

RATS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

H. Birkholz
Fraunhofer SIT
M. Wiseman
GE Global Research
H. Tschofenig
ARM Ltd.
N. Smith
Intel
M. Richardson
Sandelman Software Works
November 04, 2019

Remote Attestation Procedures Architecture
draft-birkholz-rats-architecture-03

Abstract

An entity (a relying party) requires a source of truth and evidence about a remote peer to assess the peer's trustworthiness. The evidence is typically a believable set of claims about its host, software or hardware platform. This document describes an architecture for such remote attestation procedures (RATS).

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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

<u>1.</u>	<u>Introduction</u>	<u>3</u>
<u>1.1.</u>	<u>Motivation</u>	<u>3</u>
<u>1.2.</u>	<u>Opportunities</u>	<u>3</u>
<u>1.3.</u>	<u>Overview of Document</u>	<u>4</u>
<u>1.4.</u>	<u>RATS in a Nutshell</u>	<u>5</u>
<u>1.5.</u>	<u>Remote Attestation Workflow</u>	<u>5</u>
<u>1.6.</u>	<u>Message Flows</u>	<u>7</u>
<u>1.6.1.</u>	<u>Passport Model</u>	<u>7</u>
<u>1.6.2.</u>	<u>Background Check</u>	<u>8</u>
<u>2.</u>	<u>Terminology</u>	<u>9</u>
<u>3.</u>	<u>Reference use cases</u>	<u>10</u>
<u>3.1.</u>	<u>Device Capabilities/Firmware Attestation</u>	<u>11</u>
<u>3.2.</u>	<u>IETF TEEP WG Use-Case</u>	<u>11</u>
<u>3.3.</u>	<u>Safety Critical Systems</u>	<u>12</u>
<u>3.4.</u>	<u>Virtualized Multi-Tenant Hosts</u>	<u>12</u>
<u>3.5.</u>	<u>Cryptographic Key Attestation</u>	<u>13</u>
<u>3.6.</u>	<u>Geographic Evidence</u>	<u>13</u>
<u>3.7.</u>	<u>Device Provenance Attestation</u>	<u>14</u>
<u>4.</u>	<u>Conceptual Overview</u>	<u>14</u>
<u>4.1.</u>	<u>Two Types of Environments</u>	<u>15</u>
<u>4.2.</u>	<u>Evidence Creation Prerequisites</u>	<u>16</u>
<u>4.3.</u>	<u>Trustworthiness</u>	<u>17</u>
<u>4.4.</u>	<u>Workflow</u>	<u>17</u>
<u>4.5.</u>	<u>Interoperability between RATS</u>	<u>18</u>
<u>5.</u>	<u>RATS Architecture</u>	<u>18</u>
<u>5.1.</u>	<u>Goals</u>	<u>18</u>
<u>5.2.</u>	<u>Attestation Principles</u>	<u>18</u>
<u>5.3.</u>	<u>Attestation Workflow</u>	<u>19</u>
<u>5.3.1.</u>	<u>Roles</u>	<u>19</u>
<u>5.3.2.</u>	<u>Role Messages</u>	<u>20</u>
<u>5.4.</u>	<u>Principals (Entities?) - Containers for the Roles</u>	<u>22</u>
<u>6.</u>	<u>Privacy Considerations</u>	<u>23</u>
<u>7.</u>	<u>Security Considerations</u>	<u>23</u>
<u>8.</u>	<u>Acknowledgements</u>	<u>23</u>
<u>9.</u>	<u>References</u>	<u>23</u>
<u>9.1.</u>	<u>Normative References</u>	<u>24</u>
<u>9.2.</u>	<u>Informative References</u>	<u>24</u>
	<u>Authors' Addresses</u>	<u>25</u>

1. Introduction

Remote Attestation provides a way for an entity (the Relying Party) to determine the health and provenance of an endpoint/host (the Attester). Knowledge of the health of the endpoint allows for a determination of trustworthiness of the endpoint.

1.1. Motivation

The IETF has long spent it's time focusing on threats to the communication channel (see [[RFC3552](#)] and [[DOLEV-YAO](#)]), assuming that endpoints could be trusted and were under the observation of trusted, well-trained professionals. This assumption has not been true since the days of the campus mini-computer. For some time after the desktop PC became ubiquitous, the threat to the endpoints has been dealt with as an internal matter, with generally poor results. Enterprises have done some deployment of Network Endpoint Assessment ([[RFC5209](#)]) to assess the security posture about an endpoint, but it has not been ubiquitous.

The movement towards personal mobile devices ("smartphones") and the continuing threat from unmanaged residential desktops has resulted in a renewed interest in standardizing internet-scale endpoint remote attestation. Additionally, the rise of the Internet of Things (IoT) has made this issue even more critical: some skeptics have even renamed it to the Internet of Threats [[iothreats](#)] :-) IoT devices have poor or non-existent user interfaces, as such as there are not even good ways to assess the health of the devices manually: a need to determine the health via remote attestation is now critical.

In addition to the health of the device, knowledge of its provenance helps to determine the level of trust, and prevents attacks to the supply chain.

1.2. Opportunities

The Trusted Platform Module (TPM) is now a commonly available peripheral on many commodity compute platforms, both servers and desktops. Smartphones commonly have either an actual TPM, or have the ability to emulate one in software running in a Trusted Execution Environment [[I-D.ietf-teep-architecture](#)]. There are now few barriers to creating a standards-based system for remote attestation procedures.

