Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

D. Thaler
Microsoft

November 04, 2019

Remote Attestation Architecture
draft-thaler-rats-architecture-01

Abstract

In network protocol exchanges, it is often the case that one entity (a relying party) requires evidence about the remote peer (and system components [RFC4949] thereof), in order to assess the trustworthiness of the peer. This document describes an architecture for such remote attestation procedures (RATS), which enable relying parties to decide whether to consider a remote system component trustworthy or not.

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 http://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

Table of Contents

1. Introduction

In network protocol exchanges, it is often the case that one entity (a relying party) requires evidence about the remote peer (and system components [RFC4949] thereof), in order to assess the trustworthiness of the peer. Remote attestation procedures (RATS) enable relying parties to establish a level of confidence in the trustworthiness of remote system components through the creation of attestation evidence by remote system components and a processing chain towards the relying party. A relying party can then decide whether to consider a remote system component trustworthy or not.

To improve the confidence in a system component's trustworthiness, a relying party may require evidence about:

- system component identity,

- composition of system components, including nested components,
- roots of trust,
- assertion/claim origination or provenance,
- manufacturing origin,
- system component integrity,
- system component configuration,
- operational state and measurements of steps which led to the operational state, or
- other factors that could influence trust decisions.

This document discusses an architecture for describing solutions for this problem.

2. Terminology

This document uses the following terms:

- Attestation: A process by which one entity (the "Attester") provides evidence about its identity and health to another entity, which then assesses its trustworthiness.
- Attestation Result: The evaluation results generated by a Verifier, typically including information about an Attester, where the Verifier vouches for the validity of the results.
- Attester: An entity whose attributes must be evaluated in order to determine whether the entity is considered healthy or authorized to access a resource.
- Endorsement: A secure statement that some entity (typically a manufacturer) vouches for the integrity of an Attester's signing capability. (Note: in some discussions the entity providing an Endorsement has been called an Asserter, but some believe that term is confusing and the term Endorser would be more correct. For now, this document avoids using a specific term until consensus is reached.)
- Evidence: A set of information about an Attester that is to be evaluated by a Verifier.

- Relying Party: An entity that depends on the validity of information about another entity, typically for purposes of authorization. Compare /relying party/ in [RFC4949].

- Security policy: A set of rules that direct how a system evaluates the validity of information about another entity. For example, the security policy might involve an equality comparison against known-good values (called Reference Integrity Measurements in some contexts), or might involve more complex logic. Compare /security policy/ in [RFC4949].

- Verifier: An entity that evaluates the validity of information about an Attester.

3. Use Cases

This section covers a number of representative use cases for attestation, independent of solution. The purpose is to provide motivation for various aspects of the architecture presented in this draft. Many other use cases exist, and this document does not intend to have a complete list, only to have a set of use cases that collectively cover all the functionality required in the architecture. The use cases are covered prior to discussion of architectural models in Section 5, since each use case might be addressed via different solutions that have different architectural models.

Each use case includes a description, and a summary of what an Attester and a Relying Party refer to in the use case. (Since solutions to a use case may greatly vary in architectural model, the role of a Verifier is considered part of a specific solution, not a solution-independent property of a use case, and so is not covered in this section.)

3.1. Network Endpoint Assessment

Network operators want a trustworthy report of identity and version of information of the hardware and software on the machines attached to their network, for purposes such as inventory, auditing, and/or logging. The network operator may also want a policy by which full access is only granted to devices that meet some definition of health, and so wants to get claims about such information and verify their validity. Attestation is desired to prevent vulnerable or compromised devices from getting access to the network and potentially harming others.

Typically, solutions start with some component (called a "Root of Trust") that provides device identity and protected storage for

measurements. They then perform a series of measurements, and express this with Evidence as to the hardware and firmware/software that is running.

- Attester: A device desiring access to a network

- Relying Party: A network infrastructure device such as a router, switch, or access point.

3.2. Confidential Machine Learning (ML) Model Protection

A device manufacturer wants to protect its intellectual property in terms of the ML model it developed and that runs in the devices that its customers purchased, and it wants to prevent attackers, potentially including the customer themselves, from seeing the details of the model.

This typically works by having some protected environment in the device attest to some manufacturer service. If attestation succeeds. then the manufacturer service releases either the model, or a key to decrypt a model the Attester already has in encrypted form, to the requester.

- Attester: A device desiring to run an ML model to do inferencing

- Relying Party: A server or service holding ML models it desires to protect

3.3. Confidential Data Retrieval

This is a generalization of the ML model use case above, where the data can be any highly confidential data, such as health data about customers, payroll data about employees, future business plans, etc. Attestation is desired to prevent leaking data to compromised devices.

