SEAT Working Group
Internet-Draft
Intended status: Informational
Expires: 5 November 2026

I. Mihalcea
Arm
M. U. Sardar
TU Dresden
T. Fossati
Linaro
T. Reddy
Nokia
Y. Jiang

M. Chen
China Mobile
4 May 2026

Properties and Use Cases for Integrating Remote Attestation with Secure Channel Protocols
draft-mihalcea-seat-use-cases-02

Abstract

This document outlines desirable properties and use cases for integrating remote attestation (RA) capabilities with secure channel establishment protocols (e.g., TLS and DTLS). Peer authentication in such protocols establishes trust in a peer’s network identifiers but provides no assurance regarding the integrity of its underlying software and hardware stack. Remote attestation addresses this gap by enabling a peer to provide verifiable evidence about the current state of the Target Environment. This document specifies a set of essential properties the protocol solution must have, including cryptographic binding to the secure connection, evidence freshness, and flexibility to support different attestation models. It then explores relevant use cases, such as confidential data collaboration and secure secrets provisioning, to motivate the need for this integration. This document is intended to serve as an input to the design of protocol solutions within the SEAT working group.

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

1.1. Establishing Trust in Secure Communications

Secure channel protocols, such as Transport Layer Security (TLS), primarily establish trust in a peer's identity. This is typically achieved through mechanisms like a Public Key Infrastructure (PKI), where a trusted Certification Authority (CA) vouches for the binding between a public key and an identifier (e.g., a hostname).

However, this model has one key limitation: entity authentication provides no assurance about the peer's state, such as the integrity of its software stack at boot time and during runtime. A compromised endpoint, for instance, can still present a valid X.509 certificate and be considered "trusted" by a client. This gap allows compromised endpoints to maintain network access and the trust of their peers, posing a significant security risk in many environments.

1.2. The Role of Remote Attestation

Remote Attestation (RA), as described in the RATS architecture [RFC9334], is a mechanism designed to fill this gap. RA allows an entity (the "Attester") to produce verifiable "Evidence" about its current runtime state. This Evidence covers the Attester's TCB and can thus include measurements of:

* firmware
* operating system
* application code
* the configuration of its hardware and software security features (e.g., secure boot status and memory isolation).

A "Relying Party" can then use this Evidence, often with the help of a trusted "Verifier", to appraise the Attester's trustworthiness.

Composing RA with a secure channel establishment protocol adds a second dimension of trust - trustworthiness - to complement peer authentication. This allows a peer to make authorization decisions based not just on who the other party is, but also on what it is (e.g., an AMD SEV-SNP-based server running in some known datacenter) and whether its state is acceptable.

1.3. Purpose and Scope

The purpose of this document is to establish a set of essential properties for composition of RA with secure channel protocols and to outline the key use cases that can benefit from such a composition. Most of the use cases presented in this document are provided by industry contributors in the SEAT WG, who have plans to deploy this technology. The initial focus is on TLS 1.3 [I-D.ietf-tls-rfc8446bis] and its datagram-oriented variant, DTLS 1.3 [I-D.ietf-tls-rfc9147bis].

This document is intended as an input to the design of protocol solutions within the SEAT working group. It defines the "why" (the motivation) and the "what" (the requirements), but not the "how" (the protocol design itself). The "how" part is discussed in the companion document [I-D.usama-seat-intra-vs-post], which serves as the glue between this document and the protocol specifications. A key goal of this document is to define requirements for a solution that is agnostic to any specific attestation technology (e.g., Trusted Platform Modules (TPMs), Intel TDX, AMD SEV, Arm CCA).

2. Terminology

This document uses the terminology defined in the RATS Architecture [RFC9334], including "Attester", "Relying Party", "Verifier", "Evidence", and "Attestation Results".

This document also uses the following terms:

* Trusted Computing Base (TCB) of a device: see [RFC4949]. Note that for this draft, it includes respective configurations of hardware, firmware, and software.

* Confidential Workload: as defined in [I-D.draft-ccc-wimse-twi-extensions].

* Measurements: as defined in [I-D.draft-ietf-rats-eat-measured-component].

* AI agent: An AI agent is a software principal (typically long-running) that performs closed-loop "perceive -> plan -> act" cycles using an LLM or other model, and invokes external tools/APIs that may read sensitive data or change system/network state. Its configuration (e.g., model choice, tool enablement, prompt template) can change independently of the binary/image and usually more frequently than typical platform TCB updates [AI-agents].

3. Integration Properties

This section provides a list of desirable properties for designs that compose RA with secure channel protocols. Proposed protocol specifications should clearly state which of these properties are fulfilled and explain how.