A number of niche solutions have emerged that provide for use-case specific remote attestation, but none have the generality needed to be used across the Internet.

1.3. Overview of Document

The architecture described in this document (along with the accompanying solution and reference documents) enables the use of common formats for communicating Claims about an Attester to a Relying Party. [FIXME Attester? Flows? To what end?]

Existing transports were not designed to carry attestation Claims. It is therefore necessary to design serializations of Claims that fit into a variety of transports, for instance: X.509 certificates, TLS negotiations, YANG modules or EtherNet/IP. There are also new, greenfield uses for remote attestation. Transport and serialization of these can be done without retrofitting. This is (will be) described in [INSERT reference to adopted document on transport].

While it is not anticipated that the existing niche solutions described in the use cases section [Section 3](#) will exchange claims directly, the use of a common format enables common code. As some of the code needs to be in intentionally hard to modify trusted modules, the use of a common formats and transfer protocols significantly reduces the cost of adoption to all parties. This commonality also significantly reduces the incidence of critical bugs.

In some environments the collection of Evidence by the Attester to be provided to the Verifier is part of an existing protocol: this document does not change that, rather embraces those legacy mechanisms as part of the specification. This is an evolutionary path forward, not revolutionary. Yet in other greenfield environments, there is a desire to have a standard for Evidence as well as for Attestation Results, and this architecture outlines how that is done.

This introduction gives an overview of the message flows and roles involved. Following this, is a terminology section that is referenced normatively by other documents and is a key part of this document. There is then a section on use cases and how they leverage the roles and workflows described.

In this document, terms defined within this document are consistently Capitalized [work in progress. please raise issues, if there are Blatant inconsistencies].

Current verticals that use remote attestation include:

- o The Trusted Computing Group "Network Device Attestation Workflow" [[I-D.fedorkow-rats-network-device-attestation](#)]
- o Android Keystore [[keystore](#)]

- o Fast Identity Online (FIDO) Alliance attestation [[fido](#)]
- o A number of Intel SGX niche systems based upon OTRP.

1.4. RATS in a Nutshell

1. Remote Attestation message flows typically convey Claims that contain the trustworthiness properties associated with an Attested Environment (Evidence).
2. A corresponding provisioning message flows conveys Reference trustworthiness claims that can be compared with attestation Evidence. Reference Values typically consist of firmware or software digests and details about what makes the attesting module a trusted source of Evidence.
3. Relying Parties are performing tasks such as managing a resource, controlling access, and/or managing risk. Attestation Results helps Relying Parties determine levels of trust.

1.5. Remote Attestation Workflow

The logical information flow is from Attester to Verifier to Relying Party. There are variations presented below on how this integrates into actual protocols.

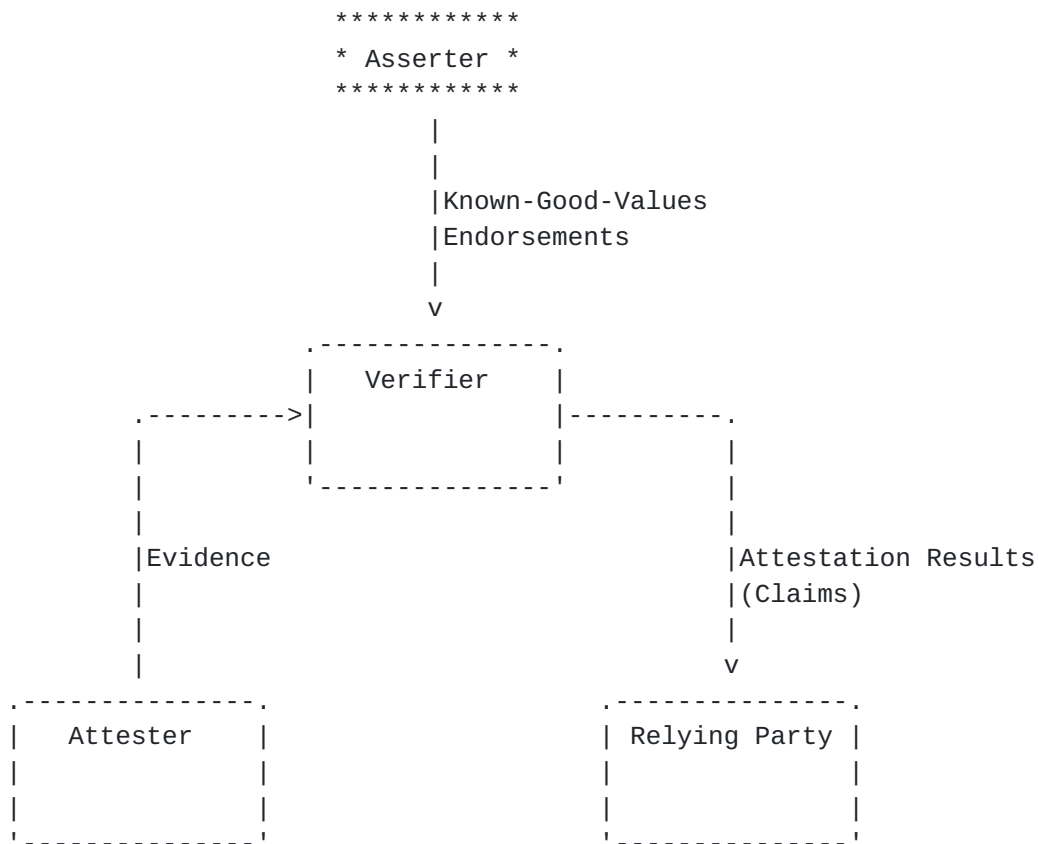


Figure 1: RATS Workflow

In the architecture shown above, specific content items (payload conveyed in message flows) are identified:

