Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 20 June 2026

P. Rust
Clearpath Intelligence

December 2025

Transparency Record Protocol for AI Governance
draft-rust-trp-00

Abstract

This document specifies the Transparency Record Protocol (TRP), an application-layer protocol for propagating AI transparency metadata across distributed systems. TRP enables organizations to maintain cryptographic chain of custody for AI-generated content, enforce governance policies at network boundaries, and provide verifiable audit trails for regulatory compliance.

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 4 June 2026.

Copyright Notice

Table of Contents

1. Introduction

The rapid deployment of AI systems across critical infrastructure requires standardized mechanisms for transparency and accountability. Current approaches rely on application-specific logging that cannot be verified across organizational boundaries.

The Transparency Record Protocol (TRP) addresses this gap by providing:

* Cryptographically signed audit trails for AI decisions

* Zero-trust identity verification using X.509 PKI

* Network-boundary policy enforcement via inspection headers

* Extensible metadata for quality metrics and governance frameworks

TRP is designed to integrate with existing HTTP infrastructure through custom headers, requiring no changes to application payloads.

2. Conventions and Definitions

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.

2.1. Terminology

TRP Record A JSON document containing transparency metadata for a single request-response cycle or streaming session.

Entity A service, system, or organization identified by a Decentralized Identifier (DID) and authenticated via X.509 certificate chain.

Processing Step A single hop in the request processing chain, recorded with timestamp and cryptographic signature.

Composite Confidence A cumulative trust score that decays as a record propagates across organizational boundaries.

3. Protocol Overview

TRP operates as an overlay protocol on HTTP/1.1 and HTTP/2. Each HTTP request carrying TRP metadata includes:

1. A compressed context header containing the full TRP record

2. An uncompressed inspection header for fast firewall decisions

3. A signature header containing the most recent cryptographic signature

Upon receiving a TRP-enabled request, a compliant service:

1. Validates the signature chain
2. Verifies entity identity against allow-lists
3. Checks security clearance compatibility
4. Enforces governance requirements
5. Appends its own processing step and signature
6. Forwards the updated record

4. Header Transport

TRP-compliant HTTP messages MUST include the following headers:

4.1. X-Transparency-Context

Contains the Base64-encoded, zstd-compressed JSON TRP record.

X-Transparency-Context: <base64-zstd-json>

4.2. X-Transparency-Protocol

Indicates the TRP version.

X-Transparency-Protocol: TRP/1.1

4.3. X-Transparency-Signature

Contains the Ed25519 signature of the current record state.

X-Transparency-Signature: <base64-signature>

4.4. X-Transparency-Inspection

Contains an uncompressed JSON summary for inline firewall evaluation.

X-Transparency-Inspection: {"record_id":"...","security_level":"SECRET"}

5. Record Structure

The TRP record is a JSON object with the following top-level fields:

5.1. Record Header

protocol REQUIRED. String. Must be "TRP/1.1".

id REQUIRED. String. UUID v4 identifying this record.

correlation_id OPTIONAL. String. UUID linking related records.

started_at REQUIRED. String. ISO 8601 timestamp.

validity OPTIONAL. Object containing freshness constraints.

5.2. Entity Context

The entity_context object identifies the originating service:

organization REQUIRED. String. DID of the organization.

service_id REQUIRED. String. Unique service identifier.

certificate_chain REQUIRED. Array of Base64-encoded X.509 certificates.

5.3. Security Context

The security_context object defines access control:

domain REQUIRED. String. Security domain identifier.

level REQUIRED. String. Classification level.

previous_level OPTIONAL. String. Prior classification.

update_reason OPTIONAL. String. Reason for level change.

5.4. Governance Context

The governance_context object identifies compliance frameworks:

frameworks_applied REQUIRED. Array of framework identifiers.

policy_version REQUIRED. String. Policy version.

compliance_assertions OPTIONAL. Array of compliance claims.

6. Signature Chain

TRP uses a cumulative signature chain to ensure tamper evidence. Each processing step appends a signature covering:

1. The hash of all previous content (SHA-256)

2. The current processing step data

3. A timestamp

The signature algorithm MUST be Ed25519 as specified in [RFC8032].

Signature verification MUST validate the entire chain from origin to the current step. Any invalid signature indicates tampering.

7. Policy Enforcement

TRP-compliant firewalls SHOULD implement the following checks:

7.1. Integrity Verification

Validate all signatures in the chain.

7.2. Freshness Verification

Check that validity.expires_at has not passed.

7.3. Identity Verification

Verify entity_context.organization is in the allow-list.

7.4. Clearance Verification

Verify security_context.level does not exceed local clearance.

7.5. Governance Verification

Verify required frameworks are present in governance_context.

7.6. Egress Verification

Verify destination is in egress_context.authorized_recipients.

8. Security Considerations

8.1. Cryptographic Strength

TRP relies on Ed25519 signatures and SHA-256 hashes. These algorithms are considered secure as of this writing.

8.2. Replay Attacks

The validity.expires_at field provides replay protection. Receivers SHOULD reject records with expired timestamps.

8.3. Key Management

Entity private keys MUST be stored in hardware security modules (HSMs) or equivalent secure enclaves in production deployments.

8.4. Denial of Service

Large TRP records may be used for DoS attacks. Implementations SHOULD enforce maximum record sizes (RECOMMENDED: 64KB compressed).

9. IANA Considerations

This document requests registration of the following HTTP headers:

* X-Transparency-Context
* X-Transparency-Protocol
* X-Transparency-Signature
* X-Transparency-Inspection
* X-Transparency-Rejection
* X-Transparency-Rejection-Reason
* X-Transparency-Rejection-Policy
* X-Transparency-Rejection-Details

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[EU-AI-ACT]
European Parliament, "Regulation on Artificial Intelligence", 2024.

[NIST-AI-RMF]
National Institute of Standards and Technology, "AI Risk Management Framework", January 2023.

Appendix A. ABNF Grammar

The following ABNF grammar defines the TRP header format:

```
TRP-Context-Header   = "X-Transparency-Context" ":" SP Base64-Value
TRP-Protocol-Header  = "X-Transparency-Protocol" ":" SP "TRP/" Version
TRP-Signature-Header = "X-Transparency-Signature" ":" SP Base64-Value
TRP-Inspection-Header = "X-Transparency-Inspection" ":" SP JSON-Object

Version      = 1*DIGIT "." 1*DIGIT
Base64-Value = 1*( ALPHA / DIGIT / "+" / "/" / "=" )
JSON-Object  = "{" *( JSON-Member ) "}"

Security-Level = "PUBLIC" / "INTERNAL" / "CONFIDENTIAL" /
                 "RESTRICTED"

DID                = "did:" Method ":" Method-Specific-ID
Method             = 1*ALPHA
Method-Specific-ID = 1*( ALPHA / DIGIT / "." / "-" / "_" / ":" )
```

Appendix B. Example Records

B.1. Minimal Valid Record

```
{
  "protocol": "TRP/1.1",
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "started_at": "2025-12-17T14:30:00Z",
  "entity_context": {
    "organization": "did:web:example.gov",
    "service_id": "gateway",
    "certificate_chain": ["MIIB..."]
  },
  "security_context": {
    "domain": "COMMERCIAL",
    "level": "PUBLIC"
  },
  "signatures": []
}
```

Acknowledgments

The author would like to thank the AI governance and security communities for their input on this specification.

Author’s Address

Pierre Rust
Clearpath Intelligence
Email: pierre.rust@clearpathintel.com