

Integration of Remote Attestation with Key Negotiation and Distribution

draft-xia-rats-key-negotiation-integration-01

<https://datatracker.ietf.org/doc/draft-xia-rats-key-negotiation-integration/>

Liang Xia, Weiyu Jiang

Huawei

Muhammad Usama Sardar

TU Dresden

Henk Birkholz

Fraunhofer SIT

Jun Zhang, Houda Labiod

Huawei Technologies France S.A.S.U.

Updates from -00 to -01

- Add 1 use cases (LLM deployment in remote cloud or Data Center)
- Add key binding mechanism to prevent diversion attack
- Improve the description of introduce the use of negotiated keys in TLS and IPsec
- Add a open source project introduction: Trustee
- Add Security Consideration contents
- Editorial revisions

Motivation & Goal

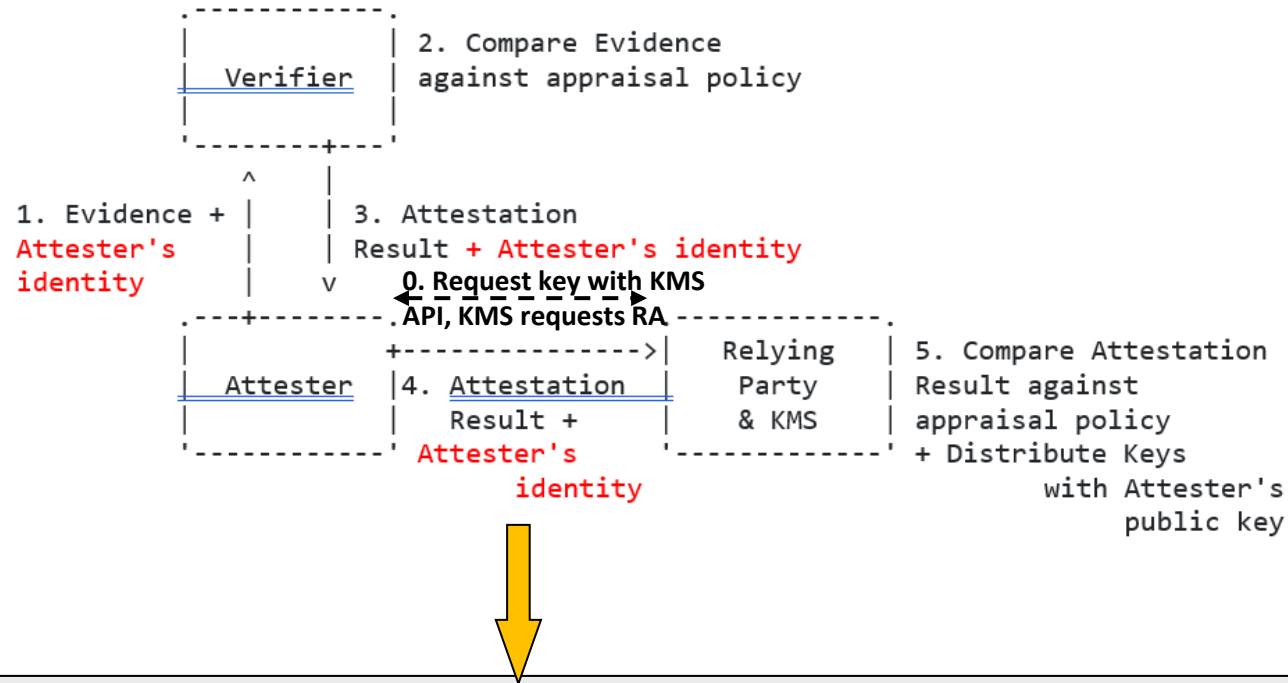
- **Goal:** Combine key management mechanisms (key distribution, key negotiation) and remote attestation.
- **Key idea:** Provide a binding between RATS messages with Attester's identity (raw public key or certificate) to prevent diversion attack.
- **Advantages:**
 - ✓ Enhance security
 - ✓ Enhance performance
 - ✓ Provide flexibility
- **Comparison with Attested TLS:** More lightweight implementation.

