RATS
Internet-Draft
Intended status: Standards Track
Expires: 16 December 2025

K. Huang
DistributedApps.ai
J. Huang
Kleiner Perkins

14 June 2025

Extending Certificate Enrollment Protocols for Scalable Agentic AI Identity

draft-huang-acme-scalable-agent-enrollment-00

Abstract

Agentic AI systems require robust, verifiable identities to operate securely. While traditional certificate enrollment protocols like SCEP and EST can be extended to include remote attestation evidence, performing a full validation for every agent's certificate request presents significant scalability and privacy challenges. This document proposes two distinct, scalable models for integrating attestation into the enrollment lifecycle. The first model leverages Zero-Knowledge Proofs (ZKP) to provide private, efficient, and continuous attestation. The second model uses a one-time, high-assurance bootstrapping process to establish a trusted host environment, which is then authorized to endorse certificate requests for the agents it runs. Both models enable high-assurance identity for AI agents while addressing the performance bottlenecks of naive per-enrollment verification.

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 16 December 2025.

Table of Contents

1. Introduction

The need for strong, verifiable identity for Agentic AI is well-established. Our related work, "Capability Attestation Extensions for the Entity Attestation Token (EAT)" [I-D.huang-rats-agentic-eat-cap-attest], defines *what* an agent should attest to. This document addresses *how* an agent can obtain a certificate reflecting that attestation in a scalable and privacy-preserving manner.

The naive approach of sending full attestation evidence with every SCEP or EST enrollment request is computationally expensive for the verifying Registration Authority (RA) and can expose sensitive

details about the agent's software stack. This document reframes the problem to prioritize scalability, proposing two advanced enrollment models designed for large-scale, dynamic agent deployments. These models allow enterprises to leverage the automation of SCEP [RFC8894] and EST [RFC7030] without creating a performance or management bottleneck.

2. Terminology and Conventions

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.

Attestation-Aware RA: A Registration Authority, acting as a SCEP or EST server, capable of functioning as a RATS Verifier.

Capability Certificate: An X.509 certificate containing extensions encoding verifiable claims about the subject's authorized capabilities.

Host Endorser: A software service running on a trusted host that is authorized to sign or endorse certificate requests on behalf of the agents it manages.

Attestation Circuit: A logical representation of a validation policy, encoded for use in a Zero-Knowledge Proof system.

3. Problem Statement: The Scalability Challenge of Per-Enrollment Attestation

A single enterprise or cloud environment may run thousands or even millions of ephemeral AI agents. A centralized RA performing a full attestation validation for every single enrollment request faces three major, prohibitive issues:

* *Performance Bottleneck:* Validating complex evidence (e.g., checking multiple software hashes, validating endorsement chains) for every request creates a centralized bottleneck that cripples the system's ability to scale for dynamic, short-lived agents.

* *Privacy Leakage:* Transmitting detailed evidence to a central RA reveals the exact software versions, configurations, and libraries used by every agent, which can be a significant and undesirable information leak.

* *Management Overhead:* The RA must be constantly updated with the "golden measurements" for every valid agent configuration, making the system brittle and difficult to manage in environments with rapid development cycles.

To be viable, any solution MUST address these challenges.

4. Proposed Scalable Enrollment Models

We propose two distinct models that leverage SCEP’s automation while providing high-assurance identity. An implementation MAY support one or both models.

4.1. Model 1: Auto-Enrollment with Zero-Knowledge Proof Attestation

This model uses a Zero-Knowledge Proof (ZKP) to allow an agent to prove it meets all attestation requirements without revealing the evidence itself. This provides an optimal balance of strong security, privacy, and performance.

Core Concept

The RA does not validate the raw evidence. Instead, it validates a small, mathematically-sound proof that the agent possesses valid evidence satisfying a pre-defined set of rules.

Specification and Steps

Phase 1: One-Time Trusted Setup (per Agent Fleet)

This phase is performed once by the entity deploying a fleet of similar agents.

1. *Define the Attestation Circuit:* A ZKP circuit is created. This circuit is the formal specification of the validation policy. It MUST encode rules such as:
    * The private input is a valid Entity Attestation Token (EAT).
    * The EAT is signed by a trusted Endorser Public Key.
    * The EAT contains a specific, required software measurement hash.
    * The EAT’s claims are bound to a public input representing the CSR’s public key hash.

2. *Generate Keys:* A trusted setup ceremony is performed on the circuit to generate a *Proving Key* and a *Verifying Key*.

    * The *Proving Key* MUST be securely distributed to every AI agent in the fleet.

    * The *Verifying Key* MUST be securely provisioned to the Attestation-Aware RA.

Phase 2: Automated Agent Enrollment (per Agent)

This is the fast, scalable, per-agent workflow.

1. *Agent Initialization:* The agent starts, generates a new key pair, and creates a CSR.

2. *Evidence Collection:* The agent collects its full attestation evidence (e.g., its EAT). This evidence REMAINS LOCAL and is not transmitted.

3. *Proof Generation:* The agent uses its *Proving Key* to execute the ZKP circuit. It provides its private evidence as private inputs and the hash of its CSR’s public key as a public input. The output is a compact *ZK-Proof*.

4. *SCEP Request:* The agent initiates a SCEP request. The payload is a ‘multipart/mixed‘ object containing the standard CSR, the generated ZK-Proof, and its public inputs.

5. *RA Verification:* The RA receives the request. It performs a single, rapid ZKP verification using its *Verifying Key*. This validates that the agent possesses the correct private evidence without ever seeing it.

