Secure Evidence and Attestation Transport
Internet-Draft
Intended status: Informational
Expires: 18 July 2026

M. U. Sardar
TU Dresden

14 January 2026

Pre-, Intra- and Post-handshake Attestation draft-usama-seat-intra-vs-post-00

Abstract

This document presents a taxonomy of extending TLS protocol with remote attestation, referred to as attested TLS. It also presents high-level analysis of benefits and limitations of each category, namely pre-handshake attestation, intra-handshake attestation and post-handshake attestation.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://muhammad-usama-sardar.github.io/seat-intra-vs-post/draft-usama-seat-intra-vs-post.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-usama-seat-intra-vs-post/.

Discussion of this document takes place on the Secure Evidence and Attestation Transport Working Group mailing list (mailto:seat@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/seat. Subscribe at https://www.ietf.org/mailman/listinfo/seat/.

Source for this draft and an issue tracker can be found at https://github.com/muhammad-usama-sardar/seat-intra-vs-post.

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/.

Copyright Notice

Table of Contents

1. Introduction

Based on our extensive analysis of attested TLS [Tech-Concepts], we classify attested TLS into three main categories:

* pre-handshake attestation,

* intra-handshake attestation, and

* post-handshake attestation.

In pre-handshake attestation, the signing of Claims [Tech-Concepts] precedes the TLS handshake, while post-handshake attestation applies the reverse. Intra-handshake attestation requires the signing of Claims to be done within the TLS handshake protocol.

In this version, we analyze the three categories (without combinations) with a focus on the last two. Regarding remote attestation, we note that:

| Remote attestation provides guarantees about the state of Attester
| *only* at the time at which signing of Claims is done to generate
| Evidence [Tech-Concepts].

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.

We use terminology from [RFC9334] and [I-D.ietf-tls-rfc8446bis] slightly loosely (intentionally) for readability. Future versions will tighten it.

In addition, we define three temporal terms:

* *Evidence Generation Time*: Time when Evidence is generated (more specifically when Claims are signed)

* *Connection Establishment Time*: Time at which TLS handshake is performed

* *Lifetime of Connection*: Time period starting from Evidence Generation Time until the connection exists.

3. Pre-handshake Attestation

Since the Evidence Generation Time could be at any arbitrary point of time in the past compared to the connection establishment time, pre-handshake attestation provides no guarantees about the state of Attester at the Evidence Generation Time and during the Lifetime of Connection.

4. Intra-handshake Attestation

Intra-handshake attestation improves the situation where Evidence Generation Time is the same as Connection Establishment Time.

In following subsections, we present the benefits and limitations of intra-handshake attestation.

4.1. Benefits

4.1.1. No Additional Application-Level Protocol

Intra-handshake attestation does not require a new application-layer protocol or message exchange. Evidence and related metadata are conveyed within handshake via TLS extensions. TLS is responsible for conveyance of the Evidence; it does not perform appraisal of Evidence or authorization. Appraisal of Evidence, policy evaluation, and trust decisions are performed by application-level components that consume the attestation properties exposed by the TLS stack. As a result, while no new application-layer protocol is required, applications do incorporate additional trust logic to interpret attested connection properties and make security-relevant decisions.

4.1.2. Avoids Extra Round Trips for One-time Attestation

It avoids extra round trips for use cases which require remote attestation only once during Connection Establishment Time.

4.2. Limitations

4.2.1. Limited Claims Availability

Since limited Claims are available at the Evidence Generation Time, it does not provide complete security posture of the Attester, such as runtime integrity of Attester.

4.2.2. Invasive Changes in TLS

To be made secure, it requires invasive changes in TLS protocol, as deep as key schedule and adding or modifying existing handshake messages [ID-Crisis].

4.2.3. State After Connection Establishment Not Covered

It provides no guarantees about the state of Attester during the lifetime of connection. This is a security concern in long-lived connections where state of Attester may change after Connection Establishment Time.

4.2.4. High Handshake Latency

Because of signature in Evidence generation and verification of signatures during appraisal, this leads to high handshake latency. This may not be desirable for some applications.

5. Post-handshake Attestation

Post-handshake attestation improves the situation further by signing the Claims during Lifetime of Connection, i.e., at the time when it is actually required. Hence, together with use cases requiring one-time attestation, it covers the use cases of long-lived connections requiring re-attestation. For post-handshake attestation, first round of remote attestation MUST be done immediately after Connection Establishment Time, and Relying Party (RP) [RFC9334] MUST not send any secure data until Evidence is successfully appraised.

