SEAT
Internet-Draft
Intended status: Informational
Expires: 17 May 2026

Y. Jiang
D. Wang

13 November 2025

Dynamic Attestation for AI Agent Communication
draft-jiang-seat-dynamic-attestation-00

Abstract

This document describes a use case for conveying remote attestation information in association with Transport Layer Security (TLS) sessions in the context of AI agent communication. It focuses on long-lived secure channel sessions where an AI agent runtime posture, covering the platform Trusted Computing Base (TCB), agent manifest (models, tools and policies) and committed runtime context, can change frequently and unpredictably. The document highlights requirements for dynamic attestation so that relying parties can base authorization decisions on the current runtime posture of the communicating agent.

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 17 May 2026.

Copyright Notice

Table of Contents

1. Introduction

Many deployed systems involve relatively static workloads, where software and configuration change primarily at onboarding or planned maintenance events. In contrast, AI agents may alter effective behavior by updating their models, tools, and policies on timescales that are not necessarily aligned with TLS session establishment or traditional maintenance cycles. Such updates may occur while long-lived TLS sessions remain open, or between sessions that are resumed or quickly re-established, raising security and privacy concerns about authorizing sensitive actions based on stale assumptions.

Remote attestation (RA), as described in the Remote Attestation Procedures (RATS) architecture [RFC9334], allows a relying party to obtain information about the software and hardware state of a remote endpoint and to assess whether that state satisfies local policy. When combined with a secure channel protocol such as TLS 1.3 [RFC8446], attestation information can be bound to a specific, authenticated session so that authorization decisions reflect the peer's current runtime posture (e.g., platform TCB plus an agent manifest covering model, tools and policy), rather than only the state at connection establishment.

This document describes a use case for dynamic attestation in AI agent communication scenarios and outlines requirements for obtaining fresh, channel-bound attestation information without unnecessarily disrupting existing connections, allowing relying parties to base authorization on agent state at the time of use.

2. 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 RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terminology defined in the RATS architecture [RFC9334] and in TLS 1.3 [RFC8446], including "Attestation", "Relying Party", "Evidence" and "Attestation Results".

Trusted Computing Base (TCB) of a device, is used to refer to security-relevant components: hardware, firmware, software, and their respective configurations.

Runtime posture denotes the set of attestation-relevant claims about the communicating endpoint at the time of use, sufficient for a relying party to appraise policy. Its exact composition is deployment- specific and may include claims about the platform, execution environment, configuration, or context.

3. Dynamic Attestation for AI Agent Communication

Goal: Support remote attestation for AI agent communication over secure channel sessions where the software and configuration on at least one endpoint can change on timescales that are not aligned with connection establishment or traditional maintenance cycles (for example, while long-lived sessions remain open or between sessions that are resumed), while the endpoint continues to exchange traffic with multiple network peers.

3.1. AI Agent Runtime Attestation

Use case: AI Agent Runtime Attestation. An AI agent service communicates with other network nodes over TLS 1.3, invoking tools that can access sensitive data or change network state. The agent’s model, tools, and policies may be updated dynamically according to operator policy, and these updates are not guaranteed to align with TLS session boundaries or a fixed maintenance schedule. Peers therefore need assurance that, at the time of a high-impact operation, the agent runs on an acceptable platform and uses an approved combination of model, code, and policy.

* Requirement 1 (fresh, channel-bound attestation): Before executing a high-impact operation over an existing secure channel, a peer MUST be able to obtain fresh attestation evidences that are cryptographically bound to that session and that reflect both the platform and the current agent-level state relevant to authorization, as determined by local policy.

* Requirement 2 (dynamic attestation): The mechanism used to obtain such attestation evidences SHOULD support lightweight, dynamic attestation without necessarily requiring a full new TLS handshake, so that changes to the runtime posture become visible to relying parties when required by local policy.

4. Security Considerations

Any mechanism that supports dynamic attestation for AI agent communication should ensure that attestation evidences are strongly bound to the TLS session for which they are intended, so that they cannot be replayed or relayed to another session or endpoint.

Because the agent software stack can change rapidly, attestation evidences should be sufficiently fresh for the relying party's policy. Implementations should provide a way to limit the validity period of evidences for a given session and to trigger attestation when that period expires or when relevant changes are detected.

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", May 2017, <https://datatracker.ietf.org/doc/html/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", August 2018, <https://datatracker.ietf.org/doc/html/rfc8446>.

[RFC9334] Birkholz, H., Fossati, T., Hardjono, T., and N. Smith, "Remote Attestation Procedures Architecture", January 2023, <https://datatracker.ietf.org/doc/html/rfc9334>.

6.2. Informative References

[RFC9261] Sullivan, N. and C. A. Wood, "Exported Authenticators in TLS and DTLS", July 2022, <https://datatracker.ietf.org/doc/html/rfc9261>.

Authors' Addresses

Yuning Jiang
Singapore
Email: jiangyuning2@h-partners.com

Donghui Wang
China
Email: wangdonghui124@huawei.com