RATS Working Group H. Birkholz
Internet-Draft M. Eckel
Intended status: Informational Fraunhofer SIT
Expires: January 9, 2020 July 08, 2019

Reference Interaction Model for Challenge-Response-based Remote Attestation
draft-birkholz-rats-reference-interaction-model-01

Abstract

This document defines an interaction model for a basic remote attestation procedure. Additionally, the required information elements are illustrated.

Status of This Memo

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Table of Contents

1. Introduction

Remote attestation procedures (RATS) are a combination of activities, in which a Verifier creates assertions about assertions of integrity and about characteristics of other system entities by the appraisal of corresponding signed assertions (evidence). In this document, a reference interaction model for a generic challenge-response-based remote attestation procedure is provided. The minimum set of components, roles and information elements that have to be conveyed between Verifier and Attester are defined as a standard reference to derive more complex RATS from.

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. Disambiguation

The term "Remote Attestation" is a common expression and often associated with certain properties. The term "Remote" in this context does not necessarily refer to a remote system entity in the scope of network topologies or the Internet. It rather refers to a decoupled system or different computing context, which also could be present locally as components of a composite device. Examples include: a Trusted Execution Environment (TEE), Baseboard Management Controllers (BMCs), as well as other physical or logical protected/ isolated execution environments.

3. Scope

This document focuses on a generic interaction model between Verifiers and Attesters. Complementary processes, functions and activities that are required for a complete semantic binding of RATS are not in scope. Examples include: identity establishment, key enrollment, and certificate revocation. Furthermore, any processes and activities that go beyond carrying out the remote attestation process are out of scope. For instance, using the result of a remote attestation that is emitted by the Verifier, such as triggering remediation actions and recovery processes, as well as the remediation actions and recovery processes themselves, are out of scope.

4. Component Roles

The Reference Interaction Model for Challenge-Response-based Remote Attestation is based on the standard roles defined in [I-D.birkholz-rats-architecture]:

Attester: The role that designates the subject of the remote attestation. A system entity that is the provider of evidence takes on the role of an Attester.

Verifier: The role that designates the system entity and that is the appraiser of evidence provided by the Attester. A system entity that is the consumer of evidence takes on the role of a Verifier.

5. Prerequisites

Attester Identity:

Attestation Authenticity: An Attestation MUST be authentic.

An attestation, in order to be authentic, MAY This Identity MUST be part of the signed assertions (attestation evidence) that the

Attester conveys to the Verifier. An Identity MAY be a unique identity or it MAY be included in a zero-knowledge proof (ZKP) or be part of a group signature.

Authentication Secret: An Authentication Secret MUST be present on the Attester. The Attester MUST sign assertions with that Authentication Secret, proving the authenticity of the assertions. The Authentication Secret MUST be established before a remote attestation procedure can take place. How it is established is out of scope for this reference model.

6. Remote Attestation Interaction Model

This section defines the information elements that have to be conveyed via a protocol, enabling the conveyance of Evidence between Verifier and Attester, as well as the interaction model for a generic challenge-response remote attestation scheme.

6.1. Information Elements

Attester Identity ('attesterIdentity'): _mandatory_

A statement about a distinguishable Attester made by an entity without accompanying evidence of its validity, used as proof of identity.

Authentication Secret ID ('authSecID'): _mandatory_

An identifier that MUST be associated with the Authentication Secret which is used to sign evidence.

Nonce ('nonce'): _mandatory_

The Nonce (number used once) is intended to be unique and practically infeasible to guess. In this reference interaction model the Nonce MUST be provided by the Verifier and MUST be used as proof of freshness. With respect to conveyed evidence, it ensures the result of an attestation activity to be created recently, e. g. sent or derived by the challenge from the Verifier. As such, the Nonce MUST be part of the signed Attestation Evidence that is sent from the Attester to the Verifier.

Assertions ('assertions'): _mandatory_

Assertions represent characteristics of an Attester. They are required for proving the integrity of an Attester. Examples are assertions about sensor data, policies that are active on the

system entity, versions of composite firmware of a platform, running software, routing tables, or information about a local time source.

