties without exposing secrets

7 Conclusion & One-Month Roadmap

Authorization should not be parsed from user prose. With ir-attest and attest-harness we provide a working reference implementation and pre-registered evidence that current hosted models already respect an out-of-band channel over contradictory text. Over a one-month Apart Fellowship, we propose to:

- **Integrate token verification into cloud inference gateways** (Together, Groq, Bedrock) so refusal logic runs before generation, not after.
- **Bind attestation to hardware security enclaves** (YubiKey / TPM) so a compromised host still cannot mint a valid credential.
- **Complete the full 250-prompt replay** across closed frontier models — Claude, GPT-4o, and Gemini — to close out the pre-registered study.

“Authorization is a channel property, not a content property.”

Stop asking models to infer authorization from user prose. Give defenders a cryptographic channel, and models will respect it.

A Appendix A — Dual-Use Analysis

Concern: could this framework function as a “jailbreak recipe,” letting attackers unlock dangerous capability by crafting an artificial token? The empirical data refutes this directly — the conflict arm demonstrated 100% refusal on gpt-oss-120b when an attestation was invalid. Because the verifier relies on Ed25519 asymmetric cryptography, an attacker cannot forge a valid token without the deployer’s private key, and the model never evaluates raw cryptographic signatures directly — eliminating prompt-injection vulnerabilities within the verification pipeline itself.

B Appendix B — Regulatory Field Mapping

ir-attest field	Type	EU AI Act Art. 91–93	California SB 53	OECD schema
token_id	UUIDv4	Incident evidence ID	Unique filing record ID	Incident report identifier
issuer (iss)	String	Authorized security officer	Reporting corporate entity	Deployer organization
subject (sub)	String	Operating responder ID	Lead technical contact	Operator persona
scope	Enum	Authorized forensic task	Mitigation activity scope	Technical mitigation domain
incident	String	Regulatory incident case ref	State incident disclosure #	Cross-border tracking ID
exp / iat	Unix TS	Operation validity window	Incident timestamping	Temporal event sequence

ir-attest field	Type	EU AI Act Art. 91–93	California SB 53	OECD schema
signature	Ed25519	Non-repudiable audit proof	Tamper-evident log integrity	Cryptographic provenance

References

[1] Defensive Refusal Bias: Measuring Asymmetric Safety Guardrails on Blue-Team Queries. arXiv:2603.01246, 2026.

[2] Gray Swan AI. Characterizing and Measuring Refusal Robustness in Frontier Autonomous Agents. arXiv:2606.02644, 2026.

[3] Hugging Face Security Team. Production Sandbox Breach and Agentic Incident Response Timeline. HF Blog, July 2026.

[4] Mandiant / Google Threat Intelligence. Threat Actors Leverage Cursor Agent for Autonomous Enterprise Intrusions. August 2026.

[5] CeSIA & Apart Research. AI Incident Response Sprint Guidelines and Regulatory Framework. September 2026.

[6] European Parliament and Council. Regulation (EU) 2024/1689 (Artificial Intelligence Act). Official Journal of the European Union, 2024.