Network Working Group
Internet-Draft
Intended status: Informational
Expires: 28 September 2026

L.J. Reilly
Independent Submission

28 March 2026

Reilly Sentinel Protocol (RSP): Blockchain-Anchored Integrity for AI Datasets, Training, Fine-Tuning, and Inference Provenance

draft-reilly-sentinel-protocol-01

Abstract

The Reilly Sentinel Protocol (RSP) specifies an interoperable, multi-layer method for establishing integrity, provenance, and auditability across the artificial intelligence (AI) lifecycle. RSP defines a Sentinel Evidence Package (SEP) that binds payload digests, provenance metadata, signatures, blockchain timestamp proofs, and resolvable identifiers (DOIs). This enables tamper-evident, independently verifiable receipts for datasets, data transformations, training jobs, checkpoints, fine-tuning runs, evaluations, inference outputs, and agentic AI action logs.

This revision (-01) expands the protocol with a quantum-resistant triple-hash cross-chain architecture (SHA-256, SHA3-512, BLAKE3), formal cross-chain binding proofs, agentic AI provenance extensions, streaming inference provenance, federated learning provenance, expanded threat model covering post-quantum adversaries, integration with the Reilly EternaMark (REM) Protocol triple-layer permanence system, implementation status updates, and a conformance framework.

RSP is transport-agnostic and serializable in JSON and CBOR. It leverages existing IETF and industry building blocks, including COSE/CMS signatures, CBOR (RFC 8949), CDDL (RFC 8610), JSON (RFC 8259), and NTS-secured time (RFC 8915). Anchoring is done via append-only blockchain receipts and identity/lineage is stabilized with a DOI registry. The result is an evidence-grade, quantum-resilient audit trail for regulated and safety-critical AI deployments worldwide.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 September 2026.

Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction

Artificial Intelligence (AI) systems are increasingly deployed in high-stakes contexts including defense, healthcare, finance, critical infrastructure, and autonomous systems. Confidence in AI outcomes depends on the ability to demonstrate where data originated, how models were trained or adapted, when inferences were produced, by which agents actions were taken, and whether any of these artifacts have been altered after the fact.

The emergence of agentic AI systems -- autonomous agents that execute multi-step tasks, invoke tools, and produce outputs without direct human supervision per step -- introduces new provenance requirements beyond static artifact integrity. An agentic AI system may process thousands of actions per session; each action represents a provenance event that may affect downstream artifacts, decisions, and liability.

Furthermore, the anticipated arrival of cryptographically relevant quantum computers (CRQCs) threatens the long-term integrity of provenance records anchored exclusively with classical hash

algorithms. SHA-256, the dominant anchoring hash, is vulnerable to Grover's algorithm which reduces its effective security from 256 bits to approximately 128 bits. For provenance records intended to remain verifiable for decades -- as required by regulated industries and legal proceedings -- this represents a material risk.

RSP addresses these needs by defining a minimal yet extensible evidence container -- the Sentinel Evidence Package (SEP) -- a quantum-resistant triple-hash cross-chain architecture, and a verification process that any independent party can execute without trusting the producer's infrastructure. RSP is content- and model-agnostic: it does not dictate model architecture or task; it standardizes how integrity, time, identity, lineage, and quantum resilience are recorded and verified.

RSP in this revision combines three complementary layers:

* Quantum-resistant triple-hash anchoring: SHA-256, SHA3-512 (post-quantum resilient), and BLAKE3 computed simultaneously and cross-chain bound via a Composite Proof Object (CPO).

* Multi-chain blockchain anchoring: append-only proof of existence across heterogeneous consensus systems, reducing single-chain failure risk.

* DOI registration: globally resolvable, citable identifiers with metadata, lineage, and retention semantics backed by institutional archival infrastructure.

The REM Protocol (draft-reilly-rem-protocol-01) provides the underlying triple-layer permanence infrastructure on which RSP anchoring operations are built.

1.1. Changes from -00

This document supersedes draft-reilly-sentinel-protocol-00. Principal changes are summarized in Appendix H. Key additions include:

* Section 5: Quantum-resistant triple-hash architecture.
* Section 6: Cross-chain binding and Composite Proof Objects.
* Section 10: REM Protocol integration.
* Section 13: Agentic AI provenance extensions.
* Section 14: Streaming inference provenance.
* Section 15: Federated learning provenance.
* Section 16: Conformance framework.
* Expanded threat model in Section 20.
* Updated implementation status in Section 22.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

SEP: Sentinel Evidence Package; the container defined by this document.

CPO: Composite Proof Object; a structure binding SHA-256, SHA3-512, and BLAKE3 chain hashes with a cross-chain binding hash.

Artifact: Any AI lifecycle asset -- dataset, data transform,

training configuration, training log, model checkpoint, fine-tuning diff, evaluation artifact, inference record, or agentic action log.

Digest: A cryptographic hash of a payload or Merkle root over chunked payloads.

OTS: OpenTimestamps-style proof object or functionally equivalent blockchain timestamp receipt.

DOI: Digital Object Identifier; a resolvable identifier providing metadata and lineage stability (DataCite, Crossref, or internal Handle).

COSE: CBOR Object Signing and Encryption; RSP uses COSE signatures [RFC9052] [RFC9053].

CMS: Cryptographic Message Syntax [RFC5652].

CRQC: Cryptographically Relevant Quantum Computer; a quantum computer capable of breaking classical cryptographic algorithms at practical scale.