- o Evidence is as set of believable Claims about distinguishable Environments made by an Attester.
- o Known-Good-Values are reference Claims used to appraise Evidence by an Verifier.
- o Endorsements are reference Claims about the type of protection that enables an Attester to create believable Evidence. Endorsements enable trust relationships towards system components or environments Evidence cannot be created for by an Attester.
- o Attestation Results are the output from the appraisal of Evidence, Known-Good-Values and Endorsements and are consumed by Relying Parties.

Attestation Results are the output of RATS.

Assessment of Attestation Results is be multi-faceted and out-of-scope for the architecture.

If appropriate Endorsements about the Attester are available, Known-Good-Values about the Attester are available, and if the Attester is capable of creating believable Evidence - then the Verifier is able to create Attestation Results that enable Relying Parties to establish a level of confidence in the trustworthiness of the Attester.

The Asserter role and the format for Known-Good-Values and Endorsements are not subject to standardization at this time. The current verticals already include provisions for encoding and/or distributing these objects.

1.6. Message Flows

Two distinct flows have been identified for passage of Evidence and production of Attestation Results. It is possible that there are additional situations which are not captured by these two flows.

1.6.1. Passport Model

In the Passport Model message flow the Attester provides it's Evidence directly to the Verifier. The Verifier will evaluate the Evidence and then sign an Attestation Result. This Attestation Result is returned to the Attester, and it is up to the Attester to communicate the Attestation Result (potentially including the Evidence, if disclosable) to the Relying Party.

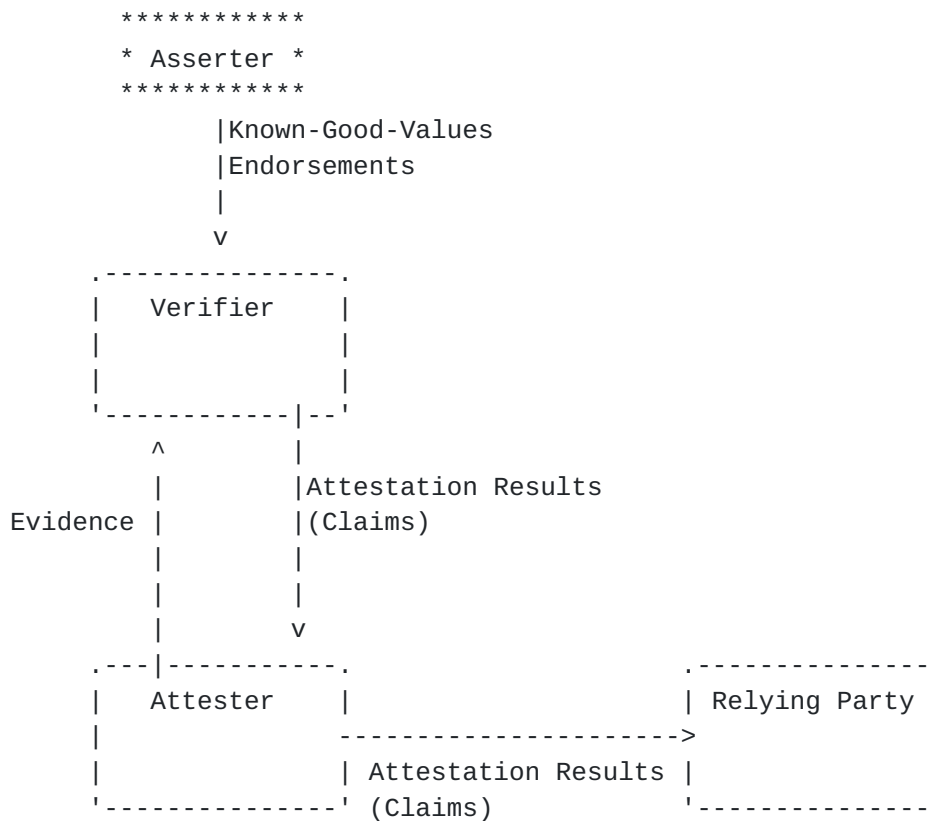


Figure 2: RATS Passport Flow

This flow is named in this way because of the resemblance of how Nations issue Passports to their citizens. The nature of the Evidence that an individual needs to provide to it's local authority is specific to the country involved. The citizen retains control of the resulting document and presents it to other entities when it needs to assert a citizenship or identity claim.

[1.6.2.](#) Background Check

In the Background-Check message flow the Attester provides it's Evidence to the Relying Party. The Relying Party sends this evidence to a Verifier of its choice. The Verifier will evaluate the Evidence and then sign an Attestation Result. This Attestation Result is returned to the Relying Party, which processes it directly.