3.1. Cryptographic Binding to Communication Channel

The Evidence or Attestation Result is cryptographically bound to the specific secure connection (e.g., the (D)TLS connection). This prevents *relay* attacks where an attacker presents valid, but unrelated Evidence from a different connection or context. This binding is paramount for all use cases because the absence of this binding can be exploited in high-severity vulnerabilities, such as [CVE-2026-33697].

3.2. Compound Authentication

RA should complement endpoint authentication rather than replace it. Combining the two security measures would ensure that the introduction of attestation increases security instead of replacing one security measure by another. A formal representation of this requirement in the form of _composition_ goal can be found in [ID-Crisis] for TLS 1.3 protocol.

3.3. Cryptographic Binding to Machine Identifier

Evidence should be cryptographically bound to the identifier provided to the machine by the infrastructure provider to prevent *diversion* attacks [ID-Crisis].

3.4. Attestation Credential Freshness

The Relying Party is able to verify that the Evidence or Attestation Result it receives was freshly generated by the Attester for the specific RA interaction. State is transient, and credentials from a previous RA interaction may no longer be valid. See Section 10 of [RFC9334] for more details about freshness in the context of RA. This is formalized for attestation nonce in [ID-Crisis].

3.5. Negotiation and Capability Discovery

Peers have a secure mechanism to discover each other's support for RA, the specific attestation formats they can produce or consume, and the attestation models they support. This enables interoperability and allows for graceful fallback for endpoints that do not support RA.

3.6. Attestation Model Flexibility

The solution supports both the Background Check and Passport models as defined in the RATS architecture [RFC9334]. The Background Check model is essential for use cases requiring maximum freshness, while the Passport model is better suited for performance, scalability, and scenarios where the Verifier may be offline or unreachable by the Relying Party.

3.7. Interaction with Peer Authentication

The solution supports using RA in conjunction with traditional PKI-based authentication (e.g., X.509 certificates). This provides two independent pillars of trust: endpoint trustworthiness (from RA) and identity (from PKI).

3.8. Runtime Attestation

Evidence collected at certificate issuance or during the initial secure channel establishment reflects only the Target Environment's state at that moment. It cannot guarantee that the Target Environment remains trustworthy for the lifetime of the certificate or even for the duration of the secure connection (e.g., the (D)TLS connection). As a result, such static Evidence is insufficient in environments where the Target Environment may change state after the connection is established and the connection is long-lived.

3.8.1. Periodic vs. On-demand Attestation

It should be possible for the Relying Party to request new Evidence periodically or on-demand during the lifetime of the connection. This may be necessary if the Target Environment has attributes that can change during the connection, thereby affecting its trustworthiness. Such changes cannot be detected using Evidence collected earlier. For example, the Evidence may include dynamic parameters such as runtime configuration flags (e.g., FIPS mode), which indicate whether the device has entered or exited an approved mode, or measurements of critical system files.

3.9. Privacy Preservation

The solution must not degrade the privacy of a standard secure connection (e.g., the (D)TLS connection). Evidence can contain highly specific, unique information about a device's hardware and software, which could be used as an advanced tracking mechanism, following a user across different connections and services. The design must consider how to minimize this leakage, especially when a third-party Verifier is involved in the protocol exchange.

3.10. Performance and Efficiency

The introduction of remote attestation should not add prohibitive latency or overhead to the connection establishment process. To be widely adopted, the solution must be practical. While some overhead is unavoidable, multiple additional round-trips or very large payloads in the initial handshake should be minimized.

4. Use Cases

This section provides the concrete motivation for the WG's work by describing specific use cases. For each case, the scenario, actors, and specific security guarantees needed from RA are described.

4.1. Secure Provisioning and High-Assurance Operations

Goal: Ensure the integrity of workloads and devices when bootstrapping their PKI-based identity or receiving critical commands.

4.1.1. Runtime Secret Provisioning

A confidential workload starts in a generic state and needs to fetch secrets (e.g., API keys, database credentials, encryption keys) to become operational.

* Requirement: The workload must attest its runtime state (TEE genuineness, software measurements) to a secrets management service. The service will only release the secrets after successful verification, ensuring they are delivered exclusively to a trustworthy environment. This use-case also covers secure device onboarding for IoT devices that lack a pre-provisioned PKI-based identity.

4.1.2. High-Assurance Command Execution

An operator sends a critical command to a remote system (e.g., an industrial controller, a financial transaction processor).

* Requirement: The system must provide fresh Evidence to the operator to prove its integrity before the command is dispatched. This prevents commands from being executed on a compromised system.