REM: Reilly EternaMark Protocol; the triple-layer permanence infrastructure used by RSP for anchoring operations.

AAL: Agentic Action Log; a structured record of autonomous agent actions, tool invocations, and decision events.

SIP: Streaming Inference Proof; a provenance record for streaming or real-time AI inference outputs.

FLP: Federated Learning Proof; a provenance record for federated model updates preserving participant privacy.

3. Architecture and Roles

Producer: Creates artifacts, computes triple-hash digests, collects device and environment attestations, signs SEPs, and requests anchoring and DOI registration via the REM Protocol infrastructure.

Registrar: Provides two sub-services:
(a) Anchoring Orchestrator that submits digests to public chains across multiple consensus systems and returns CPOs.
(b) DOI Registrar that mints DOIs and stores metadata, lineage, and fixity records.

Verifier: Recomputes triple-hash digests, validates signatures, checks OTS receipts against independent nodes, resolves DOI metadata, and verifies cross-chain binding without trusting Producer or Registrar beyond their published signatures.

Storage: WORM (Write-Once-Read-Many) or equivalent immutable object storage for SEPs, CPOs, and payloads under retention policy. IPFS content-addressed storage is RECOMMENDED for decentralized redundancy.

Time: Authenticated time sources (NTS, RFC 8915) and monotonic counters to resist back-dating and rollback attacks.

REM Infrastructure: The deployed REM Protocol verification agent provides REMID assignment, chain integrity verification, and verification page generation for SEPs anchored through RSP.

RSP does not mandate a specific trust relation among these roles;

in many deployments, Producer and Registrar are separate entities. The REM Infrastructure MAY be operated by the Producer.

4. Sentinel Evidence Package (SEP) Data Model

An SEP is a self-describing manifest that binds digest(s), metadata, signatures, proofs, and identifiers. Version 1.1 adds triple-hash fields, CPO references, and agentic extensions.

Core fields:

* version: Protocol version string. MUST be "1.1" for this revision.

* artifact.type: One of "dataset", "datatransform", "training.config", "training.log", "training.checkpoint", "fine-tune.diff", "evaluation", "inference", "inference.stream", "federated.update", "agentic.action.log", or an extension token.

* payloads[]: Optional array with entries { cid, size, mime, chunking }; payload storage MAY be external. 'cid' MUST encode algorithm and value (e.g., "sha256:HEX").

* digests: Object with sha256, sha3_512, and blake3 members (hex-encoded). sha3_512 provides post-quantum resilience. blake3 provides high-performance verification. For chunked payloads, digests.root is the Merkle root.

* cpo_ref: Reference to the Composite Proof Object binding all three hash chains. MUST be present when triple-hash anchoring is used.

* timestamps: wallclock (RFC 3339) and monotonic counter. The 'source' SHOULD indicate NTS, Roughtime, or equivalent.

* authors[]: Optional list of creators/operators; MAY include device IDs, certificate references, and ORCID identifiers.

* attestation: Optional device/firmware/build attestation claims (see RATS, RFC 9334).

* rem: Anchoring and identity sub-object:
  - rem.remid: REMID assigned by REM infrastructure.
  - rem.ots_proof_ref: URN or locator to OTS proof file.
  - rem.doi: DOI string.
  - rem.ipfs_cid: IPFS content identifier for redundancy.
  - rem.lineage: parent[], siblings[], supersedes[].
  - rem.quantum_resilient: Boolean. MUST be true when CPO includes SHA3-512 and BLAKE3 chains.

* agentic: Optional sub-object for agentic AI provenance. See Section 13.

* class: Optional classification/handling indicators (tokenized).

* policies: Retention and access policy references.

* signatures[]: One or more COSE_Sign1 or CMS signatures covering the manifest canonical form. Certificate chains MUST be included.

The manifest MUST be canonicalized prior to signing; canonical JSON (JCS, RFC 8785) or deterministic CBOR (RFC 8949 Section 4.2) MUST be used.

5. Quantum-Resistant Triple-Hash Architecture

5.1. Motivation

SHA-256 alone is insufficient for provenance records requiring long-term verifiability in a post-quantum threat environment. Grover's algorithm reduces SHA-256's effective collision resistance from 256 bits to approximately 128 bits under a CRQC adversary. While 128-bit security is currently considered adequate, provenance records anchored today may need to remain verifiable for 20 to 50 years, a time horizon in which CRQCs may become practical.

RSP addresses this by mandating three simultaneous hash computations:

* SHA-256: Classical standard; widely deployed and auditable.

* SHA3-512: Post-quantum resilient. SHA-3 is based on the Keccak sponge construction, which is structurally different from SHA-2. A quantum adversary cannot apply Grover's algorithm to SHA-3 with the same efficiency as SHA-2. SHA3-512 provides 256-bit post-quantum security.

* BLAKE3: High-performance, parallelizable hash function. Provides independent algorithmic diversity from both SHA-2 and SHA-3 families, ensuring that a break in either family does not compromise the overall proof chain.

5.2. Triple-Hash Computation

For each artifact, the Producer MUST compute:

```
sha256_hash  = SHA-256(artifact_bytes)
sha3_512_hash = SHA3-512(artifact_bytes)
blake3_hash  = BLAKE3(artifact_bytes)
```

All three values MUST be recorded in the SEP digests field.

