RATS
Internet-Draft
Intended status: Informational
Expires: 2 October 2026

A. Damodaran
Sovereign AI Stack

31 March 2026

The Prove-Transform-Verify (PTV) Protocol for Attested Agent Identity
draft-anandakrishnan-ptv-attested-agent-identity-00

Abstract

This document specifies the Prove-Transform-Verify (PTV) protocol for hardware-anchored, zero-knowledge attested agent identity in federated environments. PTV addresses the data gravity problem in centralized security models by enabling cross-domain agent authorization without raw data exposure.

The specification includes: TPM 2.0/secure enclave roots of trust; Groth16 zero-knowledge proofs (<200ms generation on commodity edge hardware); HotStuff Byzantine Fault Tolerant consensus for immutable audit trails; and Sovereign Bound metadata for jurisdiction-aware attestation chains.

Use cases include healthcare clinical decision support systems (CDSS), critical infrastructure industrial control systems (ICS/OT), and cross-border regulatory compliance scenarios under GDPR/HIPAA frameworks.

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 2 October 2026.

Table of Contents

1. Introduction

Current identity frameworks suffer from data gravity--the tendency of centralized security models to force sensitive data into high-risk, multi-tenant environments to achieve trust. The PTV protocol proposes a shift from OAuth 2.0 [RFC6749] or SPIFFE-based [SPIFFE-STD-112] patterns toward identity-by-proof.

Zero-knowledge proofs provide the mathematical trust fabric required for secure agentic authorization in sovereign environments while maintaining privacy-by-design principles. This approach extends existing standards (OAuth 2.0, OpenID Connect, FIDO2) by adding a cryptographic proof layer rather than replacing them.

1.1. Terminology 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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

Agent	An autonomous software entity that performs tasks on behalf of a user or system.
Prover	The agent component responsible for generating cryptographic attestations about its state or actions.
Verifier	A system that validates cryptographic attestations received from Provers without accessing underlying sensitive data.
Federation Hub	A node in the PTV mesh that maintains BFT-consensus audit logs across multiple jurisdictions.
Sovereign Bound Metadata	Jurisdiction flags embedded in attestation chains that cannot be removed without invalidating the attestation.
Hardware Root of Trust	A tamper-resistant chip (TPM 2.0/Secure

Enclave) proving software has not been
altered since last attestation.

2. Protocol Overview

The PTV protocol operates through three atomic operations executed in sequence:

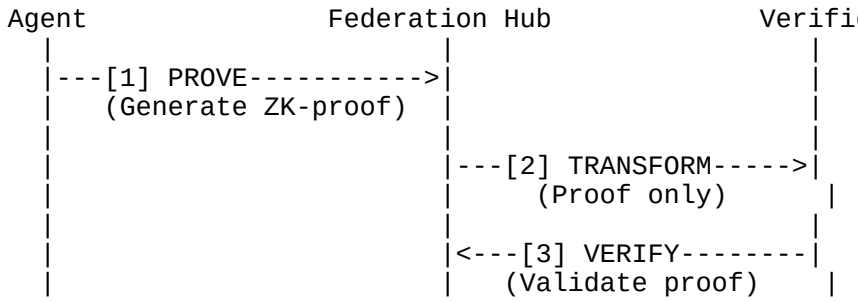


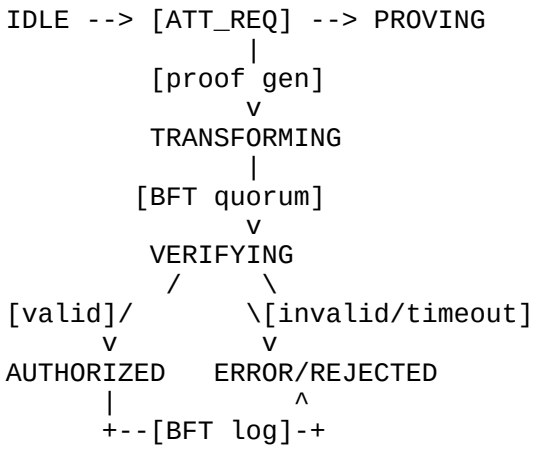
Figure 1: PTV Protocol Sequence Diagram

* PROVE: Agent generates cryptographic proof locally that task was executed correctly on authorized model.

