RATS Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 10, 2020

H. Birkholz
M. Eckel
Fraunhofer SIT

January 07, 2020

Reference Interaction Models for Remote Attestation Procedures
draft-birkholz-rats-reference-interaction-model-02

Abstract

This document defines interaction models for basic remote attestation procedures. Different methods of conveying attestation evidence securely are defined and illustrated. Analogously, the required information elements used by conveyance protocols are defined and illustrated.

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/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 10, 2020.

Table of Contents

1. Introduction

Remote ATtestation procedureS [I-D.ietf-rats-architecture] are workflows composed of roles and interactions, in which a Verifier creates assessments based on evidence about the trustworthiness of an Attester's system component characteristics. The roles _Attester_ and _Verifier_, as well as the message _Evidence_ are terms defined by the RATS Architecture. The goal of this document is to enable the design and adoption of secure conveyance methods for attestation evidence from an Attester to a Verifier.

This document defines three [note: pub/sub & time-based are still missing] reference interaction models that describe the conveyance of evidence between Attester and Verifier in order to provide the basis for reliable and believable appraisal of evidence by a Verifier.

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 entity in the scope of network topologies or the Internet. It rather refers to a decoupled system or different Types of Environments [I-D.ietf-rats-architecture], which also can be present locally as separate system components of a composite device (in a single RATS Entity). Examples include: a Trusted Execution Environment (TEE), Baseboard Management Controllers (BMCs), as well as other physical or logical protected/isolated/shielded Computing Environments.

3. Scope

This document focuses on generic interaction models between Verifiers and Attesters. Complementary procedures, duties and functions that are required for a complete semantic binding of RATS are not in scope. Examples include: identity establishment, key distribution and enrollment, as well as certificate revocation.

Furthermore, any processes and duties that go beyond carrying out remote attestation procedures are out-of-scope. For instance, using the results of a remote attestation that are created by the Verifier, e.g., triggering remediation actions or recovery processes, as well as the remediation actions and recovery processes themselves, is also out-of-scope.

The definition of Reference Interaction Models for RATS uses the role definitions of Attester and Verifier as defined in [I-D.ietf-rats-architecture].

4. Normative Prerequisites

Attester Identity: The provenance of Attestation Evidence with respect to a distinguishable Attesting Environment MUST be correct and unambiguous.

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

Attestation Evidence: Attestation Evidence MUST be a set of well-formatted and well-protected Claims that an Attester can create and convey to a Verifier.

Attestation Evidence Authenticity: Attestation Evidence MUST be correct and authentic.

Attestation Evidence, in order to provide proof of authenticity, SHOULD be cryptographically associated with an identity document (e.g. an X.509 certificate), or SHOULD include a correct and unambiguous reference to an accessible identity document.

Authentication Secret: An Authentication Secret MUST be available exclusively to an Attester's Attesting Environment. The Attester MUST sign Claims with that Authentication Secret, thereby proving the authenticity of the Claims included in the signed Attestation Evidence. The Authentication Secret MUST be established before RATS can take place. How it is established is out-of-scope for this document.

5. 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.

5.1. 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.

Claims ('claims'): _mandatory_

Claims are assertions that represent characteristics of an Attester. Claims compose attestation evidence and are, for example, used to appraise the integrity of an Attester. Examples are Claims about sensor data, policies that are active on the entity, versions of composite firmware of a platform, running software, routing tables, or information about a local time source.

Reference Claims ('refClaims') _mandatory_

Reference Claims are a specific subset of Appraisal Policies as defined in [I-D.ietf-rats-architecture]. Reference Claims are used to appraise the Claims received from an Attester via appraisal by direct comparison. For example, Reference Claims MAY be Reference Integrity Measurements (RIMs) or assertions that are implicitly trusted because they are signed by a trusted authority (see Endorsements in [I-D.ietf-rats-architecture]). RIMs represent (trusted) Claim sets about an Attester's intended platform operational state.

Claim Selection ('claimSelection'): _optional_

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

(Signed) Attestation Evidence ('signedAttestationEvidence'): _mandatory_

Attestation Evidence consists of the Authentication Secret ID that identifies an Authentication Secret, the Attester Identity, the Claims, and the Verifier-provided Nonce. Attestation Evidence

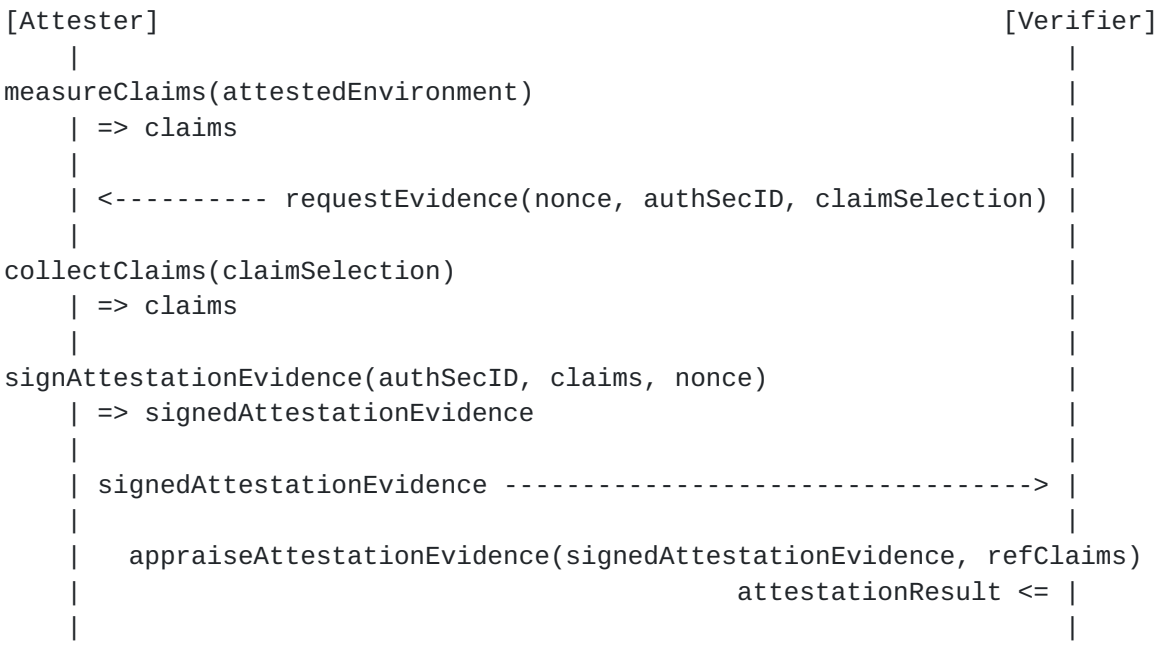
MUST cryptographically bind all of those elements. The Attestation Evidence MUST be signed by the Authentication Secret. The Authentication Secret MUST be trusted by the Verifier as authoritative.