5.3. Security Properties

An adversary seeking to forge an artifact's provenance record must simultaneously break SHA-256, SHA3-512, AND BLAKE3. No known classical or quantum algorithm achieves this. The three algorithms are from distinct design families with no known structural relationships, ensuring that a cryptanalytic break of one does not assist attacks on the others.

This architecture provides:
* Classical security: 256-bit effective strength (SHA-256).
* Post-quantum security: 256-bit effective strength (SHA3-512).
* Algorithmic diversity: Independent verification path (BLAKE3).

6. Cross-Chain Binding and Composite Proof Objects

6.1. Chain Architecture

RSP defines three independent hash chains maintained across SEP records:

* SHA-256 chain: Each record's SHA-256 chain hash is computed over the previous SHA-256 chain hash, the current artifact SHA-256 digest, record ID, REMID, and creation timestamp.

* SHA3-512 chain: Analogous construction using SHA3-512. Post-quantum resilient.

* BLAKE3 chain: Analogous construction using BLAKE3.

Each chain is independent. Tampering any single record invalidates its chain hash, which propagates forward through all subsequent records in that chain.

6.2. Cross-Chain Binding Hash

At each record, a cross-chain binding hash is computed as:

```
cross_chain_hash = SHA3-512(
    sha256_chain_hash ||
    sha3_512_chain_hash ||
    blake3_chain_hash
)
```

This binds all three chain states simultaneously at the current block. An adversary wishing to forge a record must simultaneously manipulate all three chains such that the cross-chain binding hash remains valid. This requires simultaneously breaking SHA-256, SHA3-512, and BLAKE3 -- which is computationally infeasible under both classical and quantum threat models.

6.3. Composite Proof Object (CPO)

A CPO is a JSON or CBOR structure that encapsulates the triple-hash chain state at a given record. See Appendix F for the CPO format.

CPOs MUST be signed by the Producer and MAY be countersigned by the Registrar. CPOs SHOULD be anchored to the blockchain alongside the SEP digest to provide timestamped evidence of chain state.

7. Serialization and Media Types

RSP SEPs MAY be emitted as JSON [RFC8259] or CBOR [RFC8949]. This document defines four media types for IANA registration:

* application/rsp-ep+json (unchanged from -00)
* application/rsp-ep+cbor (unchanged from -00)
* application/rsp-cpo+json (new: Composite Proof Object, JSON)
* application/rsp-cpo+cbor (new: Composite Proof Object, CBOR)

The "+" structured suffix follows RFC 6839. See Section 21 for IANA templates.

8. Anchoring and Proofs

Producers submit digests (payload or roll-up Merkle roots) to the Anchoring Orchestrator. The orchestrator MUST anchor to at least one public append-only chain and SHOULD support multiple chains for diversity and resilience. Proofs MUST be exportable and independently checkable without trusting the orchestrator.

For quantum-resilient deployments, the orchestrator MUST anchor the cross-chain binding hash (Section 6.2) in addition to individual algorithm hashes. This provides a single anchored value that commits to all three algorithm states simultaneously.

OpenTimestamps receipts are RECOMMENDED for Bitcoin anchoring. Anchoring to Ethereum via a smart contract event, or to another public chain with independently verifiable inclusion proofs, is also RECOMMENDED for multi-chain diversity.

Roll-up anchoring: Operators SHOULD periodically compute a Merkle tree over all new artifact digests and anchor the root to reduce per-artifact anchoring cost and to provide compact audit coverage. Roll-up trees SHOULD include all three hash values per leaf.

9. DOI Registration and Metadata

After anchoring, the Registrar mints a DOI and stores a metadata record containing at least: title or short label, creators, creation time, fixity (triple-hash digests), lineage links, REMID, IPFS CID, and policy URIs.

DOIs for classified material MAY be internal Handles resolvable on private networks; unclassified artifacts SHOULD use public DOIs (e.g., DataCite/Zenodo).

A DOI SHOULD be minted for each material evolution that changes fixity (e.g., new dataset version, fine-tune result, evaluation report). Prior DOIs MUST NOT be deleted; instead, metadata MAY indicate "superseded-by".

The DOI metadata record SHOULD include:

* rem_id: The REMID assigned by the REM infrastructure.
* sha256: Artifact SHA-256 digest.
* sha3_512: Artifact SHA3-512 digest.
* blake3: Artifact BLAKE3 digest.
* cross_chain: Cross-chain binding hash.
* quantum_resilient: true

10. REM Protocol Triple-Layer Permanence Integration

10.1. Overview

The Reilly EternaMark Protocol (REM) [REM01] provides a triple-layer permanence infrastructure used as the anchoring substrate for RSP. When RSP operates over REM, each SEP receives:

* A REMID: A permanent identifier in the form REMID:YYYY.MMDD/xxxxxxxx.
* A Zenodo DOI: Permanent academic archive backed by CERN.
* An IPFS CID: Content-addressed decentralized storage.
* A Bitcoin OTS proof: Blockchain timestamp via OpenTimestamps.
* A verification page: Human and machine-readable proof page resolvable at the REM verification agent endpoint.

10.2. Integration Requirements

When RSP operates over REM, the following requirements apply:

* The SEP rem.remid field MUST be populated with the REMID assigned by the REM infrastructure.
* The CPO MUST be submitted to the REM infrastructure for REMID assignment and chain integrity verification.
* The SEP rem.quantum_resilient field MUST be set to true.
* The verification page URL MUST be included in SEP metadata as rem.verification_url.

