Remote ATtestation ProcedureS
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

L. Xia
W. Jiang
Huawei Technologies
M. U. Sardar
TU Dresden
H. Birkholz
Fraunhofer SIT
J. Zhang
H. Labiod
Huawei Technologies France S.A.S.U.

7 July 2025

Integration of Remote Attestation with Key Negotiation and Distribution
draft-xia-rats-key-negotiation-integration-00

Abstract

This draft proposes a lightweight security enhancement scheme based on remote attestation—key negotiation integrated into remote attestation. Organically integrating the main steps of end-to-end key negotiation into the remote attestation process may provide more security and flexibility.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-xia-rats-key-negotiation-integration/.

Discussion of this document takes place on the Remote ATtestation ProcedureS Working Group mailing list (mailto:rats@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/rats/. Subscribe at https://www.ietf.org/mailman/listinfo/rats/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-rats/draft-xia-rats-key-negotiation-integration.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 8 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Remote attestation is a security mechanism based on trusted hardware (e.g., TPM, TEE), allowing remote verifiers to cryptographically verify the integrity of the target device's software configuration, hardware state, and runtime environment. Hence, remote attestation can effectively prove the overall security state of the endpoint. Secure channel protocols (e.g., TLS, QUIC, IPSec, SSH) establish end-to-end (E2E) secure channels based on the authentication of the endpoint's legitimate identity and secure key negotiation, ensuring the security of network communication. By organically combining remote attestation protocols with secure channel protocols and establishing cryptographic binding between them, it is possible to achieve a logical binding of endpoint security and network security, ensuring dual verification and protection of the identity and state of the endpoint in secure connections. Attested TLS [I-D.fossati-tls-attestation] [I-D.fossati-tls-exported-attestation] is currently an important related work in the industry, and other similar works include binding remote attestation with credential issuance (e.g., certificates [I-D.ietf-lamps-csr-attestation], OAuth tokens, etc.) to achieve security enhancement.

However, in some scenarios, the above binding may not be possible. For example:

* Scenario 1: When tenants in a public cloud/compute cluster deploy workloads, they need to first verify their runtime environment's security through remote attestation before requesting Key Management Service (KMS) to assign application-layer data keys to the Virtual Machine (VM)/compute node where the workload resides. At this point, these keys are used for application-layer data encryption, such as file encryption or disk encryption, and are not used for any secure channel protocols. For example, to protect model parameters from leakage and tampering, model weights and other parameters need to be encrypted before model loading and transmitted to a Trusted Execution Environment (TEE). At this time, it is necessary to ensure that only the TEE has the key and decrypts it.

* Scenario 2: The end user/client accesses the online TEE computing environment, submits his data for business processing or large model inference, and needs to ensure the security of the entire computing environment through remote attestation before establishing a secure connection. At this point, there are multiple options for the secure channel protocol that can be established, such as TLS, IPSec, QUIC, OHTTP, etc. The user may only need to complete E2E key negotiation based on remote attestation. As for which secure protocol or application layer encryption the negotiated key is used for, and how it is used, there can be various implementation methods.

In summary, considering the diversity of remote attestation application scenarios and the limitations or complexity of combining with security protocols, this draft proposes a lightweight security enhancement scheme based on remote attestation—key negotiation integrated into remote attestation. By organically integrating the key steps of E2E key negotiation into the remote attestation process, the following can be achieved:

* The key distribution of KMS or E2E key negotiation can be automatically completed based on remote attestation, improving the security of key negotiation;

* The keys negotiated automatically can be flexibly applied in various ways, whether for secure protocols or application layer encryption;

* Compared to the complete and systematic implementation of Attested TLS, a more lightweight implementation can be provided.

1.1. Requirements Notation

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.