6. *Certificate Issuance:* If the proof is valid, the RA approves the request, and the CA issues the certificate.

4.2. Model 2: Bootstrapped Trust via Host Endorsement

This model amortizes the cost of attestation. A full, heavyweight attestation is performed only **once** to enroll the host environment (e.g., a VM or server). That trusted host is then empowered to issue lightweight endorsements for the agents it runs.

Core Concept

Instead of each agent proving its own integrity from scratch, it asks its already-trusted host to vouch for it. The RA’s job is simplified to checking the host’s "signature."

Specification and Steps

Phase 1: One-Time Host Environment Bootstrap

This is the high-assurance setup performed once per host machine.

1. *Initial Host Attestation:* The host environment performs a full remote attestation using its hardware root of trust (e.g., a TPM). It sends its detailed evidence to a dedicated, high-security **Bootstrap RA**.

2. *Host Certificate Issuance:* The Bootstrap RA validates the host’s integrity. If successful, it instructs the CA to issue a special **Host Identity Certificate**. This certificate MUST have two key properties:

   * A reasonably long lifetime (e.g., 1 year).

   * A specific Extended Key Usage (EKU) OID, ‘id-kp-agentEnroller‘, authorizing it to endorse agent certificate requests.

3. *Device-Bound Key Storage:* The private key corresponding to the Host Identity Certificate **MUST** be stored in the host’s hardware-backed, non-exportable key store.

Phase 2: Automated Agent Enrollment (per Agent)

This is the highly scalable workflow that occurs on the trusted host.

1. *Agent Initialization:* An AI agent starts on the trusted host, generates a new key pair, and creates a CSR.

2. *Request for Endorsement:* The agent sends its CSR to a local **Host Endorser** service.

3. *Host Endorsement:* The Host Endorser service uses the device-bound private key from the Host Identity Certificate to create a detached signature (e.g., a PKCS#7 signature) over the agent’s CSR.

4. *SCEP Request:* The agent initiates a SCEP request to the standard **Agent RA**. The request payload includes its own CSR, the detached signature from the Host Endorser, and the public Host Identity Certificate.

5. *RA Verification (Simplified):* The Agent RA performs two simple, fast checks:

   * Does the provided signature over the CSR successfully verify using the public key from the Host Identity Certificate?

   * Does the Host Identity Certificate contain the required ‘id-kp-agentEnroller‘ EKU and is it still valid?

6. *Certificate Issuance:* If both checks pass, the RA approves the request, and the CA issues a standard, short-lived certificate to the agent.

5. Security Considerations

The security of this system relies on several key factors:

* *Model 1 (ZKP):* The primary security dependency is the integrity of the trusted setup and the secrecy of the *Proving Key*. If a Proving Key is compromised, an adversary can generate valid proofs for a malicious agent. Proving Keys MUST be protected as highly sensitive secrets. The ZKP circuit itself MUST be audited to ensure it correctly and completely represents the security policy.

* *Model 2 (Host Endorsement):* The security of this model hinges on the integrity of the host environment *after* bootstrapping. A compromise of the host could lead to the compromise of the *Host Endorser’s private key*, allowing an attacker to endorse malicious agent CSRs. This risk is mitigated by: 1) storing the key in hardware (TPM), 2) implementing continuous host-level monitoring, and 3) issuing very short-lived certificates to agents, forcing frequent re-endorsement and providing an opportunity to detect anomalies.

* *General:* In both models, it is critical to bind the identity being requested (the CSR’s public key) to the proof or endorsement being provided. This is handled explicitly as a public input in the ZKP model and by signing the full CSR in the Host Endorsement model.

6. Privacy Considerations

The two models presented have different privacy characteristics.

* The ZKP-based model (Model 1) offers strong privacy guarantees. The RA learns only that an agent has satisfied a specific policy, without learning any of the underlying evidence (e.g., software versions, configurations). This is the recommended approach for multi-tenant or privacy-sensitive environments.

* The Host Endorsement model (Model 2) relies on the privacy of the host. While agents do not reveal their details to the central RA, the Host Endorser service has visibility into every agent enrollment it endorses. Implementers MUST ensure the host environment and the Endorser service are sufficiently hardened to protect this local information.

7. IANA Considerations

This document requests IANA to make the following allocations:

1. An Object Identifier (OID) for the ‘id-kp-agentEnroller‘ Extended Key Usage from the "SMI Security for PKIX Key Purpose" registry. This OID is required for Model 2 (Host Endorsement).

2. (Optional/Transitional) An OID for a SCEP CSR attribute to carry ZKP data, under the ‘pkcs-9-at‘ registry, for implementations that do not use a multipart message format. TBD.

8. Normative References

[RFC7030] Pritikin, M., Wouters, P., Richardson, D., Turner, T., and S. Boeyen, "Enrollment over Secure Transport", October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC8894] Gutmann, P., "Simple Certificate Enrolment Protocol", September 2020, <https://www.rfc-editor.org/info/rfc8894>.

9. Informative References

[I-D.huang-rats-agentic-eat-cap-attest] Huang, K. and J. Huang, "Capability Attestation Extensions for the Entity Attestation Token (EAT) in Agentic AI Systems", June 2025.

Authors’ Addresses

Ken Huang
DistributedApps.ai
Email: ken.huang@DistributedApps.ai

Jerry Huang
Kleiner Perkins
Email: huangjerry03@gmail.com