4.2. Confidential Data Collaboration

Goal: Enable multiple parties to collaborate on sensitive, combined datasets without exposing raw data to each other or to the infrastructure operator.

4.2.1. Data Clean Rooms

Multiple _data providers_ contribute sensitive data to a confidential workload for joint analysis. _Data consumers_ receive aggregated insights without ever accessing the raw, combined dataset.

* Requirement: Before sending data, each data provider must attest the confidential workload to verify it is running the authorized analysis code in a secure Trusted Execution Environment (TEE). Similarly, data consumers must attest the workload to trust the integrity of the results.

4.2.2. Secure Multi-Party Computation (MPC)

Distributed parties collaboratively compute a function (e.g., train a machine learning model) without sharing their local data.

* Requirement: The central aggregator, as well as each participating client, must be able to mutually attest to ensure all parties are running the correct, untampered MPC algorithm in a trusted environment.

4.3. Network Infrastructure Integrity

Goal: Verify the integrity of network devices that form the foundation of communication.

4.3.1. Attestation of Network Functions

A router, switch, or firewall joins a network’s management plane. A Virtualized Network Function (VNF) is instantiated on a generic server.

* Requirement: The network orchestrator must verify the device’s integrity (e.g., secure boot enabled, running signed OS and firmware) before allowing it to join the network and receive policy. This prevents a compromised router from misdirecting traffic or a malicious VNF from inspecting sensitive packets.

4.3.2. Securing Control and Management Planes

An administrator connects to a network device’s management interface.

* Requirement: The administrator’s client must verify the integrity of the management endpoint on the network device to ensure they are not connecting to a compromised interface that could steal credentials or manipulate the device.

4.4. Operation-Triggered Attestation for High-Impact Application Operations

Goal: Ensure the integrity of application services at operation time, when security posture may change after initial channel establishment.

Use case: *High-Assurance Operation Execution in Dynamic Application Services*: An application service instance (e.g., AI agent) or confidential computing environment (which could host an AI agent) maintains a (D)TLS connection with a peer and must execute a high-impact action (e.g., payment initiation, configuration change, privileged command).

* Requirement 1: Before executing a high-impact operation over the existing connection, the peer must present fresh, connection-bound Evidence reflecting the current behavior-affecting posture (e.g., enabled capabilities, policy configuration, runtime permissions).

* Requirement 2: The mechanism should support lightweight, dynamic attestation within the existing connection, without necessarily requiring a full new TLS handshake, so that behavior-affecting posture changes are visible to relying parties when required by local policy.

4.5. Attestation of Certificate Private Key

A TLS endpoint authenticates itself using an end-entity certificate whose corresponding private key is claimed to be protected by a secure element. While standard TLS authentication verifies possession of the private key, it provides no assurance about where or how that key is stored and used.

In this scenario, the peer acting as the Relying Party requires additional assurance that the private key associated with the end-entity certificate used to authenticate the TLS connection is generated, stored, and used within an attested cryptographic module. In addition to verifying possession of the private key via the TLS handshake, the Relying Party seeks Evidence that the key is non-exportable, remains bound to the cryptographic module, and that the module is operating in an expected security configuration at the time the TLS connection is established.

Remote attestation is used to provide Evidence about the cryptographic module where the private key used for TLS authentication is stored. The Evidence may include claims about the security properties of the cryptographic module. To prevent replay attacks, this Evidence has to be fresh and tied to the current TLS connection. Replayed Evidence could otherwise be used to falsely assert key protection properties that no longer hold.

* Requirement: The Attester must be able to produce Evidence that demonstrates that the private key used for secure channel authentication:

  - is generated and stored within a specific cryptographic module or secure element,

  - is protected against export or software extraction

  - is attested using fresh Evidence that is bound to the current TLS connection.

The Relying Party uses this Evidence, potentially with the assistance of a Verifier, to determine whether the key protection properties satisfy its local security policy.

The approach described in [I-D.draft-ietf-rats-pkix-key-attestation] addresses this use case partially by providing attestation of the cryptographic module and associated private key at certificate issuance time, reflecting their state when the certificate is enrolled. This model does not provide guarantees about the continued state of the module at connection establishment or during the lifetime of the TLS connection.

4.6. Platform-to-platform communication

Goal: Allow platforms to establish a trustworthy secure channel with each other.

Use case: Migration of workloads (confidential workloads in particular) between different platforms. Migration is occasionally required in order to maintain uptime for the hosted services across periods of scheduled downtime for the hosting platform. Having remote attestation-enforced policies for such migration events provides guarantees that the services will not be exposed to lower security guarantees when migrating. Migration is typically performed by trusted, low-level components (migration agents) on both source and destination platforms, which perform the authorization checks and handle the data migration.