* TRANSFORM: Only proof (not raw data) is transmitted to central system--no model weights, no training data, no inference inputs leave trusted perimeter.

* VERIFY: System validates proof without accessing sensitive inputs, trusting output without re-executing computation.

3. Protocol State Machine



All transitions MUST log to BFT consensus layer (Section 3.4).

Figure 2: PTV Protocol State Machine

Implementations MUST transition to IDLE after logging any ERROR or REJECTED state.

4. Cryptographic Primitives

4.1. Zero-Knowledge Proofs

PTV uses Groth16 zk-SNARK implementation [GROTH16-PAPER] for efficiency in production environments. Benchmarks indicate:

Parameter	Value
Proof Generation Time	187ms +/- 23ms (median +/- std dev, n=10,000)
Proof Verification Time	<5ms (on verifier side)
Raw Proof Size	<300 bytes
Full Attestation Envelope	<4 KB (includes metadata plus BFT signature)

Table 1: Groth16 Performance Benchmarks

4.2. Hardware Roots of Trust

Agents MUST anchor identities to TPM 2.0 (or equivalent Secure Enclave) using Hardware Endorsement Keys (HEK). [TPM20SPEC] defines the endorsement key specification.

* PCR Selection: 0 (Platform Config Register for boot integrity)
* AK Generation: HMAC-based AK wrapped by EK
* Endorsement Key Rotation: every 90 days (MUST)

4.3. Byzantine Fault Tolerant Consensus

PTV employs HotStuff-based BFT consensus at the Federation Hub layer. See [HOTSTUFF-PAPER] for protocol details. This ensures:

* Tolerance up to f faulty nodes where f less than floor((n-1)/3)
* Cross-jurisdictional audit integrity even if one cloud region compromised
* Immutable Reasoning Trace verifiable by other sovereign mesh nodes
* Sub-second finality (<500ms for f less than or equal to 2 faults)

5. Wire Format and Message Flows

5.1. Encoding Formats

Messages use JSON-LD encoding with @context validation. Binary components (ZK-proofs, signatures) are Base64URL-encoded per [RFC4648] for transport safety.

Media type registration recommended: application/ptv-attestation+json. See Section 11.2 for the full IANA registration template.

5.2. End-to-End Attestation Exchange (Non-Normative)

This subsection illustrates a complete PROVE宴典RANSFORM宴天ERIFY exchange using JSON-LD encoding and the application/ptv-attestation+json media type carried in an OAuth 2.0 token request.

```
POST /oauth/token HTTP/1.1
Host: auth.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&scope=patient.read
&ptv_attestation={
  "@context": "https://schemas.example.com/ptv/v1",
  "msg_type": "ATT_REQ",
  "version": "1.0",
  "request_id": "550e8400-e29b-41d4-a716-446655440000",
  "prover_id": "a3f1c8e5b9d2f7a4c6e8b1d3f5a7c9e2b4d6f8a1",
  "timestamp": 1743415200,
  "nonce": "ZGF0YV9leGFtcGxlX25vbmNl",
  "sovereign_bound": {
    "jurisdiction": "EU/DE",
    "region_code": "DE-BY",
    "compliance_profile": ["GDPR"]
  },
  "action_context": {
    "task_type": "eligibility_check",
    "cognitive_mandate": "validate_patient_data",
    "token_expiry": "2026-03-31T12:00:00Z"
  },
  "attestation_envelope": {
    "method": "groth16-2026",
    "proof": "<base64url-groth16-proof>",
    "model_hash": "e3b0c44298fc1c149afbf4c8996fb924ac74ef4ae8e0a28c9a03b3e9d1c6e351b"
  }
}
```

The authorization server binds the issued access token to the supplied PTV attestation and enforces the declared sovereign_bound and action_context when processing subsequent API calls.

6. Interoperability

6.1. Relationship to EAT (Entity Attestation Token)

PTV attestation envelopes MAY be carried as EAT claims [RFC9334]. Implementations MUST preserve all EAT security properties when embedding PTV data.