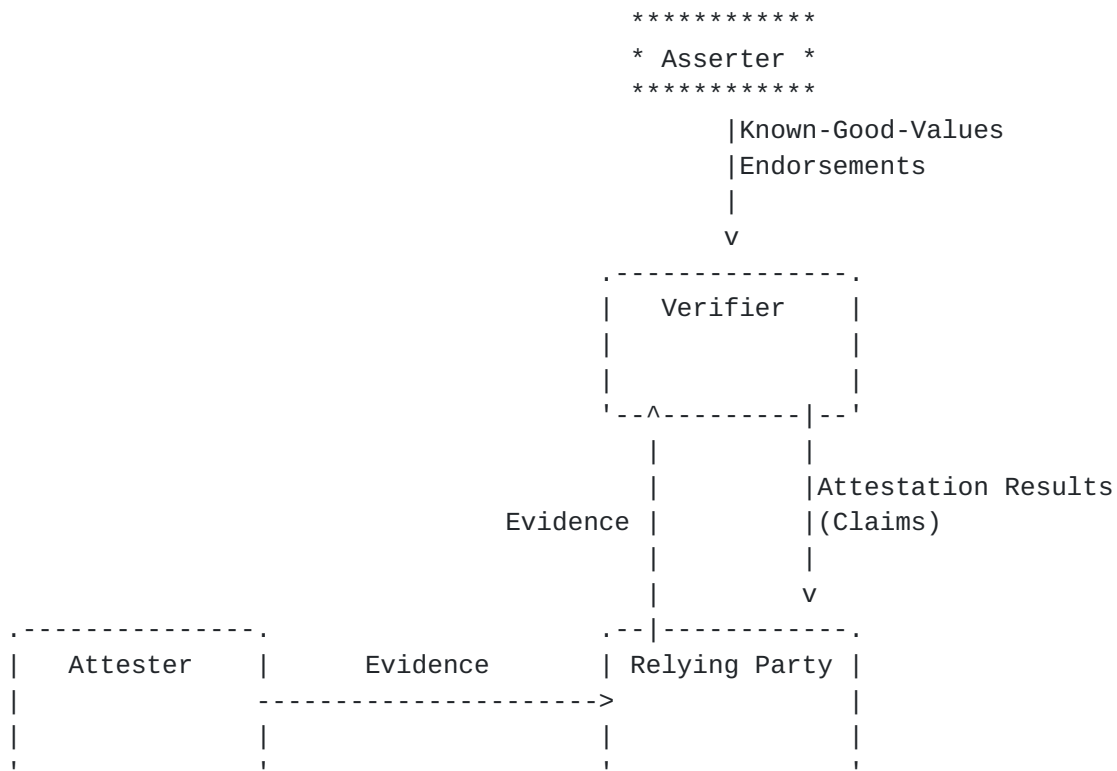


Figure 3: RATS Background Check Flow

This flow is named in this way because of the resemblance of how employers and volunteer organizations perform background checks. When a prospective employee provides claims about education or previous experience, the employer will contact the respective institutions or former employers to validate the claim. Volunteer organizations often perform police background checks on volunteers in order to determine the volunteer's trustworthiness.

2. Terminology

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.

Appraisal: A Verifier process that compares Evidence to Reference values while taking into account Endorsements and produces Attestation Results.

Asserter: See [Section 5.3.1.2](#).

Attester: See [Section 5.3.1.1](#).

Attested Environment: A target environment that is observed or controlled by an Attesting Environment.

Attesting Environment: An environment capable of making trustworthiness Claims about an Attested Environment.

Background-Check Message Flow: An attestation workflow where the Attester provides Evidence to a Relying Party, who consults one or more Verifiers who supply Attestation Results to the Relying Party. See [Section 1.6.2](#).

Claim: A statement about the construction, composition, validation or behavior of an Entity that affects trustworthiness. Evidence, Reference Values and Attestation Results are expressions that consists of one or more Claims.

Conveyance: The process of transferring Evidence, Reference Values and Attestation Results between Entities participating in attestation workflow.

Entity: A device, component (see System Component [[RFC4949](#)]), or environment that implements one or more Roles.

Evidence: See [Section 5.3.2.1](#).

Passport Message Flow: An attestation workflow where the Attester provides Evidence to a Verifier who returns Attestation Results that are then forwarded to one or more Relying Parties. See [Section 1.6.1](#).

Reference Values: See [Section 5.3.2.2](#). Also referred to as Known-Good-Values.

Relying Party: See [Section 5.3.1.4](#).

Attestation Results: See [Section 5.3.2.3](#).

Role: A function or process in an attestation workflow, typically described by: Attester, Verifier, Relying Party and Asserter.

Verifier: See [Section 5.3.1.3](#).

3. Reference use cases

This section provides an overview of a number of distinct use cases that benefit from a standardized claim format. In addition to outlining the user, the specific message flow is identified from among the flows detailed in [Section 1.6](#).

3.1. Device Capabilities/Firmware Attestation

This is a large category of claims that includes a number of subcategories, not detailed here.

Use case name: Device Identity

Who will use it: Network Operators, larger enterprises

Attester: varies

Message Flow: sometimes passport and sometimes background check

Relying Party: varies

Description: Network operators want a trustworthy report of identity and version of information of the hardware and software on the machines attached to their network. The process starts with some kind of Root of Trust that provides device identity and protected storage for measurements. The mechanism performs a series of measurements, and expresses this with an attestation as to the hardware and firmware/software which is running.

