TTTPS: A Two-Layer Verification Architecture for Autonomous AI Agents

Cryptographic timestamp sealing and formal-logic verification for AI agent decisions

Peter Jang *(pen name: Heime Jorgen)*
Independent research | heime.jorgen@proton.me

draft-helmprotocol-tttps-08 | IETF Independent Submission stream | 12 August 2026 | pending author approval

Specification. draft-helmprotocol-tttps-08, IETF Independent Submission stream, datatracker.ietf.org/doc/draft-helmprotocol-tttps/. Stream history, measured directly against the datatracker document-history page (12 August 2026): submitted 14 March 2026 as -00; ISE stream assigned, *Submission Received*, 30 June 2026; *In ISE Review* 25 July 2026, same day *Response to Review Needed*; revisions -07 (26 July) and -08 (29 July) followed in direct response.

Interim presentation. Presented live at the IETF Dispatch/SecDispatch interim session, 30 June 2026 – the same day as the ISE stream assignment above. Two local artifacts confirm this independently: the session slide deck (draft-helmprotocol-tttps-04, 9 slides) and a same-day follow-up email to secdispatch@ietf.org quoting dispatch chair Jim Fenton's live comment that "ISE is a viable option," and John Klensin's related remark. We do not claim presentation at a numbered IETF meeting week; the confirmed event is this interim session.

Companion paper (not yet re-verified in this document). *Proof-of-Time: Byzantine-Resilient Temporal Ordering in Untrusted Networks*, Heime Jorgen, SSRN, abstract_id 6467220. Cited for completeness; treat as unverified until an explicit verification pass is recorded.

Abstract

Systems that log their own AI agent decisions are self-attesting: nothing external prevents a record from being written, or rewritten, after the fact.

TTTPS (TLS TimeToken Protocol Suite) closes this gap by sealing a record's timestamp and content binding at generation time, using mechanisms already deployed in TLS (RFC 5705 Keying Material Exporters) rather than a new handshake extension.

Two integrity layers are defined. The mandatory-to-implement layer (`0x0001`, arity-2) is an unkeyed SHA-256 commitment: recomputation mismatch is an immediate tamper verdict, and this algorithm carries no encumbrance. A second layer, GRG (`0x0100`, arity-3: Golomb-Rice $\rightarrow$ Reed-Solomon(4,6) $\rightarrow$ Golay(23,12,7)), was fully specified with proven coding-theoretic bounds in draft versions `-06` and `-07` and operated experimentally for six months; as of `-08` its registry slot is *reserved* rather than assigned, because it is the subject of a pending patent application by the author, deliberately separated from the royalty-free mandatory-to-implement path under IETF BCP 79.

A second, independent component, K-Lean Vault, is a Lean 4 library of 2,593 theorems, each checked sorry-free and axiom-free, used where a claim benefits from deterministic formal verification rather than a cryptographic receipt alone.

O(1)
per-record verification cost, independent of ledger size

2,593
Lean 4 theorems, sorry-free & axiom-free

70,612
experimental PoT records over six months (55% AI-agent-originated)

The slide deck presented at the interim session itself states the 70,612-record figure is "experimental evidence, not a customer count," with §15.4 of the draft left as a placeholder for confirmed production deployers – we carry that caveat forward rather than upgrade it.

The problem this closes

A system that attests its own timestamps proves nothing about when a record was actually sealed. Backdating a self-attested log is, by construction, undetectable from inside that system.

TTTPS's design question is where the seal happens: before a record enters any system-controlled store (pre-ingestion), or after (post-ingestion – the class SCITT and Certificate Transparency belong to). TTTPS is pre-ingestion by design (draft §2.5, requirement R3); post-ingestion systems were not designed to satisfy that requirement, and the two are complementary rather than competing.

Two integrity layers

2.1 SHA-256 – mandatory-to-implement

The active standards-track algorithm registered in `-08` §3.3.2 is `0x0001`, arity-2, unkeyed SHA-256 over the full record fields. Any recomputation mismatch is an immediate, unconditional tamper verdict; this is the only algorithm in the current mandatory-to-implement table, and it carries no licensing encumbrance.

2.2 GRG – reserved, arity-3

GRG is not an incomplete design; it is a fully specified three-stage pipeline present in `-06` §6 with named coding-theoretic bounds this document cites precisely rather than re-derives.