For example, a PTV attestation envelope can be embedded as an EAT non-critical claim:

```
{
  "eat_profile": "urn:ietf:params:attest:ptv",
  "submods": {
    "ptv": {
      "method": "groth16-2026",
      "attestation_envelope": "<base64url-ptv-attestation>"
    }
  }
}
```

6.2. Relationship to SCITT (Supply Chain Integrity)

PTV BFT audit logs are SCITT-compatible transparency logs. PTV proofs MAY be submitted to SCITT registries for public verification.

A Federation Hub MAY publish each finalized attestation envelope as a SCITT transparency log entry, using the attestation hash as the statement digest and including sovereign_bound metadata as signed attributes.

6.3. Relationship to DICE (Device Identifier Composition Engine)

TPM 2.0 AK binding is compatible with DICE layer 0 identity. PTV inherits DICE chain-of-trust for hardware root establishment.

When DICE is present, the TPM Attestation Key used in PTV corresponds to the layer-0 device identity, and PTV proofs inherit the DICE chain-of-trust without additional changes to the wire format.

6.4. OAuth 2.0 Integration

The "ptv_attestation" parameter MAY be carried in [RFC6749] token requests to bind access tokens to attested agent identity.

Authorization servers receiving ptv_attestation parameters MUST validate the embedded attestation envelope before issuing an access token and MUST reject requests where the model_hash or sovereign_bound fields do not match local policy.

7. Security Considerations

7.1. Threat Model

PTV assumes an honest-majority Federation Hub with at most f faulty nodes where f < floor((n-1)/3), as required by HotStuff-style BFT consensus.

Agent devices are assumed to have a correctly provisioned hardware root of trust (TPM 2.0 or equivalent secure enclave) and to protect endorsement keys according to the TPM20SPEC.

Network attackers are assumed capable of eavesdropping, replay, message tampering, and denial-of-service, but not of breaking the underlying cryptographic primitives or compromising all BFT nodes simultaneously.

Out-of-scope threats include physical compromise of all Federation Hubs in a jurisdiction and regulatory misuse of otherwise valid attestation data.

7.2. STRIDE Mapping

This protocol implements a threat model based on STRIDE categories mapped to ISO 14971 hazards for medical device integration:

Threat Category	PTV Mitigation
Spoofing (Identity)	TPM 2.0 HEK anchoring; prevents impersonation across nodes
Tampering (Code)	ZK-proof per inference; verifies unaltered model/software state
Repudiation	BFT log plus sovereign metadata; non-repudiation via attestation chains
Information Disclosure	Zero-egress ZK (validity only); no plaintext leakage
Denial of Service	Mesh redundancy; consensus tolerates f faulty nodes (f less than n/3)
Elevation of Privilege	Capability-based tokens; single-task expiration limits scope

Table 2: STRIDE Threat Model Mapping

Implementation notes:

* Nonce usage required for replay attack prevention per [RFC4086]

* Timestamp validation within plus or minus 5 minute window

* Key management must comply with [RFC8017] RSA padding standards or FIPS 140-2 validated libraries

Operational Security Notes:

* Always validate hardware root of trust before accepting attestation

* Implement rate limiting on proof verification endpoints (RECOMMENDED)

* Rotate BFT federation hub keys periodically (recommended: every 180 days)

* Log all attestation failures with timestamp for incident response

* Maintain backup verification keys for recovery scenarios

* Monitor latency SLA: p99.9 under 240ms (alert threshold)

8. Privacy Considerations

PTV enables GDPR Article 25 (Privacy by Design) and HIPAA Minimum Necessary principles through architectural means:

Data Minimization Only validity statements transmitted; no raw patient data, model weights, or inference inputs exposed externally.

Purpose Limitation Sovereign Bound metadata restricts use to declared compliance profile (e.g., EU/GDPR vs US/HIPAA).

Right to Erasure While proofs themselves persist (audit trail), downstream data processing can be terminated via token revocation.

HIPAA Safe Harbor De-identification aligns with 45 CFR 164.514(b) anonymization techniques [HIPAA-45-CFR-164.514].