Attestation Result ('attestationResult'): _mandatory_

An Attestation Result is produced by the Verifier as the output of the appraisal of Attestation Evidence. The Attestation Result represents Claims about integrity and other characteristics of the corresponding Attester.

5.2. Interaction Model

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



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 a Claim 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 Claim Selection narrows down or increases the amount of received Claims, if required. If the Claim Selection is empty, then by default all Claims that are available on the Attester MUST be signed and returned as Attestation Evidence. For example, a Verifier may only be requesting a particular subset of information about the Attester, such as evidence

about BIOS and firmware the Attester booted up with - and not include information about all currently running software.

The Attester, after receiving the attestation request, collects the corresponding Claims that have been measured beforehand to compose the Attestation Evidence that the Verifier requested. In the case that the Verifier did not provide a Claim Selection, the Attester collects all information that can be used as complementary Claims in the scope of the semantics of the remote attestation procedure. Conclusively, the Attester creates Attestation Evidence by signing the Attester Identity, the Claims, and the Nonce with the Authentication Secret identified by the Authentication Secret ID. The signed Attestation Evidence is transferred back to the Verifier.

It is crucial at this point that Claims, 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 section Section 8.

As soon as the Verifier receives the signed Attestation Evidence, it verifies the signature, the Attester Identity, the Nonce, and appraises the Claims. This procedure is application-specific and can be carried out by comparing the Claims with corresponding Reference Claims, e.g., Reference Integrity Measurements (RIMs), or using other appraisal policies. The final output of the Verifier are Attestation Results. Attestation Results constitute new Claims about an Attester's properties and characteristics that enables relying parties, for example, to assess an Attester's trustworthiness.

6. Further Context

Depending on the use cases covered, there can be additional requirements. An exemplary subset is illustrated in this section.

6.1. 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.

6.2. 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.

6.3. 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 roots of trusts, such as Hardware Security Modules (HSM), Physically Unclonable Functions (PUFs), Shielded Secrets, or Trusted Executions Environments (TEEs).

7. Implementation Status

Note to RFC Editor: Please remove this section as well as references to [BCP205] before AUTH48.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [BCP205]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [BCP205], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. Implementer

The open-source implementation was initiated and is maintained by the Fraunhofer Institute for Secure Information Technology - SIT.

7.2. Implementation Name

The open-source implementation is named "CHAllenge-Response based Remote Attestation" or in short: CHARRA.

7.3. Implementation URL

The open-source implementation project resource can be located via: https://github.com/Fraunhofer-SIT/charra

7.4. Maturity

The code's level of maturity is considered to be "prototype".

7.5. Coverage and Version Compatibility

The current version (commit '847bcde') is aligned with the exemplary specification of the CoAP FETCH bodies defined in section Appendix A of this document.

7.6. License

The CHARRA project and all corresponding code and data maintained on github are provided under the BSD 3-Clause "New" or "Revised" license.

7.7. Implementation Dependencies

The implementation requires the use of the official Trusted Computing Group (TCG) open-source Trusted Software Stack (TSS) for the Trusted Platform Module (TPM) 2.0. The corresponding code and data is also maintained on github and the project resources can be located via: https://github.com/tpm2-software/tpm2-tss/

The implementation uses the Constrained Application Protocol [RFC7252] (http://coap.technology/) and the Concise Binary Object Representation [RFC7049] (https://cbor.io/).

7.8. Contact

Michael Eckel (michael.eckel@sit.fraunhofer.de)

8. Security and Privacy Considerations

In a remote attestation procedure 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

Olaf Bergmann and Ned Smith

10. Change Log

- Initial draft -00
- Changes from version 00 to version 01:
  - Added details to the flow diagram
  - Integrated comments from Ned Smith (Intel)
  - Reorganized sections and
  - Updated interaction model
  - Replaced "claims" with "assertions"
  - Added proof-of-concept CDDL for CBOR via CoAP based on a TPM 2.0 quote operation
- Changes from version 01 to version 02:
  - Revised the relabeling of "claims" with "assertion" in alignment with the RATS Architecture I-D.
  - Added Implementation Status section
  - Updated interaction model
  - Text revisions based on changes in [I-D.ietf-rats-architecture] and comments provided on rats@ietf.org.

11. References

11.1. Normative References

[BCP205] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[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>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[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>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

11.2. Informative References

[I-D.ietf-rats-architecture] Birkholz, H., Thaler, D., Richardson, M., and N. Smith, "Remote Attestation Procedures Architecture", draft-ietf-rats-architecture-00 (work in progress), December 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