The bounds

- **Golomb-Rice (G_1)**: reaches the Shannon source-coding limit for PoT timestamp-delta fields (-06 §6.3).
- **Reed-Solomon(4,6) over $\mathrm{GF}(2^8)$ (R)**: an MDS code meeting the Singleton bound – any 4 of 6 shards reconstruct the original.
- **Golay(23,12,7) (G_2)**: a perfect code meeting the Hamming bound exactly (Tietäväinen 1973), correcting up to 3 bit errors per 23-bit block.

Why this order

The stage order $G_1 \to R \to G_2 \to H$ is not arbitrary: the companion paper's Theorem 1 requires G_1 output to be byte-aligned for R's symbol boundaries to be optimal, and Theorem 2 requires R's fixed-size shards for G_2's lossless encoding, with combined failure probability

$$P(\text{fail}) = P(R) \cdot P(G_2) < P(R) + P(G_2).$$

HMAC-SHA256 (H) is checked before the Ed25519 double-seal because it is roughly 15$\times$ cheaper ($\sim$6µs vs. $\sim$100µs, -06 §6.5).

Because every stage operates on a fixed-size block, verifying one record's integrity does not depend on how many other records exist in the ledger: per-record verification cost is $O(1)$ in total ledger size, unlike schemes that require replaying a full hash chain or walking a Merkle path whose depth grows with the ledger.

> -08 §10, verbatim: "The integrity algorithm for which Section 9.3 reserves 0x0100 (GRG) is the subject of a pending patent application by the author … This document is fully implementable without GRG. The mandatory-to-implement integrity algorithm is unkeyed SHA-256 (0x0001), which is unencumbered … This arrangement is deliberate: the specification does not require any implementer to obtain a licence." This is the standard IETF pattern of separating a royalty-free mandatory path from a RAND-licensed optional extension under BCP 79, not evidence of an unfinished design.

K-Lean Vault

K-Lean Vault is a deterministic verification library: 2,593 theorems across 1,491 modules and, per its own indexer, 16 domains, each checked sorry-free and axiom-free by the Lean 4 kernel.

Machine-checked: `K-Lean Vault corpus, per its own indexer, 12 August 2026. lake build` exit 0, no `sorry`, `#print axioms` reports only `propext`, `Classical.choice`, `Quot.sound`.

It answers a different question than TTTPS: TTTPS proves *when* a record was sealed and that it has not been altered since; K-Lean Vault proves that a stated rule (a policy bound, a compliance constraint) holds deterministically, rather than being asserted by a probabilistic model. The two compose – TTTPS seals the record of an inference; K-Lean Vault can certify that the inference obeyed a formally stated rule – but neither substitutes for the other. This paper does not claim K-Lean Vault verifies the correctness of an underlying AI model's output in general, only of formally stated properties checked against it.

Regulatory positioning

Both layers are described here as a *supplementary evidence layer* on top of existing requirements, not as satisfying a requirement that mandates cryptographic timestamping by name.

FDA 21 CFR Part 11 §11.10(e) requires automated audit trails; TTTPS adds a cryptographically sealed timestamp on top of that requirement rather than being required by it. EU AI Act Art. 12/53 requires record-keeping for high-risk AI systems without naming a cryptographic mechanism – the same framing applies. Prior drafting in this project incorrectly cited SEC Rule 17a-4 and DORA as requiring AI-specific cryptographic timestamps – 17a-4's actual requirement is WORM storage, and DORA contains no such AI-timestamp requirement – and those citations are withdrawn here rather than repeated.

What is measured, and what is not

The SHA-256 tamper-detection property is unconditional and needs no qualification. The GRG bounds (Shannon, Singleton, Hamming) are proven in the cited draft sections, not re-derived in this paper. The 2,593-theorem count is machine-counted against K-Lean Vault's own indexer. The 70,612-record, six-month, 55%-AI-agent-originated figures come from `-07` §6 and the interim slide deck itself, both of which describe this as experimental deployment evidence, explicitly not a customer count; §15.4 of the draft is left as a placeholder for confirmed production deployers, and this paper does not fill that placeholder. The companion PoT paper (SSRN 6467220) is cited for context above but its primary source has not been re-verified in the session that produced this document; readers should treat that citation as unverified until an explicit verification pass is recorded.