Use Cases

- **Use case 1 – Public cloud / computing network KMS (Key Management Service)**
 - KMS distribute application-layer keys (such as: encryption symmetric/asymmetric public keys, authentication public keys, decrypted plaintext and calculated signatures)
 - After remote attestation, these keys are used **for encryption/authentication** (ex: file, disk, LLM parameters (weight...)). There is no secure channel protocol.
- **Use case 2 – End user' s data access**
 - The end user sends his sensitive data to TEE for LLM inference.
 - Various security protocols can be used like TLS, IPsec, QUIC, OHTTP.
 - The user only requires basic end-to-end key negotiation along with remote attestation, which can then be utilized for **specific secure channel protocols**.
- **Use case 3 – Large language model (or file system) deployment in remote cloud or Data Center**
 - Enterprise deploys/uploads its private large language model (or file system).
 - The enterprise first uploads the encrypted objects to the cloud platform.
 - The enterprise conducts a remote attestation on the cloud platform.
 - After RA process, the decryption key is sent to **decrypt the uploaded objects**.

Integration Scheme 1 – Public Cloud KMS

Key Distribution Integrated with Remote Attestation

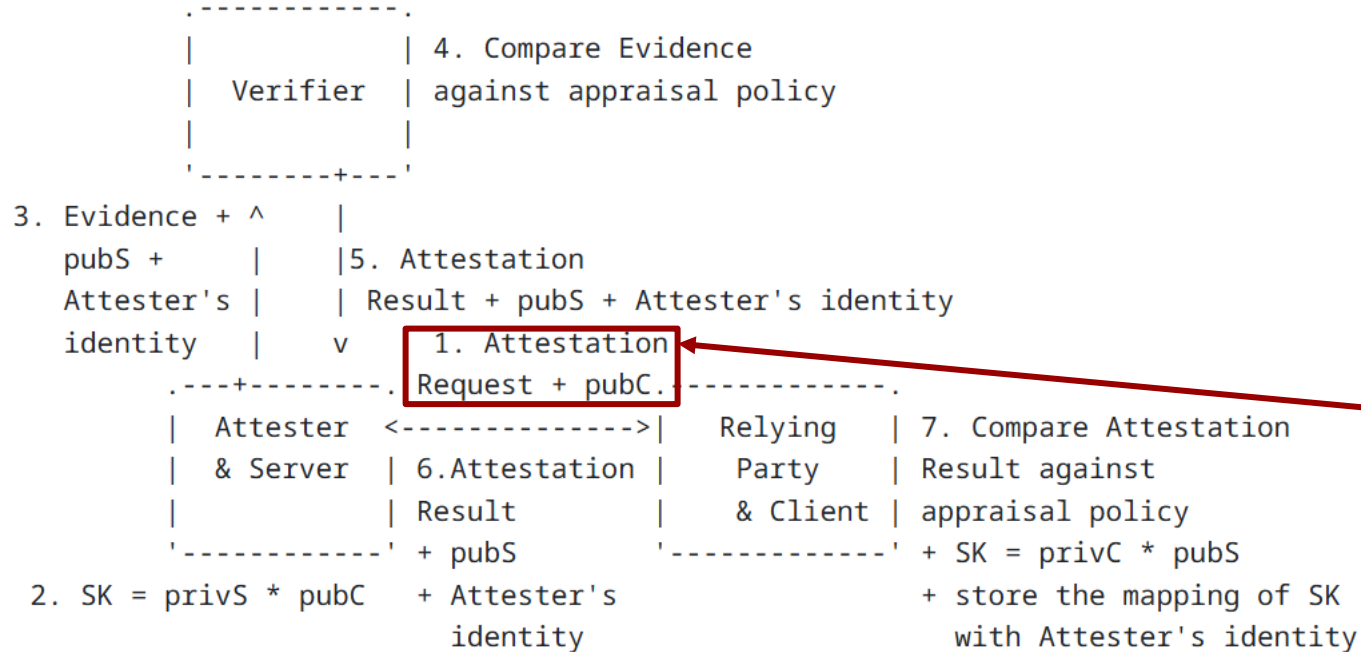


Background check model **requires no extension to RATS protocol and messages.**

- By including the **attester's identity (raw public key or certificate)** in the messages throughout the remote attestation process and **having the attester and verifier endorse and sign it**, a **key binding mechanism** between the attester's attestation result and its identity is implemented.
- **The KMS can use the identity's public key for key distribution**, ensuring that the keys are distributed to the correct attester, thereby **eliminating the risk of diversion attack.**

Integration Scheme 2

Integrating E2E Key Negotiation Into Remote Attestation



- **Passport model** used
- Background check model will be added later.

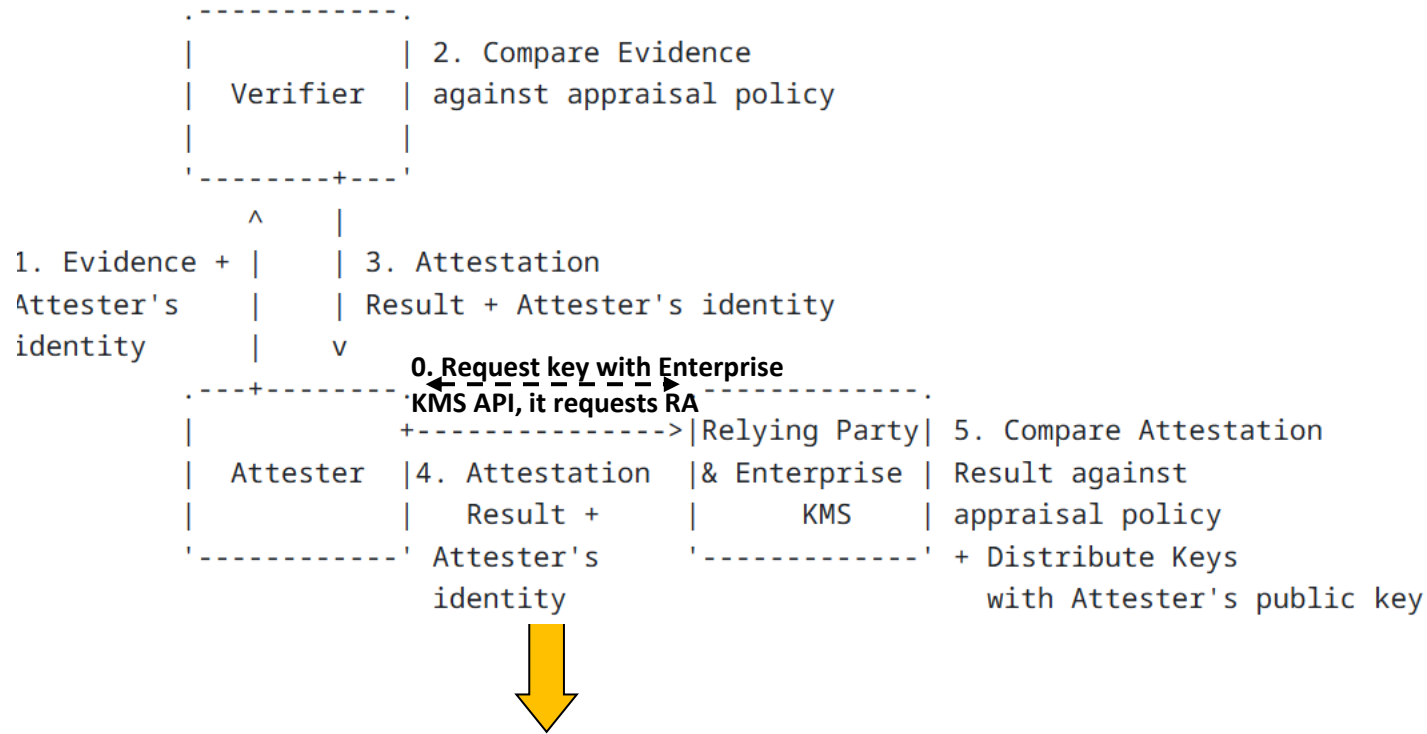
Suggest RATS to define attestation request