This is a general description for which there are many specific use cases, including [[I-D.fedorkow-rats-network-device-attestation](#)] [section 1.2](#), "Software Inventory"

3.2. IETF TEEP WG Use-Case

Use case name: TAM validation

Who will use it: The TAM server

Message Flow: background check

Attester: Trusted Execution Environment (TEE)

Relying Party: end-application

Description: The "Trusted Application Manager (TAM)" server wants to verify the state of a TEE, or applications in the TEE, of a device. The TEE attests to the TAM, which can then decide whether to install sensitive data in the TEE, or whether the TEE is out of compliance and the TAM needs to install updated code in the TEE to bring it back into compliance with the TAM's policy.

3.3. Safety Critical Systems

Use case name: Safety Critical Systems

Who will use it: Power plants and other systems that need to assert their current state, but which can not accept any inputs from the outside. The corollary system is a black-box (such as in an aircraft), which needs to log the state of a system, but which can never initiate a handshake.

Message Flow: background check

Attester: web services and other sources of status/sensor information

Relying Party: open

Claims used as Evidence: the beginning and ending time as endorsed by a Time Stamp Authority, represented by a time stamp token. The real time clock of the system itself. A Root of Trust for time; the TPM has a relative time from startup.

Description: These requirements motivate the creation of the Time-Base Unidirectional Attestation (TUDA) [[I-D.birkholz-rats-tuda](#)], the output of TUDA is typically a secure audit log, where freshness is determined by synchronization to a trusted source of external time.

The freshness is preserved in the Evidence by the use of a Time Stamp Authority (TSA) which provides Time Stamp Tokens (TST).

3.4. Virtualized Multi-Tenant Hosts

Use case name: Multi-Tenant Hosts

Who will use it: Virtual machine systems

Message Flow: passport

Attester: virtual machine hypervisor

Relying Party: network operators

Description: The host system will do verification as per [Section 3.1](#)

The tenant virtual machines will do verification as per [Section 3.1](#).

The network operator wants to know if the system _as a whole_ is free of malware, but the network operator is not allowed to know who the tenants are.

This is contrasted to the Chassis + Line Cards case (To Be Defined: TBD).

Multiple Line Cards, but a small attestation system on the main card can combine things together. This is a kind of proxy.

3.5. Cryptographic Key Attestation

Cryptographic Attestation includes subcategories such as Device Type Attestation (the FIDO use case), and Key storage Attestation (the Android Keystore use case), and End-User Authorization.

Use case name: Key Attestation

Who will use it: network authentication systems

Message Flow: passport

Attester: device platform

Relying Party: internet peers

Description: The relying party wants to know how secure a private key that identifies an entity is. Unlike the network attestation, the relying party is not part of the network infrastructure, nor do they necessarily have a business relationship (such as ownership) over the end device.

The Device Type Attestation is provided by a Firmware TPM performing the Verifier function, creating Attestation Results that indicate a particular model/type of device. In TCG terms, this is called Implicit Attestation, in this case the Attested Environment is the (smartphone) Rich Execution Environment (REE) ([\[I-D.ietf-teep-architecture\]](#) [section 2](#)), and the Attesting Environment is within the TEE.

3.6. Geographic Evidence

Use case name: Location Evidence

Who will use it: geo-fenced systems

Message Flow: passport (probably)

Attester: secure GPS system(s)

Relying Party: internet peers

Description: The relying party wants to know the physical location (on the planet earth, using a geodetic system) of the device. This may be provided directly by a GPS/GLONASS/BeiDou/Galileo module that is incorporated into a TPM. This may also be provided by collecting other proximity messages from other device that the relying party can form a trust relationship with.

3.7. Device Provenance Attestation

Use case name: RIV - Device Provenance

Who will use it: Industrial IoT devices

Message Flow: passport

Attester: network management station

Relying Party: a network entity

Description: A newly manufactured device needs to be onboarded into a network where many if not all device management duties are performed by the network owner. The device owner wants to verify the device originated from a legitimate vendor. A cryptographic device identity such as an IEEE802.1AR is embedded during manufacturing and a certificate identifying the device is delivered to the owner onboarding agent. The device authenticates using its 802.1AR IDevID to prove it originated from the expected vendor.

The device chain of custody from the original device manufacturer to the new owner may also be verified as part of device provenance attestation. The chain of custody history may be collected by a cloud service or similar capability that the supply chain and owner agree to use.

[I-D.fedorkow-rats-network-device-attestation] [section 1.2](#) refers to this as "Provable Device Identity", and [section 2.3](#) details the parties.

4. Conceptual Overview

In network protocol exchanges, it is often the case that one entity (a Relying Party) requires an assessment of the trustworthiness of a remote entity (an Attester or specific system components [[RFC4949](#)])

thereof). Remote Attestation procedures (RATS) enable Relying Parties to establish a level of confidence in the trustworthiness of Attesters through the

- o Creation,
- o Conveyance, and
- o Appraisal

of attestation Evidence.

Qualities of Evidence: Evidence is composed of Claims about trustworthiness (the set of Claims is unbounded). The system characteristics of Attesters - the Environments they are composed of, and their continuous development - have an impact on the veracity of trustworthiness Claims included in valid Evidence.

Valid Evidence about the intactness of an Attester must be impossible to create by an untrustworthy or compromised Environment of an Attester.

Qualities of Environments: The resilience of Environments that are part of an Attester can vary widely - ranging from those highly resistant to attacks to those having little or no resistance to attacks. Configuration options, if set poorly, can result in a highly resistant environment being operationally less resistant. When a trustworthy Environment changes, it is possible that it transitions from being trustworthy to being untrustworthy.