2. Integration Scheme

The current specification is based on passport model of RATS. Future versions of specification will include the background-check model. We present two integration schemes:

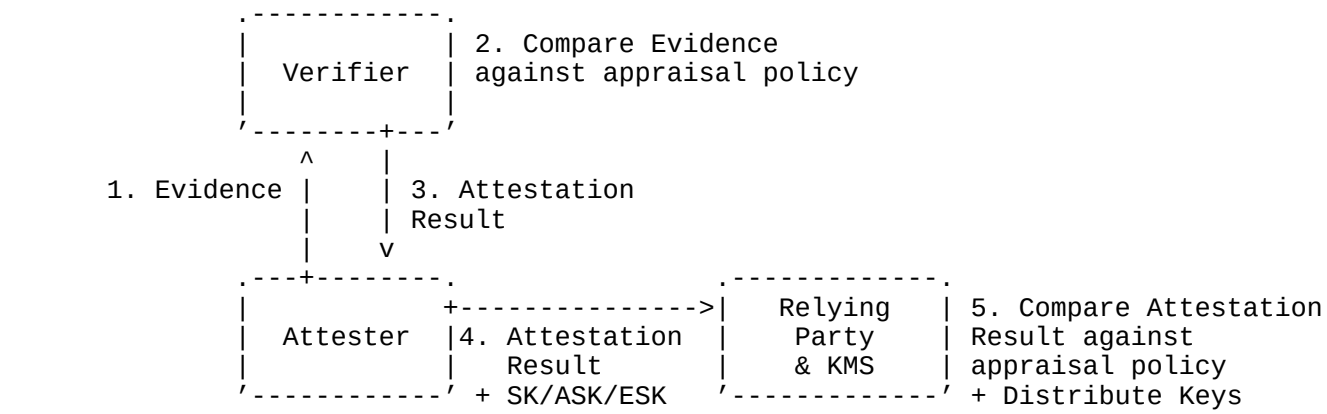
2.1. Key Distribution Based on KMS Integrated with Remote Attestation

The KMS mechanism on public cloud networks includes the root of trust, secure channels between Attester and KMS, full lifecycle management of keys (including key generation, storage, rotation, and destruction), hierarchical encryption architecture (such as Envelope Encryption), and access control mechanisms. There are two cases here:

* Attester generates keys to be distributed to other applications via KMS.

* Attester obtains keys from the KMS.

So KMS enables applications to generate/obtain their own application-layer encryption symmetric/asymmetric keys and distribute these keys between the required applications. Furthermore, the method of integrating key distribution into the remote attestation interaction process is shown in the following diagram:



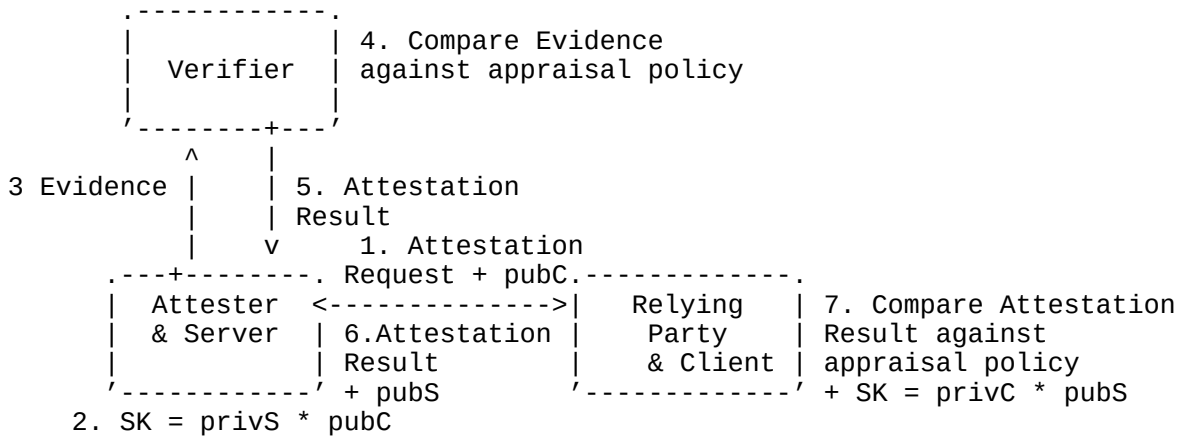
SK : Symmetric Key
ASK : Asymmetric Key
ESK : Encrypted Symmetric Key

Figure 1: Key Distribution Based on KMS Integrated with Remote Attestation

In the standard remote attestation process described above, the Attester can generate and provide/request the Attester's application layer SK/ASK/ESK keys in the result returned to the KMS. SK MUST be conveyed over a secure channel. The KMS can generate or distribute these keys accordingly. During key rotation, the KMS can proactively trigger the above process to complete the update and rotation of the new and old keys.