- Attester (server) and Relying Party (client) perform **ECDHE key negotiation** within the RA process.
- by **including the Attester's identity (raw public key or certificate)** in all remote attestation messages **having the Attester and Verifier endorse and sign it, a key binding mechanism** (recording and checking the mapping of attester identity with negotiated SK) between the Attester's attestation result and its identity is implemented, thereby **eliminating the risk of diversion attack.**

Integration Scheme 3

Enterprise Customer Key Distribution to Public Cloud within Remote Attestation



Background check model requires no extension to RATS protocol and messages.

- This scenario is similar to the public cloud KMS key distribution scenario, except that the public cloud is no longer responsible for key distribution. **Instead, the enterprise manages the keys itself and completes the key distribution after passing the remote attestation.**
- The same **key binding mechanism** is implemented, thereby **eliminating the risk of diversion attack.**

Next Steps

- To be added:
 - Remote Attestation Protocol details and Message Extensions
 - Background check model for use case 2
 - Security and privacy considerations
 - Others: Relationship with attested EDHOC, ...
- Question to WG:
 - Is this work in scope of RATS?
 - Is RATS WG interested?
- Comments are warmly welcome 😊

Thanks

Background: Introduce the Use of Negotiated Keys in Different Security Protocols

- TLS:
 - ✓ The negotiated key can be used as a **pre-shared key** for subsequent TLS handshakes, or as an **externally imported shared key for TLS hybrid key exchange** (<https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/>).
 - ✓ Integrating all these pre-shared keys into the TLS protocol can be achieved through the **hybrid authentication mode of TLS_CERT_WITH_EXTERN_PSK**.
- IPsec:
 - ✓ The negotiated key can be used as a **pre-shared key** for subsequent IKEv2 handshakes, or as **Post-quantum Preshared Keys (PPKs) [RFC8784]** to be used in the IPsec protocol.
 - ✓ Similar to TLS, the **IKEv2 hybrid authentication mode** is used.
- The pre-negotiated key achieves **2 advantages**:
 - ✓ **Binding** RA with TLS/IPsec.
 - ✓ **mutual authentication** between both sides of TLS/IPsec .

Background: Implementation Status – Trustee

- Responsible Organization: Trustee (open source project within the Confidential Containers).
- Location: <https://github.com/confidential-containers/trustee>
- Description:

Trustee contains tools and components for attesting confidential guests and providing secrets to them.

Trustee components include:

- **Key Broker Service:** The KBS is a server that facilitates remote attestation and secret delivery. Its role is similar to that of the **Relying Party** in the RATS model;
- **Attestation Service:** The AS verifies TEE evidence. In the RATS model this is a **Verifier**;
- **Reference Value Provider Service:** The RVPS manages reference values used to verify TEE evidence;
- **KBS Client Tool:** This is a simple tool which can be used to test or configure the KBS and AS.

There are two main ways to deploy Trustee: with Docker Compose, on Kubernetes.

- **Level of Maturity:** This is a proof-of-concept prototype implementation.
- **License:** Apache-2.0.
- **Coverage:** This implementation covers most of the aspects of the use case 1 and 3 of this draft.
- **Contact:** Ding Ma, xynnn@linux.alibaba.com