An untrustworthy or compromised Environment must never be able to create valid Evidence expressing the intactness of an Attester.

The architecture provides a framework for anticipating when a relevant change with respect to a trustworthiness attribute occurs, what exactly changed and how relevant it is. The architecture also creates a context for enabling an appropriate response by applications, system software and protocol endpoints when changes to trustworthiness attributes do occur.

Detailed protocol specifications for message flows are defined in separate documents.

4.1. Two Types of Environments

An Attester produces Evidence about its own integrity, which means "it measures itself". To disambiguate this recursive or circular

looking relationships, two types of Environments inside an Attester are distinguished:

The attest-ED Environments and the attest-ING Environments.

Attested Environments are measured. They provide the raw values and the information to be represented in Claims and ultimately expressed as Evidence.

Attesting Environments conduct the measuring. They collect the Claims, format them appropriately, and typically use key material and cryptographic functions, such as signing or cipher algorithms, to create Evidence.

Attesting Environments use system components that have to be trusted. As a result, Evidence includes Claims about the Attested and the Attesting Environments. Claims about the Attested Environments are appraised using Reference Values and Claims about the Attesting Environments are appraised using Endorsements. It is not mandated that both Environments have to be separate, but it is highly encouraged. Examples of separated Environments that can be used as Attesting Environments include: Trusted Execution Environments (TEE), embedded Secure Elements (eSE), or Hardware Security Modules (HSM).

In summary, the majority of the creation of evidence can take place in an Attested Environments. Exemplary duties include the collection and formatting of Claim values, or the trigger for creating Evidence. A trusted sub-set of the creation of evidence can take place in an Attesting Environment, that provide special protection with respect to key material, identity documents, or primitive functions to create the Evidence itself.

4.2. Evidence Creation Prerequisites

One or more Environments that are part of an Attester must be able to conduct the following duties in order to create Evidence:

- o monitoring trustworthiness attributes of other Environments,
- o collecting trustworthiness attributes and create Claims about them,
- o serialize Claims using interoperable representations,
- o provide integrity protection for the sets of Claims, and
- o add appropriate attestation provenance attributes about the sets of Claims.

4.3. Trustworthiness

The trustworthiness of an Attester and therefore the believability of the Evidence it creates relies on the protection methods in place to shield and restrict the use of key material and the duties conducted by the Attesting Environment. In order to assess trustworthiness effectively, it is mandatory to understand the trustworthiness properties of the environments of an Attester. The corresponding appraisal of Evidence that leads to such an assessment of trustworthiness is the duty of a Verifier.

Trusting the assessment of a Verifier might come from trusting the Verifier's key material (direct trust), or trusting an Entity that the Verifier is associated with via a certification path (indirect trust).

The trustworthiness of corresponding Attestation Results also relies on trust towards manufacturers and those manufacturer's hardware in order to assess the integrity and resilience of that manufacturer's devices.

A stronger level of security comes when information can be vouched for by hardware or by (unchangeable) firmware, especially if such hardware is physically resistant to hardware tampering. The component that is implicitly trusted is often referred to as a Root of Trust.

4.4. Workflow

The basic function of RATS is creation, conveyance and appraisal of attestation Evidence. An Attester creates attestation Evidence that are conveyed to a Verifier for appraisal. The appraisals compare Evidence with expected Known-Good-Values obtained from Asserters (e.g. Principals that are Supply Chain Entities). There can be multiple forms of appraisal (e.g., software integrity verification, device composition and configuration verification, device identity and provenance verification). Attestation Results are the output of appraisals. Attestation Results are signed and conveyed to Relying Parties. Attestation Results provide the basis by which the Relying Party may determine a level of confidence to place in the application data or operations that follow.

The architecture defines attestation Roles: Attester, Verifier, Asserter and Relying Party. Roles exchange messages, but their structure is not defined in this document. The detailed definition of the messages is in an appropriate document, such as [\[I-D.ietf-rats-eat\]](#) or other protocols to be defined. Roles can be combined in various ways into Principals, depending upon the needs of

the use case. Information Model representations are realized as data structure and conveyance protocol specifications.

[4.5.](#) Interoperability between RATS

The RATS architecture anticipates use of information modeling techniques that describe computing structures - their components/ computational elements and corresponding capabilities - so that verification operations may rely on the information model as an interoperable way to navigate the structural complexity.

[5.](#) RATS Architecture

[5.1.](#) Goals

RATS architecture has the following goals:

- o Enable semantic interoperability of attestation semantics through information models about computing environments and trustworthiness.
- o Enable data structure interoperability related to claims, endpoint composition / structure, and end-to-end integrity and confidentiality protection mechanisms.
- o Enable programmatic assessment of trustworthiness. (Note: Mechanisms that manage risk, justify a level of confidence, or determine a consequence of an attestation result are out of scope).
- o Provide the building blocks, including Roles and Principals that enable the composition of service-chains/hierarchies and workflows that can create and appraise evidence about the trustworthiness of devices and services.
- o Use-case driven architecture and design (see [\[I-D.richardson-rats-usecases\]](#) and [Section 3](#))
- o Terminology conventions that are consistently applied across RATS specifications.
- o Reinforce trusted computing principles that include attestation.

