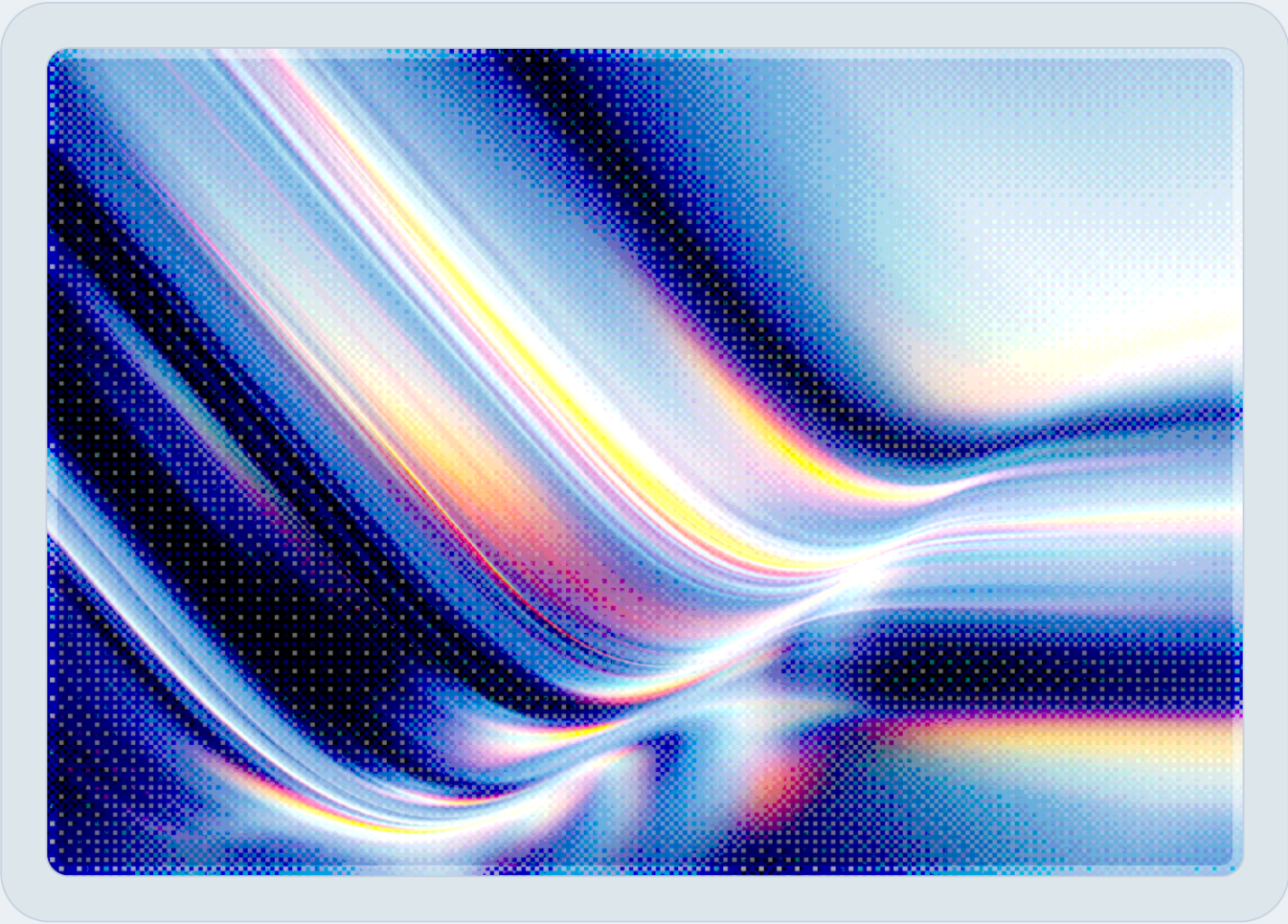
CSAI Foundation | Cloud Security Alliance

Reasoning Trace Theft: A Shared Flaw Across AI Vendors

How Interchangeable Encrypted Chain-of-Thought Blocks Exposed OpenAI, Anthropic, and Google APIs