10.3. Chain Integrity

The REM infrastructure maintains chain integrity across all submitted artifacts. Agent 11 performs autonomous self-healing repair cycles that detect and flag chain integrity violations. Producers relying on REM for anchoring SHOULD monitor chain integrity reports and respond to any flagged records.

11. Protocol Operations

The following abstract operations are defined; concrete APIs are out of scope, but typical deployments use REST or gRPC.

11.1 SEAL

Inputs: manifest draft (unsigned), triple-hash digests, optional attestation, optional agentic extension.
Action: Canonicalize (JCS or deterministic CBOR), sign (COSE/CMS), compute CPO, return signed SEP.

11.2 ANCHOR

Inputs: cross-chain binding hash (from CPO).
Action: Anchor to one or more chains; return proof reference(s) and OTS receipts.

11.3 REGISTER

Inputs: metadata including triple-hash fixity, lineage, IPFS CID.
Action: Mint DOI; bind DOI <-> SEP <-> CPO <-> proofs; assign REMID; return DOI and REMID.

11.4 VERIFY

Inputs: DOI, REMID, or SEP URI (plus payload, if accessible).
Action: Recompute triple-hash digests; validate CPO; validate signatures; check time discipline; verify proofs; emit Verification Report.

11.5 REVOKE

Inputs: SEP identifier and revocation reason.
Action: Append a signed revocation fact to the chain; the original SEP MUST NOT be deleted. All subsequent verifications MUST surface the revocation status.

12. Verification Algorithm

Given a DOI, REMID, or SEP:

1. Resolve DOI or REMID to retrieve minimal metadata including triple-hash fixity, lineage, IPFS CID, and pointers to SEP, CPO, and payload (subject to policy).

2. Fetch SEP and CPO. Parse and canonicalize per serialization. Verify that all three digests (sha256, sha3_512, blake3) match the payload (if accessible).

3. Verify CPO cross-chain binding hash by recomputing: SHA3-512(sha256_chain || sha3_512_chain || blake3_chain).

4. Validate signatures (COSE/CMS). Check certificate validity and revocation (OCSP/CRL). Attestation claims MAY be

evaluated via a RATS verifier (RFC 9334).

5. Validate time: ensure wallclock is plausible (NTS/secure time) and monotonic counter continuity holds relative to adjacent SEPs.

6. Validate anchoring proofs independently using public chain data. For OTS proofs, verify against Bitcoin block headers via an independent node.

7. Check IPFS CID by resolving the content-addressed payload.

8. Check revocation status against the chain record.

9. Emit a Verification Report with fields: hash_ok (all three algorithms), cpo_ok, sig_ok, attestation_ok, ots_ok, time_ok, doi_ok, remid_ok, ipfs_ok, revocation_status, and overall verdict. The report SHOULD itself be signed.

13. Agentic AI Provenance Extensions

13.1. Motivation

Agentic AI systems execute multi-step workflows autonomously. Each action -- tool invocation, API call, file write, model inference, decision branch -- represents a provenance event. Traditional SEP structures capture static artifact integrity; agentic provenance requires a structured log of action sequences that preserves causality, timing, and agent identity.

13.2. Agentic Action Log (AAL)

An AAL is an artifact.type "agentic.action.log" SEP extension. The SEP agentic sub-object MUST contain:

* session_id: Unique identifier for the agent session.
* agent_id: Identifier for the agent or agent version.
* model_id: Identifier for the underlying model, with DOI reference to the model checkpoint SEP.
* action_count: Total number of logged actions.
* actions[]: Ordered array of action records.
* merkle_root: Merkle root over all action record digests, providing compact integrity over the log.

Each action record in actions[] MUST contain:

* seq: Sequence number (monotonically increasing).
* timestamp: RFC 3339 wallclock at action execution.
* action_type: One of "tool_call", "inference", "read", "write", "decision", "delegation", or an extension token.
* input_digest: SHA3-512 digest of action input.
* output_digest: SHA3-512 digest of action output.
* tool_id: Identifier of invoked tool (if applicable).
* parent_seq: Sequence number of causal parent action.
* attestation_ref: Optional reference to device attestation.

13.3. AAL Anchoring

The Merkle root over all action records MUST be anchored via the standard RSP ANCHOR operation. For long-running agent sessions, intermediate Merkle roots MAY be anchored at configurable intervals to provide bounded recovery windows.

13.4. Privacy in Agentic Logs

Action inputs and outputs are identified by digest only; raw content is not included in the AAL. Producers MAY encrypt payloads and include encrypted payload references in action records. This enables selective disclosure: a party may prove that a specific action occurred without revealing the content of inputs or outputs.

14. Streaming Inference Provenance

14.1. Motivation

Real-time AI systems produce inference outputs as streams. Standard SEP structures assume bounded artifacts; streaming inference requires provenance over unbounded token sequences or data streams.

14.2. Streaming Inference Proof (SIP)

A SIP is an artifact.type "inference.stream" SEP extension. The stream is chunked into fixed-size windows (default: 1000 tokens or 30 seconds of data, whichever is smaller).

For each window:
* Compute SHA3-512 digest of the window content.
* Append to a rolling Merkle tree.
* Record the window's position in the stream and the Merkle path at that position.

At stream termination:
* Anchor the final Merkle root via ANCHOR.
* Issue a SEP with artifact.type "inference.stream" referencing all window digests.

Verifiers MAY verify any individual window by checking its Merkle path against the anchored root, without retrieving the entire stream.