* Requirement: The migration agent on the destination platform typically acts as Attester, proving its state for its peer on the source platform (where the workload initially resides).

* Example: Intel TDX offers migration capabilities via its Migration Trust Domain (MigTD) [MigTD]. Peer MigTDs on the initiating and target platforms set up an attested TLS connection to perform the migration over.

4.7. AI Governance and Accountability

Goal: Design framework for governing autonomous AI agents.

Use case: See [I-D.aylward-aiga-2] for details. Contrary to Section 4.4, the entity verifying the Evidence in this case is the governance body and for the purposes of ensuring that no unethical or harmful action is performed.

* Requirement: Runtime attestation based on agent risk tiers defined in Section 2.2 of [I-D.aylward-aiga-2]

5. Security Considerations

This document describes use cases and integration properties. The security of any protocol designed to fulfill these properties will depend on its specific mechanisms. However, any solution must address the following high-level considerations:

* Replay and Relay Protection: The requirements for cryptographic binding and freshness are critical. Failure to bind attestation credentials tightly to the current connection would allow an adversary to replay or relay old or stolen, yet valid credentials from a compromised system, completely undermining the security goals.

* Verifier Trust and Privacy: In the Background Check model, the Relying Party communicates with a Verifier. This reveals to the Verifier that the Relying Party is communicating with the Attester. Depending on the scenario, this could leak sensitive information about business relationships or user activity. Solutions should consider mechanisms to minimize the data revealed to the Verifier.

* Downgrade Attacks: The negotiation of attestation capabilities must be secure. An active attacker must not be able to trick two parties that both support attestation into negotiating a connection without it.

* Evidence Semantics: This document does not define attestation appraisal policies. However, a Relying Party must be careful when interpreting Attestation Results. A "valid" attestation only means the Evidence is authentic and correctly signed; it does not automatically mean the underlying system is "secure". The Relying Party must have a clear policy for what measurements, software versions, and security configurations are acceptable.

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[AI-agents]
Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., and A. Narayanan, "AI agents that matter", July 2024, <https://arxiv.org/abs/2407.01502>.

[CVE-2026-33697]
"CVE-2026-33697", n.d., <https://www.cve.org/CVERecord?id=CVE-2026-33697>.

[I-D.aylward-aiga-2]
Aylward, E. R., "AI Governance and Accountability Protocol (AIGA)", Work in Progress, Internet-Draft, draft-aylward-aiga-2-00, 26 January 2026, <https://datatracker.ietf.org/doc/html/draft-aylward-aiga-2-00>.

[I-D.draft-ccc-wimse-twi-extensions]
Novak, M., Deshpande, Y., and H. Birkholz, "WIMSE Extensions for Trustworthy Workload Identity", Work in Progress, Internet-Draft, draft-ccc-wimse-twi-extensions-01, 5 January 2026, <https://datatracker.ietf.org/doc/html/draft-ccc-wimse-twi-extensions-01>.

[I-D.draft-ietf-rats-eat-measured-component]
Frost, S., Fossati, T., Tschofenig, H., and H. Birkholz, "Entity Attestation Token (EAT) Measured Component", Work in Progress, Internet-Draft, draft-ietf-rats-eat-measured-component-12, 20 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-measured-component-12>.

[I-D.draft-ietf-rats-pkix-key-attestation]
Ounsworth, M., Fiset, J., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Evidence Encoding for Hardware Security Modules", Work in Progress, Internet-Draft, draft-ietf-rats-pkix-key-attestation-05, 17 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-pkix-key-attestation-05>.

[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.ietf-tls-rfc9147bis]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-rfc9147bis-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc9147bis-01>.

[I-D.usama-seat-intra-vs-post]
Sardar, M. U., "Pre-, Intra- and Post-handshake Attestation", Work in Progress, Internet-Draft, draft-usama-seat-intra-vs-post-03, 22 January 2026, <https://datatracker.ietf.org/doc/html/draft-usama-seat-intra-vs-post-03>.

[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>.

[MigTD] "Intel TDX Migration TD", n.d., <https://github.com/intel/MigTD>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/rfc/rfc4949>.

[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>.

Acknowledgments

TODO

Authors' Addresses

Ionu Mihalcea
Arm
Email: ionut.mihalcea@arm.com

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

Thomas Fossati
Linaro
Email: thomas.fossati@linaro.org

Tirumaleswar Reddy
Nokia
Email: kondtir@gmail.com

Yuning Jiang
Email: jiangyuning2@h-partners.com

Meiling Chen
China Mobile
Email: chenmeiling@chinamobile.com