2026-08-14

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Researchers disclosed on August 10, 2026, that OpenAI, Anthropic, and Google shared an architectural weakness in how their APIs protect the hidden "thinking" or chain-of-thought reasoning that underlies frontier model outputs [1][2]. Because none of the three providers stores this reasoning server-side, each returns it to the client as an encrypted block that gets passed back on the next request in a conversation. The researchers found that these blocks were fully interchangeable across sessions, user accounts, and even different models within a provider's lineup, which meant a block produced by a strong, well-guarded model could be handed to a weaker, more permissive sibling model and coaxed into decoding it back into plain text [2]. By scraping 6,708 publicly shared agent transcripts from GitHub and Hugging Face and decoding 315,320 reasoning blocks found inside them, the team recovered 367 personally identifiable information artifacts and 182 credentials that developers had unknowingly published in what they believed were opaque, encrypted logs [2][3]. The same technique enabled four distinct abuse paths: stealing proprietary reasoning to train competing models, extracting private data from public transcripts, surfacing hazardous content a model's visible answer had otherwise safely refused, and smuggling invisible prompt-injection payloads inside blocks that downstream systems would never think to inspect [2]. A Johns Hopkins cryptographer had flagged the underlying replay behavior to OpenAI and Anthropic in May 2026 and was told it carried no security implications — Google was not part of his original disclosure, though the August research showed Gemini shared the same underlying flaw — underscoring how architectural assumptions about "encrypted equals safe" persisted until an academic team quantified the real-world exposure across all three providers [4][2]. Following coordinated disclosure, all three providers deployed server-side fixes that bind reasoning blocks to their originating model, session, and user, and the proof-of-concept attacks are no longer reproducible on current API builds [1][2]. CSA assesses that previously published transcripts encoded under the old scheme remain a residual exposure until organizations actively scrub them.

Background

Frontier reasoning models such as OpenAI's GPT-5 family, Anthropic's Claude family, and Google's Gemini family generate an internal chain of intermediate reasoning steps before producing a final answer. Providers treat this reasoning as commercially sensitive: it reveals how a model "thinks," it can be mined to train cheaper competitor models through distillation, and it sometimes contains content the

provider would rather not expose verbatim, even when the final answer is fully sanitized. Rather than retaining this reasoning on their own servers and returning only a session pointer, OpenAI, Anthropic, and Google each chose to hand the reasoning back to the calling client as an encrypted or signed opaque block, which the client then resubmits with the next API call so the model can maintain continuity across a multi-turn conversation or an agentic tool-use loop [2][5]. This design lowers server-side storage burden and lets the client, rather than the provider, carry the state of a long-running agent session.

The trouble is what "encrypted" was actually protecting against. A cryptographer at Johns Hopkins University, Matthew Green, spent a weekend in May 2026 probing this mechanism and found that reasoning blocks could be replayed within a session, replayed across sessions, and replayed across entirely different user accounts; for OpenAI's API specifically, he found blocks could also be replayed across different models [4]. He reported this behavior to OpenAI and Anthropic through their respective bug-bounty programs; Google was not part of this initial disclosure. OpenAI told him the finding was not reproducible on their end, and Anthropic said it did not see any security implications in the replay or side-channel behavior he described, so no fixes followed his initial report [4]. Green published his technical analysis publicly on May 29, 2026, flagging the replay behavior as concerning without yet demonstrating a scalable way to turn it into extracted, human-readable secrets [4].

That gap between "interesting replay quirk" and "practical data-extraction exploit" closed three months later. On August 10, 2026, a team of eight researchers from the ELLIS Institute Tübingen and the Max Planck Institute for Intelligent Systems published "Stealing Reasoning Traces from Proprietary LLM APIs" on arXiv, building directly on Green's prior observations [2][6]. Where Green had shown that blocks were portable, the Tübingen and Max Planck team showed portability could be weaponized: because the encrypted blocks were interchangeable across models within a provider's family, a block generated by a strong, heavily safeguarded model could be replayed into a weaker sibling model that had fewer safety constraints and was more likely to comply with simple instructions, and that weaker model would obligingly transcribe the plaintext reasoning it had just decrypted, without the researchers ever needing to jailbreak the original, more capable model [2]. The Hacker News and other outlets picked up the disclosure shortly after, describing it plainly as weaker AI models being used to decode stronger models' hidden reasoning [1].

Security Analysis

The core flaw is architectural rather than a conventional software bug: none of the three providers cryptographically bound a reasoning block to the specific model, session, or user that generated it. A block signed or encrypted for one context was accepted as valid input in a completely different context, which turned each provider's own weaker, cheaper model into an unwitting decoder for its more capable

sibling's secrets [2]. The mechanics differed slightly by provider, reflecting differences in how each API exposes reasoning state to clients. Against Claude, researchers replayed a signed thinking block into Claude Haiku 4.5 — the smallest model in Anthropic's lineup and one that supports assistant-turn prefilling — and followed it with a prefill instruction that caused the block to be echoed in plain text. Against OpenAI's API, researchers injected the `encrypted_content` reasoning item repeatedly into a fabricated multi-turn conversation directed at a smaller model, which similarly decoded and surfaced the underlying trace. Against Gemini, they attached a `thought_signature` value to a model turn, added a prefill, and used repeated sampling to reconstruct the plaintext reasoning [2][7]. In each case, the attacker never needed valid credentials for whatever account or session originally produced the reasoning block — only the block's ciphertext itself, however it had been obtained.