Reference Assertions ('refAssertions') _mandatory_

Reference Assertions are used to verify the assertions received from an Attester in an attestation verification process. For example, Reference Assertions MAY be Reference Integrity Measurements (RIMs) or assertions that are implicitly trusted because they are signed by a trusted authority. RIMs represent (trusted) assertions about the intended platform operational state of the Attester.

Assertion Selection ('assertionSelection'): _optional_

An Attester MAY provide a selection of assertions in order to reduce or increase retrieved assertions to those that are relevant to the conducted appraisal. Usually, all available assertions that are available to the Attester SHOULD be conveyed. The Assertion Selection MAY be composed as complementary signed assertions or MAY be encapsulated assertions in the signed Attestation Evidence. An Attester MAY decide whether or not to provide all requested assertions or not. An example for an Assertion Selection is a Verifier requesting (signed) RIMs from an Attester.

(Signed) Attestation Evidence ('signedAttestationEvidence'): _mandat ory_

Attestation Evidence consists of the Authentication Secret ID that identifies an Authentication Secret, the Attester Identity, the Assertions, and the Verifier-provided Nonce. Attestation Evidence MUST cryptographically bind all of those elements. The Attestation Evidence MUST be signed by the Authentication Secret. The Authentication Secret MUST be trusted by the Verifier as authoritative.

Attestation Result ('attestationResult'): _mandatory_

An Attestation Result is produced by the Verifier as a result of a Verification of Attestation Evidence. The Attestation Result represents assertions about integrity and other characteristics of the corresponding Attester.

6.2. Interaction Model

The following sequence diagram illustrates the reference remote attestation procedure defined by this document.

```
[Attester]                                                      [Verifier]
    |                                                               |
    | <--- requestAttestation(nonce, authSecID, assertionSelection) |
    |                                                               |
collectAssertions(assertionSelection)                               |
    | => assertions                                                     |
    |                                                               |
signAttestationEvidence(authSecID, assertions, nonce)               |
    | => signedAttestationEvidence                                  |
    |                                                               |
    | signedAttestationEvidence ----------------------------------> |
    |                                                               |
    | verifyAttestationEvidence(signedAttestationEvidence, refAssertions)
    |                                          attestationResult <= |
    |                                                               |
```

The remote attestation procedure is initiated by the Verifier, sending an attestation request to the Attester. The attestation request consists of a Nonce, a Authentication Secret ID, and an Assertion Selection. The Nonce guarantees attestation freshness. The Authentication Secret ID selects the secret with which the Attester is requested to sign the Attestation Evidence. The Assertions Selection narrows down or increases the amount of received Assertions, if required. If the Assertions Selection is empty, then by default all assertions that are available on the system of the Attester SHOULD be signed and returned as Attestation Evidence. For example, a Verifier may only be interested in particular information about the Attester, such as proof of with which BIOS and firmware it booted up, and not include information about all currently running software.

The Attester, after receiving the attestation request, collects the corresponding Assertions to compose the Attestation Evidence that the Verifier requested--or, in case the Verifier did not provide an Assertions Selection, the Attester collects all information that can be used as complementary Assertions in the scope of the semantics of the remote attestation procedure. After that, the Attester produces Attestation Evidence by signing the Attester Identity, the Assertions, and the Nonce with the Authentication Secret identified by the Authentication Secret ID. Then the Attester sends the signed Attestation Evidence back to the Verifier.

Important at this point is that Assertions, the Nonce as well as the Attester Identity information MUST be cryptographically bound to the signature of the Attestation Evidence. It is not required for them to be present in plain text, though. Cryptographic blinding MAY be used at this point. For further reference see Security and Privacy Considerations (Section 8)

As soon as the Verifier receives the signed Attestation Evidence, it verifies the signature, the Attester Identity, the Nonce, and the Assertions. This process is application-specific and can be carried out by, e. g., comparing the Assertions to known (good), expected Reference Assertions, such as Reference Integrity Measurements (RIMs), or evaluating it in other ways. The final output of the Verifier is the Attestation Result. It constitutes an new assertion about properties and characteristics of the Attester, i. e. whether or not it is compliant to policies, or even can be "trusted".