In following subsections, we present the benefits and limitations of post-handshake attestation.

5.1. Benefits

In general, it allows re-authentication and re-attestation without tearing down the connection.

5.1.1. Full Claims Availability

Since all Claims are available at the time of post-handshake attestation (during Lifetime of Connection), it provides complete security posture of the Attester.

5.1.2. No Change in TLS

It does not require any change in TLS protocol.

5.1.3. State After Connection Establishment Is Covered

It provides guarantees about the state of Attester during the Lifetime of Connection. This is particularly helpful in long-lived connections where state of Attester may change after Connection Establishment Time.

5.1.4. Standard Handshake Latency

Since the signature in Evidence generation and verification of signatures during appraisal happen after Connection Establishment Time, there is no additional latency.

5.1.5. Avoids Extra Round Trips

Except for first round of remote attestation, post-handshake attestation outperforms the intra-handshake attestation (one round trip), which requires re-establishing the connection (1.5 round trip).

5.2. Limitations

5.2.1. Impact on Application Layer

Post-handshake attestation may require changes at the application layer. However, changes at the application layer do not necessarily imply modifications to application business logic or data exchange protocols. Attestation-related functionality may be realized via application-level signalling (Exported Authenticators [RFC9261]) and trust logic, which may be implemented in intermediary components (e.g., proxies, sidecars, or middleware) on both client and server sides. These components are responsible for exchanging and appraising attestation evidence and enforcing trust or authorization decisions before application data is processed. This is analogous to common production deployments in which TLS termination and certificate handling are performed by a fronting proxy, while the application itself remains unchanged and resides behind it.

6. Need for Post-handshake Attestation

We argue that post-handshake attestation is unavoidable (e.g., re-attestation to track changes after Connection Establishment Time for long-lived connections). Use cases where pre-handshake attestation and intra-handshake attestation are insufficient include include AI agents/agentic AI [I-D.jiang-seat-dynamic-attestation].

Intra-handshake attestation only adds unnecessary complexity which is avoidable. All use cases of intra-handshake attestation can be covered by post-handshake attestation (by doing attestation round immediately after Connection Establishment Time) but not the other way around.

6.1. IoT Constraints

[SEAT-Charter] includes TLS client as RATS Attester. Client could be a low-power IoT device. There are use cases where periodic or on-demand attestation is required, such as periodic attestation for long-lived, low-power IoT devices or in IoT swarms that need to synchronize software versions before coordinated operations or after configuration updates.

Moreover, we note some observations from LAKE WG:

Michael Richardson shares his insight [MCR-LAKE]:

> I have a half-written document on putting EAT into the full BRSKI protocol. A reason that I stopped is that I realized that doing security posture evaluation at onboarding time (only) wasn't enough. It has to be done regularly. So having a protocol used at onboarding time and another one during normal operation meant that the onboarding one would have bugs that never get fixed, since the code only runs once.

G旦ran Selander observes [Goran-LAKE]:

> Indeed, if the authentication procedure is repeated at a later stage, for whatever reason, e.g. key rotation, it should be possible to repeat the attestation procedure.

7. Existing Implementations

7.1. Intra-handshake Attestation

Prominent implementations of intra-handshake attestation are all vulnerable to relay attacks [RelayAttacks]. Some of them are abusing the extensions of TLS, such as SNI and ALPN, for conveyance of attestation nonce [RelayAttacks].

7.2. Post-handshake Attestation

Google [Keith-STET-CCC], Microsoft [Stunes-vTPM-CCC], and SCONE [SoK-Attestation] are all using post-handshake attestation.

8. Security Considerations

Most of the document is about security considerations. Also, Security Considerations of [RFC9334] and [I-D.ietf-tls-rfc8446bis] apply. In addition:

* Pre-handshake attestation is vulnerable to *replay* [RA-TLS] and *diversion* [ID-Crisis] attacks.

* Without significant changes to the TLS protocol: Intra-handshake attestation is vulnerable to *diversion* attacks [ID-Crisis]. It also does not bind the Evidence to the application traffic secrets, resulting in *relay* attacks [RelayAttacks].

* No attacks on post-handshake attestation are currently known. Post-handshake attestation avoids replay attacks by using fresh attestation nonce. Moreover, it avoids diversion and relay attacks by binding the Evidence to the underlying TLS connection, such as using Exported Keying Material (EKM) [I-D.ietf-tls-rfc8446bis]. [RFC9261] and [RFC9266] provides mechanisms for such bindings.