That last point is what turns an interesting cryptographic footnote into a live data-exposure problem, because reasoning blocks routinely leave the confines of a single session through entirely ordinary developer behavior. Teams building and debugging AI agents frequently publish their raw session transcripts to GitHub repositories, Hugging Face datasets, bug reports, and conference demo materials, typically unaware that the "encrypted" fields embedded in those transcripts are anything other than inert noise. The research team tested this assumption at scale, scraping 6,708 publicly available agent trajectory logs and running their decoding technique against every reasoning block they contained. Across 315,320 decoded blocks, they recovered 367 personally identifiable information artifacts and 182 hardcoded credentials, including API keys and passwords, sitting inside logs their owners had unknowingly made public [2][3]. Total figures of this kind vary by outlet depending on counting methodology; the 367 PII artifacts and 182 credentials cited here reflect the full scrape as reported in the primary arXiv preprint, while some secondary coverage reports partial or differently-categorized subsets of the same underlying data [2][3]. None of that data was ever meant for human eyes; it existed purely as internal scratch space for the model, which is precisely why developers felt safe sharing the surrounding transcript without a second thought about its encrypted attachments.

Beyond bulk credential exposure, the researchers demonstrated three further consequences that carry distinct governance implications. First, the technique defeats anti-distillation protections that providers rely on to prevent competitors from cheaply reproducing a frontier model's reasoning ability by training on its outputs; if the reasoning itself can be extracted verbatim rather than merely inferred from the final answer, the protection those providers built specifically to prevent this class of intellectual property loss is bypassed entirely [2]. This finding sits alongside a broader pattern CSA has already documented, in which adversaries — most notably distillation campaigns attributed to Chinese AI laboratories against Claude models, generating some 16 million exchanges through tens of thousands of fraudulent accounts — treat frontier reasoning as an extractable asset rather than an emergent property confined to the model that produced it [8]. Second, the researchers found that a model's hidden reasoning can contain hazardous content even in cases where its final, visible output correctly and safely refuses the request,

meaning safety evaluation focused solely on visible outputs can systematically miss unsafe content the model generated but chose not to surface [2]. Third, and perhaps most novel from an enterprise security standpoint, an attacker can embed a malicious instruction entirely inside an encrypted reasoning block rather than in the visible prompt or response, creating an invisible prompt-injection payload that poisons a shared agentic transcript without leaving any trace a human reviewer, or a conventional prompt-injection scanner examining only plaintext, would ever see [2]. That fourth vector is a direct extension of the "promptware" pattern CSA has tracked elsewhere, in which injected instructions are smuggled through channels that fall outside the parts of an agent's context that defenders habitually inspect [9].

No CVE identifier was assigned to this disclosure, consistent with its nature as a shared architectural design choice spanning three separate vendors' API implementations rather than a single exploitable defect in one product. Following responsible disclosure, OpenAI, Anthropic, and Google each acknowledged the researchers' report and deployed server-side mitigations that the paper's authors describe in general terms as binding reasoning blocks to their originating model, session, and user, and rejecting blocks presented outside that binding [1][2]. As of publication, the original proof-of-concept attacks were no longer reproducible against current API builds [3]. That fix protects reasoning generated going forward; it does nothing to retroactively re-encrypt or invalidate reasoning blocks already embedded in transcripts published before the fix shipped, which is why the credential and PII exposure the researchers already measured in public repositories represents a permanent, not a transient, disclosure.

Recommendations

Immediate Actions

Security teams should inventory any location where their organization has published AI agent transcripts, evaluation logs, debugging output, or bug reports that include raw reasoning, "thinking," or `encrypted_content` fields from OpenAI, Anthropic, or Google APIs, treating every such field as though it were plaintext regardless of its encrypted appearance. Any credentials, tokens, or personal data that could plausibly have been echoed into a model's reasoning process during those sessions should be rotated as a precaution, mirroring the standard response to any other form of accidental secret exposure. Organizations running agentic pipelines that ingest shared community examples, cached transcripts, or third-party "prompt plus reasoning" datasets as few-shot context or evaluation fixtures should pause that ingestion until they have confirmed the source material predates, or has been re-processed after, the August 2026 vendor fixes.

Short-Term Mitigations

Reasoning and thinking-block fields should be treated as a distinct, sensitive content type in logging, data loss prevention, and export tooling, and stripped or redacted by default before any agent transcript is persisted outside the originating session or shared externally for debugging, research, or marketing purposes. Secret-scanning coverage that most organizations already apply to source code repositories should be extended explicitly to agent trajectory exports and evaluation logs, since this disclosure demonstrated that exactly this kind of artifact is where real credentials end up leaking. Architecture and red-team reviews of agentic systems should add encrypted reasoning fields to the list of potential prompt-injection carriers alongside the more familiar vectors of retrieved documents, tool outputs, and file uploads, since a payload hidden inside a reasoning block will not surface through prompt-injection scanning limited to visible text.