7. Further Context

Depending on the use cases to cover, there may be additional requirements. Some of them are mentioned in this section.

7.1. Confidentiality

Confidentiality of exchanged attestation information may be desirable. This requirement usually is present when communication takes place over insecure channels, such as the public Internet. In such cases, TLS may be uses as a suitable communication protocol that preserves confidentiality. In private networks, such as carrier management networks, it must be evaluated whether or not the transport medium is considered confidential.

7.2. Mutual Authentication

In particular use cases mutual authentication may be desirable in such a way that a Verifier also needs to prove its identity to the Attester, instead of only the Attester proving its identity to the Verifier.

7.3. Hardware-Enforcement/Support

Depending on the requirements, hardware support for secure storage of cryptographic keys, crypto accelerators, or protected or isolated execution environments may be useful. Well-known technologies are Hardware Security Modules (HSM), Physically Unclonable Functions (PUFs), Shielded Secrets, and Trusted Executions Environments (TEEs).

8. Security and Privacy Considerations

In a remote attestation process the Verifier or the Attester MAY want to cryptographically blind several attributes. For instance, information can be part of the signature after applying a one-way function (e. g. a hash function).

There is also a possibility to scramble the Nonce or Attester Identity with other information that is known to both the Verifier and Attester. A prominent example is the IP address of the Attester that usually is known by the Attester itself as well as the Verifier. This extra information can be used to scramble the Nonce in order to counter certain types of relay attacks.

9. Acknowledgments

Very likely.

10. Change Log

- Initial draft -00

- Changes from version 00 to version 01:

  - Added details to the flow diagram

- Changes from version 01 to version 02:

  - Integrated comments from Ned Smith (Intel)

  - Reorganized sections and

  - Updated interaction model

- Changes from version 02 to version 03:

  - Replaced "claims" with "assertions"

  - Added proof-of-concept CDDL for CBOR via CoAP based on a TPM 2.0 quote operation

11. References

11.1. 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/info/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/info/rfc8174>.

11.2. Informative References

[I-D.birkholz-rats-architecture]
Birkholz, H., Wiseman, M., Tschofenig, H., and N. Smith, "Architecture and Reference Terminology for Remote Attestation Procedures", draft-birkholz-rats-architecture-01 (work in progress), March 2019.

Appendix A. CDDL Specification for a simple CoAP Challenge/Response Interaction

The following CDDL specification is an examplary proof-of-concept to illustrate a potential implementation of the Reference Interaction Model. The transfer protocol used is CoAP using the FETCH operation. The actual resource operated on can be empty. Both the Challenge Message and the Response Message are exchanged via the FETCH Request and FETCH Response body.

In this example, the root-of-trust for reporting primitive operation "quote" is provided by a TPM 2.0.

```
RAIM-Bodies = CoAP-FETCH-Body / CoAP-FETCH-Response-Body

CoAP-FETCH-Body = [ hello: bool, ; if true, the AK-Cert is conveyed
                    nonce: bytes,
                    pcr-selection: [ + [ tcg-hash-alg-id: uint .size 2, ;
TPM2_ALG_ID
                                         [ + pcr: uint .size 1 ],
                                       ]
                                   ],
                  ]

CoAP-FETCH-Response-Body = [ attestation-evidence: TPMS_ATTEST-quote,
                             tpm-native-signature: bytes,
                             ? ak-cert: bytes, ; attestation key certificate
                           ]

TPMS_ATTEST-quote = [ qualifiediSigner: uint .size 2, ;TPM2B_NAME
                      TPMS_CLOCK_INFO,
                      firmwareVersion: uint .size 8
                      quote-responses: [ * [ pcr: uint .size 1,
                                             + [ pcr-value: bytes,
                                                 ? hash-alg-id: uint .size 2,
                                               ],
                                           ],
                                         ? pcr-digest: bytes,
                                       ],
                    ]

TPMS_CLOCK_INFO = [ clock: uint .size 8,
                    resetCounter: uint .size 4,
                    restartCounter: uint .size 4,
                    save: bool,
                  ]
```

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Michael Eckel
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: michael.eckel@sit.fraunhofer.de