14.3. Real-Time Monitoring

For safety-critical inference (medical diagnosis, autonomous vehicle control), producers SHOULD maintain a live anchor feed submitting intermediate Merkle roots at configurable intervals (e.g., every 10 seconds). This bounds the maximum provenance gap in the event of unexpected stream termination.

15. Federated Learning Provenance

15.1. Motivation

Federated learning (FL) trains models across distributed clients without centralizing raw data. Each client computes a local model update which is aggregated at a central server. Provenance for FL must record the integrity of updates without revealing client data or update content, and must capture the aggregation operation.

15.2. Federated Learning Proof (FLP)

An FLP is an artifact.type "federated.update" SEP extension. The SEP agentic sub-object is unused; instead, the SEP MUST contain a federated sub-object with:

* round_id: Federated learning round identifier.
* aggregation_id: Identifier for the aggregation operation.

* participant_count: Number of clients contributing updates.
* update_root: Merkle root over all client update digests, computed by the aggregation server.
* global_model_doi: DOI reference to the resulting global model checkpoint SEP.
* privacy_mechanism: One of "dp_gaussian", "dp_laplace", "secure_aggregation", "none", or extension.

Individual client update digests are included in the Merkle tree but individual client identities are NOT recorded in the FLP. The update_root provides aggregate integrity without revealing per-client contributions.

15.3. FLP Anchoring

The update_root MUST be anchored via the standard RSP ANCHOR operation. The resulting SEP MUST be registered with a DOI. The global model checkpoint SEP MUST include a lineage reference to the FLP SEP, establishing a verifiable provenance chain from training data contributions through aggregation to the deployed model.

16. Conformance Framework

16.1. Conformance Levels

RSP defines three conformance levels for implementations:

RSP-Basic:
* SEP version 1.1 with SHA-256 digests.
* Single-chain OTS anchoring.
* DOI registration.
* Standard SEAL, ANCHOR, REGISTER, VERIFY operations.

RSP-Quantum:
* All RSP-Basic requirements.
* Triple-hash digests (SHA-256, SHA3-512, BLAKE3).
* CPO generation and verification.
* Cross-chain binding hash.
* rem.quantum_resilient MUST be true.

RSP-Full:
* All RSP-Quantum requirements.
* REM Protocol integration (REMID, IPFS CID).
* Agentic Action Log support (Section 13).
* Streaming inference provenance (Section 14) or federated learning provenance (Section 15).
* Multi-chain anchoring (minimum two independent chains).

16.2. Conformance Testing

Implementations claiming an RSP conformance level MUST pass the verification algorithm in Section 12 for all artifact types applicable to that level. Test vectors for triple-hash computation and CPO verification are provided in Appendix F.

17. Error Handling

RSP defines abstract error conditions with suggested codes:

ERR_PARSE	malformed SEP, CPO, or encoding
ERR_DIGEST_MISMATCH	recomputed digest != manifest (any alg)
ERR_CPO_INVALID	cross-chain binding hash mismatch
ERR_SIG_INVALID	signature or chain invalid

ERR_TIME_INVALID	unauthenticated or non-monotonic time
ERR_PROOF_INVALID	proof not verifiable on stated chain
ERR_DOI_RESOLVE	DOI metadata unavailable/inconsistent
ERR_REMID_RESOLVE	REMID not found or chain violation
ERR_IPFS_RESOLVE	IPFS CID not resolvable
ERR_POLICY_DENIED	access denied by policy
ERR_REVOKED	SEP has been revoked

Implementations SHOULD map these to transport-specific error responses. ERR_CPO_INVALID and ERR_REVOKED MUST surface as prominent verification failures; they MUST NOT be silently ignored.

18. Manageability and Telemetry

Deployments SHOULD monitor:

* Anchoring SLA (time to first chain inclusion per algorithm).
* Triple-hash coverage (% artifacts with all three digests).
* CPO generation success rate.
* Verification pass rate and dominant failure reasons.
* Chain integrity health (monitored by REM Agent 11 or equivalent).
* Key health (expiration, revocation events).
* Time-source health (NTS/stratum, Roughtime reachability).
* IPFS pin health (replication factor and retrieval latency).

For agentic deployments, additional monitoring SHOULD include:

* AAL action count per session.
* AAL Merkle root anchor latency.
* Streaming inference window gap rate.

19. Privacy Considerations

SEPs record integrity metadata; payloads MAY be withheld. Where privacy is required, field-level encryption SHOULD be used for sensitive manifest attributes while keeping digests and proofs public. DOI records for sensitive artifacts SHOULD minimize personally identifiable information. Policy URIs SHOULD reflect access constraints.

For agentic logs, action inputs and outputs are represented by digest only (Section 13.4). Raw content MUST NOT be embedded in AAL records unless explicitly authorized by applicable policy.

For federated learning, per-client identities MUST NOT appear in FLP records (Section 15.2).

Blockchain anchoring publishes digest values to a public ledger. Producers MUST ensure that digest values alone do not reveal sensitive information. Pre-image resistance of SHA3-512 provides strong protection against reverse-engineering payload content from anchored digests.

20. Security Considerations

20.1. Key Compromise

Private keys used for COSE/CMS MUST be protected in HSMs or equivalent. Short-lived device certificates and revocation (OCSP/CRL) are RECOMMENDED. Compromised keys MUST NOT trigger data deletion; instead, append signed revocation facts and supersede with new signatures. The immutability of prior

records is a fundamental security property of RSP.