[5.2.](#) Attestation Principles

Specifications developed by the RATS working group apply the following principles:

- o Freshness - replay of previously asserted Claims about an Attested Environment can be detected.
- o Identity - the Attesting Environment is identifiable (non-anonymous).
- o Context - the Attested Environment is well-defined (unambiguous).
- o Provenance - the origin of Claims with respect to the Attested and Attesting Environments are known.
- o Validity - the expected lifetime of Claims about an Attested Environment is known.
- o Veracity - the believability (level of confidence) of Claims is based on verifiable proofs.

5.3. Attestation Workflow

Attestation workflow helps a Relying Party make better decisions by providing insight about the trustworthiness of endpoints participating in a distributed system. The workflow consists primarily of four roles; Relying Party, Verifier, Attester and Asserter. Attestation messages contain information useful for appraising the trustworthiness of an Attester endpoint and informing the Relying Party of the appraisal result.

This section details the primary roles of an attestation workflow and the messages they exchange.

5.3.1. Roles

An endpoint system (a.k.a., Entity) may implement one or more attestation Roles to accommodate a variety of possible message flows. Exemplary message flows are described in [Section 1.6.1](#) and [Section 1.6.2](#). Role messages are secured by the Entity that generated it. Entities possess credentials (e.g., cryptographic keys) that authenticate, integrity protect and optionally confidentiality protect attestation messages.

5.3.1.1. Attester

The Attester consists of both an Attesting Environment and an Attested Environment. In some implementations these environments may be combined. Other implementations may have multiples of Attesting and Attested environments. Although endpoint environments can be complex, and that complexity is security relevant, the basic function

of an Attester is to create Evidence that captures operational conditions affecting trustworthiness.

5.3.1.2. Asserter

The Asserter role is out of scope. The mechanism by which an Asserter communicates Known-Good-Values to a Verifier is also not subject to standardization. Users of the RATS architecture are assumed to have pre-existing mechanisms.

5.3.1.3. Verifier

The Verifier workflow function accepts Evidence from an Attester and accepts Reference Values from one or more Asserters. Reference values may be supplied a priori, cached or used to create policies. The Verifier performs an appraisal by matching Claims found in Evidence with Claims found in Reference Values and policies. If an attested Claim value differs from an expected Claim value, the Verifier flags this as a change possibly impacting trust level.

Endorsements may not have corresponding Claims in Evidence (because of their intrinsic nature). Consequently, the Verifier need only authenticate the endpoint and verify the Endorsements match the endpoint identity.

The result of appraisals and Endorsements, informed by owner policies, produces a new set of Claims that a Relying Party is suited to consume.

5.3.1.4. Relying Party

A Role in an attestation workflow that accepts Attestation Results from a Verifier that may be used by the Relying Party to inform application specific decision making. How Attestation Results are used to inform decision making is out-of-scope for this architecture.

5.3.2. Role Messages

5.3.2.1. Evidence

Claims that are formatted and protected by an Attester.

Evidence SHOULD satisfy Verifier expectations for freshness, identity, context, provenance, validity, and veracity.

5.3.2.2. Reference Values

Reference-values are Claims that a manufacturer, vendor or other supply chain entity makes that affects the trustworthiness of an Attester endpoint.

Claims may be persistent properties of the endpoint due to the physical nature of how it was manufactured or may reflect the processes that were followed as part of moving the endpoint through a supply-chain; e.g., validation or compliance testing. This class of Reference-values is known as Endorsements.

Another class of Reference-values identifies the firmware and software that could be installed in the endpoint after its manufacture. A digest of the the firmware or software can be an effective identifier for keeping track of the images produced by vendors and installed on an endpoint. This class of Reference-value is referred to as Known-Good-Value (KGV).

Known-Good-Values: Claims about the Attested Environment.

Typically, Known-Good-Value (KGV) Claims are message digests of firmware, software or configuration data supplied by various vendors. If an Attesting Environment implements cryptography, they include Claims about key material.

Like Claims, Known-Good-Values SHOULD satisfy a Verifier's expectations for freshness, identity, context, provenance, validity, relevance and veracity. Known-Good-Values are reference Claims that are - like Evidence - well formatted and protected (e.g. signed).

Endorsements: Claims about immutable and implicit characteristics of the Attesting Environment. Typically, endorsement Claims are created by manufacturing or supply chain entities.

Endorsements are intended to increase the level of confidence with respect to Evidence created by an Attester.

5.3.2.3. Attestation Results

Statements about the output of an appraisal of Evidence that are created, formatted and protected by a Verifier.

Attestation Results provide the basis for a Relying Party to establish a level of confidence in the trustworthiness of an Attester. Attestation Results SHOULD satisfy Relying Party expectations for freshness, identity, context, provenance, validity, relevance and veracity.

5.4. Principals (Entities?) - Containers for the Roles

[The authors are unhappy with the term Principal, and have been looking for something else. JOSE/JWT uses the term Principal]

Principals are Containers for the Roles.

Principals are users, organizations, devices and computing environments (e.g., devices, platforms, services, peripherals).

Principals may implement one or more Roles. Message flows occurring within the same Principal are out-of-scope.

The methods whereby Principals may be identified, discovered, authenticated, connected and trusted, though important, are out-of-scope.

Principal operations that apply resiliency, scaling, load balancing or replication are generally believed to be out-of-scope.