2.2. Integrating E2E Key Negotiation Into Remote Attestation

The current main implementation mechanism for E2E key negotiation is DHE and ECDHE. Taking ECDHE as an example, the method of integrating it into remote attestation is shown in the following figure:



SK : Symmetric Key
(privC,pubC): Client's ECDHE key pair
(privS,pubS): Server's ECDHE key pair

Figure 2: Integrating E2E Key Negotiation Into Remote Attestation

In the standard remote attestation process described above, the Client includes the public key pubC from its dynamically generated ECDHE key pair (privC, pubC) in the remote attestation request message, while retaining its private key privC. Upon receiving pubC, the Server can compute the symmetric key SK using its private key privS from its dynamically generated ECDHE key pair (privS, pubS). After completing the remote attestation with the Verifier, the Server includes its pubS in the Attestation Result returned to the Client. Once the Client verifies the Attestation Result, it can compute the symmetric key SK using pubS and its own private key privC, thereby completing both the remote attestation and ECDHE key agreement.

3. Use of Negotiated Keys in Different Security Protocols

Through the above process, key negotiation/distribution is completed during the remote attestation process, and is synchronized based on the results of the remote attestation. If the negotiated key is used for application layer encryption, its specific usage is strongly related to the application and can be very flexible. When the key is used for the security protocols, such as TLS, IPSec, etc., there are at least the following binding methods:

* TLS: The negotiated key can be used as a pre-shared key for subsequent TLS handshakes; the key can also be used as an externally imported shared key to participate in TLS Hybrid key exchange [I-D.ietf-tls-hybrid-design];

* IPSec: The negotiated key can be used as a pre-shared key for subsequent IKEv2 handshakes; or the key can be directly used as a session key for data plane encryption and integrity protection in IPSec ESP; or the key can also be used as Post-quantum Preshared Keys (PPKs) [RFC8784] to achieve binding with the IPSec protocol.

4. Remote Attestation Protocol and Message Extensions

This section describes how to extend RATS protocol and message to incorporate key negotiation into the remote attestation process.

TBD

5. Security Considerations

Risk of relay attacks needs to be evaluated in the design.

TBD

6. Privacy Considerations

TBD

7. IANA Considerations

TBD

8. References

8.1. 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>.

[RFC8784] Fluhrer, S., Kampanakis, P., McGrew, D., and V. Smyslov, "Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security", RFC 8784, DOI 10.17487/RFC8784, June 2020, <https://www.rfc-editor.org/rfc/rfc8784>.

8.2. Informative References

[I-D.fossati-tls-attestation] Tschofenig, H., Sheffer, Y., Howard, P., Mihalcea, I., Deshpande, Y., Niemi, A., and T. Fossati, "Using Attestation in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, draft-fossati-tls-attestation-09, 30 April 2025, <https://datatracker.ietf.org/doc/html/draft-fossati-tls-attestation-09>.

[I-D.fossati-tls-exported-attestation] Fossati, T., Sardar, M. U., Reddy.K, T., Sheffer, Y., Tschofenig, H., and I. Mihalcea, "Remote Attestation with Exported Authenticators", Work in Progress, Internet-Draft, draft-fossati-tls-exported-attestation-02, 3 July 2025, <https://datatracker.ietf.org/doc/html/draft-fossati-tls-exported-attestation-02>.

[I-D.ietf-lamps-csr-attestation] Ounsworth, M., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Use of Remote Attestation with Certification Signing Requests", Work in Progress, Internet-Draft, draft-ietf-lamps-csr-attestation-19, 25 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-csr-attestation-19>.

[I-D.ietf-tls-hybrid-design] Stebila, D., Fluhrer, S., and S. Gueron, "Hybrid key exchange in TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-hybrid-design-13, 17 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-13>.

Authors' Addresses

Liang Xia
Huawei Technologies
Email: frank.xialiang@huawei.com

Weiyu Jiang
Huawei Technologies
Email: jiangweiyu1@huawei.com

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

Henk Birkholz
Fraunhofer SIT
Email: henk.birkholz@ietf.contact

Jun Zhang
Huawei Technologies France S.A.S.U.
Email: junzhang1@huawei.com

Houda Labiod
Huawei Technologies France S.A.S.U.
Email: houda.labiod@huawei.com