20.2. Time Attacks

Producers MUST use authenticated time (NTS, RFC 8915) and monotonic counters. Multiple independent time sources are RECOMMENDED. Large backward jumps MUST be flagged as ERR_TIME_INVALID. Roughtime [ROUGHTIME] provides an additional authenticated time source with accountability for incorrect time.

20.3. Quantum Threat Model

RSP-Quantum and RSP-Full implementations are designed to resist the following quantum adversary capabilities:

* Grover's algorithm against SHA-256: Mitigated by SHA3-512 (post-quantum resilient, 256-bit effective security under Grover) and BLAKE3 (independent algorithmic family).

* Structural attacks on SHA-2 family: Mitigated by SHA3-512 (Keccak sponge construction, structurally independent of SHA-2) and BLAKE3.

* Cross-chain binding attack: An adversary seeking to forge the CPO must simultaneously break SHA-256, SHA3-512, and BLAKE3 AND produce a valid cross-chain binding hash. No known quantum algorithm achieves this.

RSP-Basic implementations do not claim quantum resilience. Operators of long-term provenance systems (20+ year horizon) SHOULD deploy RSP-Quantum or RSP-Full.

20.4. Proof Stability

Anchoring to multiple independent chains (heterogeneous consensus) reduces correlation risk and single-chain failure. Re-anchoring of historical Merkle roots over time is RECOMMENDED for longevity as new cryptographic standards emerge.

20.5. Replay and Substitution

Bind signatures to digests and policy context. DOIs and REMIDs provide stable identity to detect artifact swaps. Monotonic counters in AALs prevent action replay attacks in agentic deployments.

20.6. Supply Chain

Model and dataset supply chain attestations (e.g., SBOM-like metadata, RATS evidence) are RECOMMENDED but out of scope to normatively specify here. The FLP structure (Section 15) provides aggregate supply chain provenance for federated training.

Security considerations follow RFC 3552 guidance.

21. IANA Considerations

IANA is requested to register the following media types:

21.1 application/rsp-ep+json (unchanged from -00)
21.2 application/rsp-ep+cbor (unchanged from -00)

21.3 application/rsp-cpo+json

Type name: application

```
Subtype name:        rsp-cpo+json
Required parameters: none
Optional parameters: charset (per RFC 6838)
Encoding considerations:  binary; JSON (RFC 8259)
Security considerations:  see Section 20
Published specification:  this document
Applications:  AI provenance systems using RSP triple-hash
File extension(s):  .rsp-cpo.json
Person & email address:
  Lawrence John Reilly Jr. <lawrencejohnreilly@gmail.com>
Intended usage:  COMMON
```

21.4 application/rsp-cpo+cbor

```
Type name:           application
Subtype name:        rsp-cpo+cbor
Required parameters: none
Encoding considerations:  binary; CBOR (RFC 8949)
Security considerations:  see Section 20
Published specification:  this document
Applications:  AI provenance systems using RSP triple-hash
File extension(s):  .rsp-cpo.cbor
Person & email address:
  Lawrence John Reilly Jr. <lawrencejohnreilly@gmail.com>
Intended usage:  COMMON
```

22. Implementation Status

Note to RFC Editor: Please remove this section before publication.

This section documents the implementation status of RSP as of March 2026, per RFC 7942.

22.1. REM Protocol Verification Agent

```
Organization:  Lawrence John Reilly Jr., Independent
Description:   Live deployment of the REM Protocol infrastructure
               providing REMID assignment, triple-hash chain
               integrity, IPFS pinning via Pinata, Bitcoin OTS
               anchoring, and verification page generation.
URL:           https://rem-protocol-agent-production.up.railway.app
Coverage:      RSP-Full anchoring substrate.
License:       Proprietary.
Contact:       lawrencejohnreilly@gmail.com
```

22.2. Reference SEP Records

Multiple SEPs have been issued and anchored through the REM infrastructure, including:

* REMID: 2026.0328/e39dd6db
  DOI: 10.5281/zenodo.19299622
  Title: Spreadsheet AI Invention Declaration
  Anchored: 2026-03-28T22:08:38 UTC

* REMID: zenodo/19192612
  DOI: 10.5281/zenodo.19192612
  Title: Token config (RLT genesis)
  Anchored: 2026-03-23T19:07:09 UTC

22.3. Planned Implementations

Reference implementations for SEP creation, CPO generation, triple-hash computation, and verification are planned for public release under open-source license. The Python implementation of

the REM Multi-Algorithm Stack (REM-MAS) with 27/27 tests passing provides the cryptographic substrate for triple-hash operations.

23. References

23.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC8259] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 8259, December 2017.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL)", RFC 8610, June 2019.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020.

[RFC8915] Franke, D., "Network Time Security for the Network Time Protocol", RFC 8915, September 2020.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, Dec 2020.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", RFC 9052, Aug 2022.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Algorithms", RFC 9053, August 2022.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

[RFC9334] Birkholz, H., et al., "Remote Attestation Procedures Architecture", RFC 9334, January 2023.

23.2. Informative References

[RSP00] Reilly, L. J., "Reilly Sentinel Protocol (RSP) -00", draft-reilly-sentinel-protocol-00, September 2025.

[REM01] Reilly, L. J., "Reilly EternaMark (REM) Protocol", draft-reilly-rem-protocol-01, 2026. Zenodo: doi:10.5281/zenodo.17129012.