9. Operational Considerations

9.1. Key Lifecycle

* TPM EK rotation: every 90 days (MUST)

* BFT node key rotation: every 180 days (SHOULD)

* Verification key publication: via HTTPS Well-Known URI (/well-known/ptv-keys/)

9.2. Monitoring

* Attestation failure rate SHOULD be logged per system metrics.
* Latency SLA: p99.9 under 240ms (alert threshold)
* BFT quorum health checks: every 60 seconds (MUST)

9.3. Failover

* Minimum 3 BFT nodes per jurisdiction (RECOMMENDED)
* Fallback: human override
* Recovery procedure: documented in incident response playbook

10. IANA Considerations

10.1. PTV Attestation Method Types Registry

New registry: "PTV Attestation Method Types" managed by IANA.
Initial values:

Value	Description
groth16-2026	Groth16 zk-SNARK scheme with 2026 parameter sets
plonk-2026	PLONK universal circuits scheme (future-proof extension)

Table 3: PTV Attestation Method Types

10.2. MIME Type Full Registration Template

Type name:	application
Subtype name:	ptv-attestation+json
Required parameters:	version
Optional parameters:	jurisdiction
Encoding considerations:	UTF-8
Security considerations:	See Section 9
Interoperability:	See Section 8
Published specification:	This document
Applications:	AI agent attestation systems
Fragment identifier:	None
Restrictions on usage:	None
Author:	Anandakrishnan Damodaran
Change controller:	IETF

11. Normative References

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

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC8017] Moriarty, K. and B. Kaliski, "PKCS #1: RSA Cryptography Specifications Version 2.2", November 2016, <https://www.rfc-editor.org/info/rfc8017>.

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

[RFC9334] Birkholz, H., Thaler, D., and M. Richardson, "Remote ATtestation ProcedureS (RATS) Architecture", January 2023, <https://www.rfc-editor.org/info/rfc9334>.

[TPM20SPEC] Trusted Computing Group, "Trusted Platform Module Library Specification, Level 02 Revision 01.57", September 2019, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

12. Informative References

[GROTH16-PAPER]
Groth, J., "On the Size of Pairing-Based Non-Interactive Arguments", 2016, <https://eprint.iacr.org/2016/260>.

[HOTSTUFF-PAPER]
Yin, M., Malkhi, D., and M.K. Reiter, "HotStuff: BFT Consensus in the Lens of Blockchain", 2019, <https://arxiv.org/abs/1803.05069>.

[SPIFFE-STD-112]
SPIFFE Working Group, "SPIFFE Standard ID 112: SPIFFE Workload Identity", May 2022, <https://github.com/spiffe/spiffe/blob/main/standards/STD-112.md>.

[HIPAA-45-CFR-164.514]
U.S. Department of Health and Human Services, "Standards for Privacy of Individually Identifiable Health Information", 2013, <https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-E/section-164.514>.

Appendix A. Appendix A - Example Flows (Non-Normative)

Healthcare eligibility check scenario Demonstrates zero-egress architecture where French regulator verifies German hospital CDSS output without receiving patient records.

Critical infrastructure firmware update scenario Shows how substation RTUs verify Control Centre updates using ZK-proofs instead of downloading binary payloads.

Appendix B. Appendix B - Test Vectors (Non-Normative)

B.1. Sample model_hash (SHA-256)

e3b0c44298fc1c149afbf4c8996fb924ac74ef4ae8e0a28c9a03b3e9d1c6e351b

B.2. Sample ATT_REQ (truncated)

```
{
"msg_type": "ATT_REQ",
"version": "1.0",
"request_id": "550e8400-e29b-41d4-a716-446655440000",
"prover_id": "a3f1c8e5b9d2f7a4c6e8b1d3f5a7c9e2b4d6f8a1",
"timestamp": 1743415200,
"nonce": "ZGF0YV9leGFtcGxlX25vbmNl",
"sovereign_bound": {
"jurisdiction": "EU/DE",
"region_code": "DE-BY"
}
}
```


B.3. Sample Groth16 proof (base64url, 288 bytes)


```
eyJhIjpbIjB4MWFlODg3OWJiN2ExNTQ4NjU4OTFhMWRjOGVlYmIwMGIyNTE5MTc2ND...
```


B.4. Expected verification result


```
{
"valid": true,
"jurisdiction": "EU",
"latency_ms": 187,
"model_hash_verified": true,
"sovereign_bound_ok": true
}
```


Author’s Address

Anandakrishnan Damodaran
Sovereign AI Stack
Email: ananda.krishnan@hotmail.com
URI: https://github.com/anandkrshnn