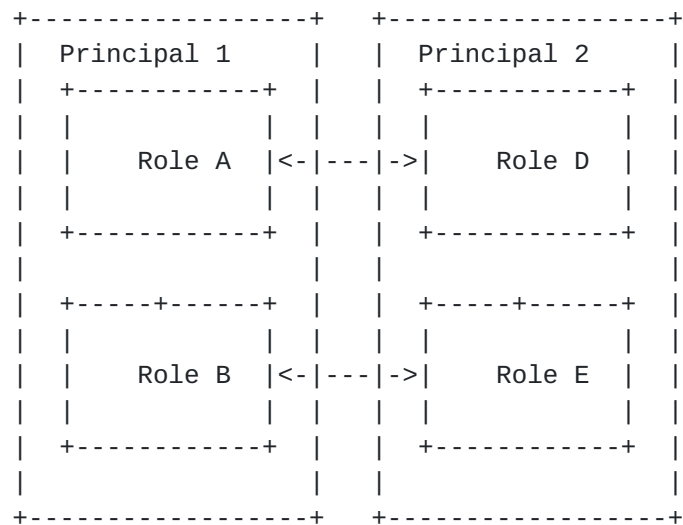


Figure 4: Principals-Role Composition

Principals have the following properties:

- o Multiplicity - Multiple instances of Principals that possess the same Roles can exist.
- o Composition - Principals possessing different Roles can be combined into a singleton Principal possessing the union of Roles. Message flows between combined Principals is uninteresting.

- o Decomposition - A singleton Principal possessing multiple Roles can be divided into multiple Principals.

6. Privacy Considerations

The conveyance of Evidence and the resulting Attestation Results reveal a great deal of information about the internal state of a device. In many cases the whole point of the Attestation process is to provide reliable evidence about the type of the device and the firmware that the device is running. This information is particularly interesting to many attackers: knowing that a device is running a weak version of the firmware provides a way to aim attacks better.

Just knowing the existence of a device is itself a disclosure.

Conveyance protocols must detail what kinds of information is disclosed, and to whom it is exposed.

7. Security Considerations

Evidence, Verifiable Assertions and Attestation Results SHOULD use formats that support end-to-end integrity protection and MAY support end-to-end confidentiality protection.

Replay attacks are a concern that protocol implementations MUST deal with. This is typically done via a Nonce Claim, but the details belong to the protocol.

All other attacks involving RATS structures are not explicitly addressed by the architecture.

Additional security protections MAY be required of conveyance mechanisms. For example, additional means of authentication, confidentiality, integrity, replay, denial of service and privacy protection of RATS payloads and Principals may be needed.

8. Acknowledgements

Dave Thaler created the concepts of "Passport" and "Background Check".

9. References

9.1. 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>>.

9.2. Informative References

- [ABLP] Abadi, M., Burrows, M., Lampson, B., and G. Plotkin, "A Calculus for Access Control in Distributed Systems", Springer Annual International Cryptology Conference, page 1-23, DOI 10.1.1.36.691, 1991.
- [DOLEV-YAO] Dolev, D. and A. Yao, "On the security of public key protocols", IEEE Transactions on Information Theory Vol. 29, pp. 198-208, DOI 10.1109/tit.1983.1056650, March 1983.
- [fido] FIDO Alliance, ., "FIDO Specification Overview", 2019, <<https://fidoalliance.org/specifications/>>.
- [I-D.birkholz-rats-tuda] Fuchs, A., Birkholz, H., McDonald, I., and C. Bormann, "Time-Based Uni-Directional Attestation", [draft-birkholz-rats-tuda-01](#) (work in progress), September 2019.
- [I-D.fedorkow-rats-network-device-attestation] Fedorkow, G. and J. Fitzgerald-McKay, "Network Device Attestation Workflow", [draft-fedorkow-rats-network-device-attestation-00](#) (work in progress), July 2019.
- [I-D.ietf-rats-eat] Mandyam, G., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", [draft-ietf-rats-eat-01](#) (work in progress), July 2019.
- [I-D.ietf-teep-architecture] Pei, M., Tschofenig, H., Wheeler, D., Atyeo, A., and D. Liu, "Trusted Execution Environment Provisioning (TEEP) Architecture", [draft-ietf-teep-architecture-03](#) (work in progress), July 2019.

[I-D.richardson-rats-usecases]

Richardson, M., Wallace, C., and W. Pan, "Use cases for Remote Attestation common encodings", [draft-richardson-rats-usecases-05](#) (work in progress), October 2019.

[iothreats]

GDN, ., "The Internet of Things or the Internet of threats?", 2016, <<https://gcn.com/articles/2016/05/03/internet-of-threats.aspx>>.

[keystore]

Google, ., "Android Keystore System", 2019, <<https://developer.android.com/training/articles/keystore>>.

[Lampson2007]

Lampson, B., "Practical Principles for Computer Security", IOSPress Proceedings of Software System Reliability and Security, page 151-195, DOI 10.1.1.63.5360, 2007.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", [BCP 72](#), [RFC 3552](#), DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.

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

[RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", [RFC 5209](#), DOI 10.17487/RFC5209, June 2008, <<https://www.rfc-editor.org/info/rfc5209>>.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Monty Wiseman
GE Global Research
USA

Email: monty.wiseman@ge.com

Hannes Tschofenig
ARM Ltd.
110 Fulbourn Rd
Cambridge CB1 9NJ
UK

Email: hannes.tschofenig@gmx.net

Ned Smith
Intel Corporation
USA

Email: ned.smith@intel.com

Michael Richardson
Sandelman Software Works
Canada

Email: mcr+ietf@sandelman.ca