[RBIP01] Reilly, L. J., "Reilly Banking Integrity Protocol", draft-reilly-banking-integrity-01, 2026.

[RRP01] Reilly, L. J., "Reilly Resilience Protocol", draft-reilly-resilience-protocol-01, 2026.

[CTS00] Reilly, L. J., "Cognitive Trust Stack", draft-reilly-cts-00, 2026. Zenodo: doi:10.5281/zenodo.19098735.

[SCITT] IETF, "Supply Chain Integrity, Transparency, and Trust (SCITT) Working Group Charter", IETF Datatracker.

[C2PA] Coalition for Content Provenance and Authenticity, "C2PA Specification", 2023.

[OTS] OpenTimestamps community, "OpenTimestamps: Scalable Timestamping", online documentation.

[DATACITE] DataCite Metadata Schema 4.x, DataCite.

[BLAKE3] O'Connor, J., et al., "BLAKE3: One Function, Fast Everywhere", 2020.

[ROUGHTIME] Sleep, W., et al., "Roughtime", draft-ietf-ntp-roughtime, work in progress.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 7942, July 2016.

[RFC6839] Hansen, T. and A. Melnikov, "Additional Media Type Structured Syntax Suffixes", RFC 6839, January 2013.

Appendix A. CDDL for CBOR SEP (v1.1)

```
sep = {
  version: tstr,               ; "1.1"
  artifact: {
    type: tstr
  },
  ? payloads: [* payload],
  digests: {
    sha256: tstr,              ; lowercase hex
    sha3_512: tstr,            ; lowercase hex (post-quantum)
    blake3: tstr,              ; lowercase hex
    ? root: tstr               ; Merkle root for chunked
  },
  ? cpo_ref: tstr,             ; Reference to CPO
  timestamps: {
    wallclock: tstr,           ; RFC3339
    ? monotonic: uint,
    ? source: tstr             ; "nts", "roughtime", etc.
  },
  ? authors: [* { name: tstr, ? id: tstr, ? orcid: tstr }],
  ? attestation: any,
  rem: {
    ? remid: tstr,             ; REMID:YYYY.MMDD/xxxxxxxx
    ots_proof_ref: tstr,
    doi: tstr,
    ? ipfs_cid: tstr,
    ? verification_url: tstr,
    lineage: {
      ? parent: [* tstr],
      ? siblings: [* tstr],
      ? supersedes: [* tstr]
    },
    ? quantum_resilient: bool
  },
  ? agentic: {
    session_id: tstr,
    agent_id: tstr,
    model_id: tstr,
    action_count: uint,
    merkle_root: tstr,
    ? actions: [* action_record]
  },
  ? federated: {
```

```
      round_id: tstr,
      aggregation_id: tstr,
      participant_count: uint,
      update_root: tstr,
      global_model_doi: tstr,
      privacy_mechanism: tstr
    },
    ? class: any,
    ? policies: any,
    signatures: [* any]
  }

  payload = {
    cid: tstr,
    size: uint,
    ? mime: tstr,
    ? chunking: tstr
  }

  action_record = {
    seq: uint,
    timestamp: tstr,
    action_type: tstr,
    input_digest: tstr,          ; SHA3-512
    output_digest: tstr,         ; SHA3-512
    ? tool_id: tstr,
    ? parent_seq: uint,
    ? attestation_ref: tstr
  }
```

Appendix B. Example JSON SEP with Triple-Hash

```
  {
    "version": "1.1",
    "artifact": {"type": "training.checkpoint"},
    "digests": {
      "sha256":   "a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4...",
      "sha3_512": "b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5...",
      "blake3":   "c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6..."
    },
    "cpo_ref": "urn:rem:cpo:YYYY.MMDD/xxxxxxxx",
    "timestamps": {
      "wallclock": "2026-01-01T00:00:00Z",
      "source": "nts"
    },
    "rem": {
      "remid": "REMID:YYYY.MMDD/xxxxxxxx",
      "ots_proof_ref": "urn:ots:btc:txid:example",
      "doi": "10.5281/zenodo.example",
      "ipfs_cid": "bafkreiexamplecidvalue...",
      "verification_url":
        "https://rem-protocol-agent-production.up.railway.app/
         verify/REMID%3AYYYY.MMDD%2Fxxxxxxxx",
      "lineage": {},
      "quantum_resilient": true
    },
    "signatures": [{"type": "COSE_Sign1", "value": "..."}]
  }
```

Appendix C. Verification Report (JSON)

```
  {
    "doi": "10.5281/zenodo.example",
    "remid": "REMID:YYYY.MMDD/xxxxxxxx",
```

```
  "hash_ok": {
    "sha256": true,
    "sha3_512": true,
    "blake3": true
  },
  "cpo_ok": true,
  "sig_ok": true,
  "attestation_ok": true,
  "ots_ok": true,
  "time_ok": true,
  "doi_ok": true,
  "remid_ok": true,
  "ipfs_ok": true,
  "revocation_status": "not_revoked",
  "quantum_resilient": true,
  "overall": "VERIFIED",
  "verified_at": "2026-03-28T22:30:00Z"
}
```

Appendix D. Example OpenTimestamps Receipt (Non-normative)

An OTS receipt includes commitment operations and attestations that enable independent reconstruction of inclusion in a public chain. See [OTS]. For RSP, OTS receipts SHOULD cover the cross-chain binding hash (Section 6.2) rather than individual algorithm hashes, providing a single anchored commitment to all three chain states.