8.1. Exploit of Sensitive Hardware-level Information

From the view of the TLS server, post-handshake attestation offers better security than intra-handshake attestation when the server acts as the Attester. In intra-handshake attestation, due to the inherent asymmetry of the TLS protocol, a malicious TLS client could potentially retrieve sensitive hardware-level information from the Evidence *without the client’s trustworthiness (i.e., authentication) first being established by the server*. This information (e.g., vulnerable firmware version) can be exploited for attacks. In post-handshake attestation, the server can ask for client authentication and only send the Evidence after successful client authentication.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

10.2. Informative References

[Goran-LAKE] G旦ran Selander, "Comments for remote attestation over EDHOC", May 2024, <https://mailarchive.ietf.org/arch/msg/lake/Bb3eTcQxDA-F1AYJ0hZZy3p9wpQ/>.

[I-D.ietf-tls-rfc8446bis] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-rfc8446bis-14, 13 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8446bis-14>.

[I-D.jiang-seat-dynamic-attestation] Jiang, Y. and Wangdonghui, "Dynamic Attestation for AI Agent Communication", Work in Progress, Internet-Draft, draft-jiang-seat-dynamic-attestation-00, 13 November 2025, <https://datatracker.ietf.org/doc/html/draft-jiang-seat-dynamic-attestation-00>.

[ID-Crisis] Sardar, M. U., Moustafa, M., and T. Aura, "Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS", November 2025, <https://www.researchgate.net/publication/398839141_Identity_Crisis_in_Confidential_Computing_Formal_Analysis_of_Attested_TLS>.

[Keith-STET-CCC]
Keith Moyer, "Split-Trust Encryption Tool Attested Session Handling", March 2022, <https://github.com/CCC-Attestation/meetings/blob/main/materials/KeithMoyer_STET.pdf>.

[MCR-LAKE] Michael Richardson, "Comments for remote attestation over EDHOC", May 2024, <https://mailarchive.ietf.org/arch/msg/lake/RseQknOug41sTzW7xBJ60oRdvq0/>.

[RA-TLS] Sardar, M. U., Niemi, A., Tschofenig, H., and T. Fossati, "Towards Validation of TLS 1.3 Formal Model and Vulnerabilities in Intel’s RA-TLS Protocol", November 2024, <https://www.researchgate.net/publication/385384309_Towards_Validation_of_TLS_13_Formal_Model_and_Vulnerabilities_in_Intel’s_RA-TLS_Protocol>.

[RelayAttacks]
Sardar, M. U., "Relay Attacks in Intra-handshake Attestation for Confidential Agentic AI Systems", January 2026, <https://mailarchive.ietf.org/arch/msg/seat/x3eQxFjQFJLceae6l4_NgXnmsDY/>.

[RFC9261] Sullivan, N., "Exported Authenticators in TLS", RFC 9261, DOI 10.17487/RFC9261, July 2022, <https://www.rfc-editor.org/rfc/rfc9261>.

[RFC9266] Whited, S., "Channel Bindings for TLS 1.3", RFC 9266, DOI 10.17487/RFC9266, July 2022, <https://www.rfc-editor.org/rfc/rfc9266>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

[SEAT-Charter]
IETF, "Secure Evidence and Attestation Transport (SEAT): Charter for Working Group", <https://datatracker.ietf.org/wg/seat/about/>.

[SoK-Attestation]
Sardar, M. U., Fossati, T., and S. Frost, "SoK: Attestation in Confidential Computing", January 2023, <https://www.researchgate.net/publication/367284929_SoK_Attestation_in_Confidential_Computing>.

[Stunes-vTPM-CCC]
Mike Stunes, "Azure vTPM Attestation and Binding", July 2025, <https://www.youtube.com/watch?v=J7SibeZmQsE>.

[Tech-Concepts]
Sardar, M. U., "Perspicuity of Attestation Mechanisms in Confidential Computing: Technical Concepts", October 2025, <https://www.researchgate.net/publication/396199290_Perspicuity_of_Attestation_Mechanisms_in_Confidential_Computing_Technical_Concepts>.

Acknowledgments

We gratefully thank Peg Jones for review.

Contributors

Pavel Nikonorov (GENXT / IIAP NAS RA) contributed text in Section 4.1.1 and Section 5.2.1.

Author’s Address

Muhammad Usama Sardar
TU Dresden
Email: muhammad_usama.sardar@tu-dresden.de