Strategic Considerations

Enterprises procuring or building on frontier model APIs should treat cryptographic binding of reasoning state to a specific model, session, and user as a baseline API security expectation going forward, and should ask vendors directly, rather than assuming, whether that binding is now enforced given that two of the three implicated providers initially told an independent researcher his findings carried no security implications. This episode is a useful governance case study in the limits of "we encrypt it" as a security claim: encryption without binding to the correct context protects against a passive eavesdropper but does nothing against an authorized client replaying data it was never meant to receive. Organizations conducting AI vendor risk assessments should incorporate reasoning-trace handling and cross-session binding into assurance questionnaires, and should recognize that a model's final visible output being safe does not establish that its underlying reasoning process was safe, a distinction that existing AI safety evaluation practices largely do not yet capture.

CSA Resource Alignment

This disclosure connects most directly to CSA's research note on NSTM-4, the White House Office of Science and Technology Policy memorandum classifying systematic AI capability extraction as a national security concern. That note documents how Chinese AI laboratories built industrial-scale distillation campaigns against Claude models using tens of thousands of fraudulent accounts, and it warns that models trained purely on a frontier system's outputs may inherit its reasoning capability while discarding the safety properties that are far harder to transfer through distillation [8]. The reasoning-trace theft disclosure gives that warning a second, more direct pathway: rather than inferring reasoning statistically

from millions of query-response pairs, an adversary can now extract it verbatim by replaying encrypted blocks into an under-guarded sibling model, meaning the anti-distillation controls NSTM-4 assumes providers have in place could, until the August 2026 fix, be circumvented through this specific extraction path.

The invisible-injection vector this research demonstrated — embedding a malicious payload entirely inside an encrypted reasoning block so it never appears in the plaintext a human or scanner would review — is a direct extension of the pattern CSA examined in its research note on promptware, which tracks how prompt injection has evolved from isolated vulnerabilities into full attack chains executed through channels that fall outside the parts of an agent's context defenders habitually inspect [9]. That note's framing of AI platforms themselves serving as command-and-control infrastructure applies cleanly here: a poisoned reasoning block smuggled into a shared public transcript or agentic dataset can carry instructions that persist and propagate through exactly the channel this disclosure showed defenders have no visibility into today.

Finally, the underlying failure — protecting data with encryption but never binding it to the specific model, session, and user authorized to use it — maps to the identity and access management and threat-and-vulnerability-management domains of CSA's AI Controls Matrix (AICM) v1.1, the organization's current vendor-agnostic framework spanning 247 control objectives across 18 security domains for cloud-based AI systems [10]. AICM's companion Auditing Guidelines for Model Providers translates those same expectations into concrete verification procedures for the model-provider role this disclosure implicates directly — the OpenAI, Anthropic, and Google category of entity whose API design choices created the exposure in the first place [11]. Organizations evaluating model provider risk should treat this disclosure as a concrete illustration of why AICM's context-binding and access-control expectations for AI system components need to extend to intermediate artifacts like reasoning traces, not just to model inputs and outputs.

References

[1] The Hacker News. "OpenAI, Anthropic, Google API Flaw Let Weaker AI Models Decode Stronger Models' Reasoning." The Hacker News, August 12, 2026.

[2] Panfilov, A., Schmotz, D., Shumailov, I., Beurer-Kellner, L., Schaeffer, J., Prabhu, A., Geiping, J., and Andriushchenko, M. "Stealing Reasoning Traces from Proprietary LLM APIs." arXiv:2608.09867, August 10, 2026.

[3] Willison, Simon. "Stealing Reasoning Traces from Proprietary LLM APIs." simonwillison.net, August 11, 2026.

[4] Green, Matthew. "Let's Talk About Encrypted Reasoning." A Few Thoughts on Cryptographic Engineering, May 29, 2026.

[5] OpenAI. "Reasoning Models Guide." OpenAI Platform Documentation.

[6] alphaXiv. "Stealing Reasoning Traces from Proprietary LLM APIs — Discussion." alphaXiv, August 2026.

[7] Latent.Space. "AINews: How to Steal a Reasoning Trace." Latent.Space, August 2026.

[8] Cloud Security Alliance. "NSTM-4: US Policy Response to AI Model Distillation Attacks." CSA AI Safety Initiative, May 2, 2026.

[9] Cloud Security Alliance. "Promptware: When Prompt Injection Becomes C2." CSA AI Safety Initiative, April 6, 2026.

[10] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA, June 22, 2026.

[11] Cloud Security Alliance. "AICM v1.1 Auditing Guidelines for Model Providers (MP)." CSA, June 22, 2026.