Appendix E. Example DOI Metadata Mapping (DataCite)

```
title:       "Example AI Training Checkpoint"
creators:    [{"name": "Lawrence John Reilly Jr."}]
created:     "2026-01-01T00:00:00Z"
fixity:
  sha256:    "a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4..."
  sha3_512:  "b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5..."
  blake3:    "c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6..."
  cross_chain: "d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1..."
rem_id:      "REMID:YYYY.MMDD/xxxxxxxx"
ipfs_cid:    "bafkreiexamplecidvalue..."
quantum_resilient: true
```

Appendix F. Composite Proof Object (CPO) Format

A CPO binds three chain hashes and their cross-chain binding:

```
{
  "cpo_version": "1.0",
  "record_id": "<uuid>",
  "remid": "REMID:YYYY.MMDD/xxxxxxxx",
  "created_at": "<RFC3339>",
  "chain": {
    "block_index": <uint>,
    "sha256_chain_hash": "<hex>",
    "sha3_512_chain_hash": "<hex>",
    "blake3_chain_hash": "<hex>",
    "cross_chain_hash": "<hex>",
    "prev_sha256_chain": "<hex>",
    "prev_sha3_512_chain": "<hex>",
    "prev_blake3_chain": "<hex>"
  },
  "quantum_resilient": true,
  "signatures": [{"type": "COSE_Sign1", "value": "..."}]
```

```
}
```

The cross_chain_hash MUST equal:
SHA3-512(sha256_chain_hash || sha3_512_chain_hash ||
blake3_chain_hash)

Verifiers MUST recompute this value and reject CPOs where the computed value does not match the recorded cross_chain_hash.

Appendix G. Agentic Action Log Example (Non-normative)

```
{
  "version": "1.1",
  "artifact": {"type": "agentic.action.log"},
  "agentic": {
    "session_id": "sess-20260328-001",
    "agent_id": "spreadsheet-ai-v1.0",
    "model_id": "claude-haiku-4-5",
    "action_count": 13,
    "merkle_root": "sha3_512:abc123...",
    "actions": [
      {
        "seq": 1,
        "timestamp": "2026-03-28T21:06:00Z",
        "action_type": "inference",
        "input_digest": "sha3_512:input_hash_1...",
        "output_digest": "sha3_512:output_hash_1...",
        "tool_id": "spreadsheet-ai:rewrite",
        "parent_seq": null
      }
    ]
  }
}
```

Appendix H. Change Log

Changes from draft-reilly-sentinel-protocol-00 to -01:

H.1 Abstract

Extended to cover quantum-resistant triple-hash architecture, agentic AI provenance, streaming inference, and federated learning provenance.

H.2 Section 1 (Introduction)

Added motivation for quantum resilience and agentic AI provenance. Added Section 1.1 summarizing changes.

H.3 Section 2 (Terminology)

Added CPO, CRQC, REM, AAL, SIP, FLP definitions.

H.4 Section 4 (SEP Data Model)

Added sha3_512, blake3, cpo_ref, rem.remid, rem.ipfs_cid, rem.verification_url, rem.quantum_resilient, agentic, and federated sub-objects. Version bumped to "1.1".

H.5 Section 5 (new)

Quantum-resistant triple-hash architecture.

H.6 Section 6 (new)

Cross-chain binding and Composite Proof Objects.

H.7 Section 8 (Anchoring)

Added requirement to anchor cross-chain binding hash.

H.8 Section 9 (DOI)
Added triple-hash fields and REMID to DOI metadata.

H.9 Section 10 (new)
REM Protocol triple-layer permanence integration.

H.10 Section 11 (Operations)
Added REVOKE operation. Updated SEAL and ANCHOR for CPO.

H.11 Section 12 (Verification)
Added CPO verification, IPFS check, and revocation check.
Expanded Verification Report fields.

H.12 Section 13 (new)
Agentic AI provenance extensions.

H.13 Section 14 (new)
Streaming inference provenance.

H.14 Section 15 (new)
Federated learning provenance.

H.15 Section 16 (new)
Conformance framework with RSP-Basic, RSP-Quantum, RSP-Full.

H.16 Section 17 (Error Handling)
Added ERR_CPO_INVALID, ERR_REMID_RESOLVE, ERR_IPFS_RESOLVE,
ERR_REVOKED.

H.17 Section 18 (Telemetry)
Added agentic monitoring requirements.

H.18 Section 20 (Security)
Expanded threat model with quantum attack analysis.

H.19 Section 21 (IANA)
Added rsp-cpo+json and rsp-cpo+cbor media types.

H.20 Section 22 (Implementation Status)
Updated with live REM deployment and reference SEP records.

H.21 Appendices
Added Appendix F (CPO format), Appendix G (AAL example),
Appendix H (this change log).
Updated Appendices A, B, C, E for triple-hash fields.

Acknowledgments

The author acknowledges the foundational work of the IETF SCITT working group, the OpenTimestamps community, and the DataCite consortium whose infrastructure and standards underpin RSP deployments. The REM Protocol triple-layer permanence infrastructure described in this document was designed and implemented by the author.

Author’s Address

Lawrence John Reilly Jr.
Independent Researcher

Email: lawrencejohnreilly@gmail.com
IETF Datatracker:
https://datatracker.ietf.org/person/lawrencejohnreilly@gmail.com
Zenodo Archive: https://zenodo.org/records/17103522

REM Verification:
  https://rem-protocol-agent-production.up.railway.app