- Attester: An entity desiring to retrieve confidential data

- Relying Party: An entity that holds confidential data for retrieval by other entities

3.4. Critical Infrastructure Control

In this use case, potentially dangerous physical equipment (e.g., power grid, traffic control, hazardous chemical processing, etc.) is connected to a network. The organization managing such infrastructure needs to ensure that only authorized code and users can control such processes, and they are protected from malware or

other adversaries. When a protocol operation can affect some critical system, the device attached to the critical equipment thus wants some assurance that the requester has not been compromised. As such, attestation can be used to only accept commands from requesters that are within policy.

- Attester: A device or application wishing to control physical equipment.

- Relying Party: A device or application connected to potentially dangerous physical equipment (hazardous chemical processing, traffic control, power grid, etc.

3.5. Trusted Execution Environment (TEE) Provisioning

A "Trusted Application Manager (TAM)" server is responsible for managing the applications running in the TEE of a client device. To do this, the TAM wants to verify the state of a TEE, or of applications in the TEE, of a client device. The TEE attests to the TAM, which can then decide whether the TEE is already in compliance with the TAM's latest policy, or if the TAM needs to uninstall, update, or install approved applications in the TEE to bring it back into compliance with the TAM's policy.

- Attester: A device with a trusted execution environment capable of running trusted applications that can be updated.

- Relying Party: A Trusted Application Manager.

3.6. Hardware Watchdog

One significant problem is malware that holds a device hostage and does not allow it to reboot to prevent updates to be applied. This is a significant problem, because it allows a fleet of devices to be held hostage for ransom.

A hardware watchdog can be implemented by forcing a reboot unless attestation to a remote server succeeds within a periodic interval, and having the reboot do remediation by bringing a device into compliance, including installation of patches as needed.

- Attester: The device that is desired to keep from being held hostage for a long period of time.

- Relying Party: A remote server that will securely grant the Attester permission to continue operating (i.e., not reboot) for a period of time.

4. Serialization Formats

The following diagram illustrates a relationship to which attestation is desired to be added:

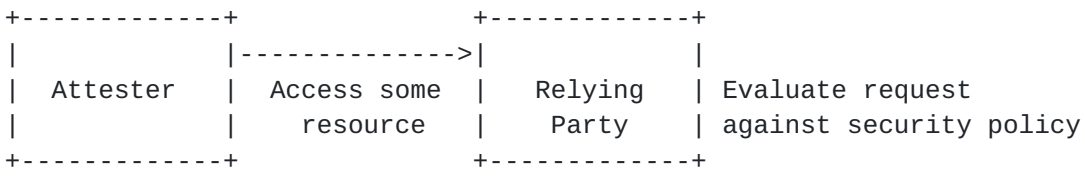

Figure 1: Typical Resource Access

In this diagram, the protocol between Attester and a Relying Party can be any new or existing protocol (e.g., HTTP(S), COAP(S), 802.1x, OPC UA, etc.), depending on the use case. Such protocols typically already have mechanisms for passing security information for purposes of authentication and authorization. Common formats include JWTs [RFC7519], CWTs [RFC8392], and X.509 certificates.

In many cases, it is desirable to add attestation to existing protocols, enabling a higher level of assurance against malware for example. To enable such integration, it is important that information needed for evaluating the Attester be usable with existing protocols that have constraints around what formats they can transport. For example, OPC UA [OPCUA] (probably the most common protocol in industrial IoT environments) is defined to carry X.509 certificates and so security information must be embedded into an X.509 certificate to be passed in the protocol. Thus, attestation-related information could be natively encoded in X.509 certificate extensions, or could be natively encoded in some other format (e.g., a CWT) which in turn is then encoded in an X.509 certificate extension.

Especially for constrained nodes, however, there is a desire to minimize the amount of parsing code needed in a Relying Party, in order to both minimize footprint and to minimize the attack surface area. So while it would be possible to embed a CWT inside a JWT, or a JWT inside an X.509 extension, etc., there is a desire to encode the information natively in the format that is natural for the Relying Party.

This motivates having a common "information model" that describes the set of attestation related information in an encoding-agnostic way, and allowing multiple serialization formats (CWT, JWT, X.509, etc.) that encode the same information into the format needed by the Relying Party.

5. Architectural Models

There are multiple possible models for communication between an Attester, a Verifier, and a Relying Party.

5.1. Passport Model

In this model, an Attester sends Evidence to a Verifier, which compares the Evidence against its security policy. The Verifier then gives back an Attestation Result. If the Attestation Result was a successful one, the Attester can then present the Attestation Result to a Relying Party, which then compares the Attestation Result against its own security policy.

Since the resource access protocol between the Attester and Relying Party includes an Attestation Result, in this model the details of that protocol constrain the serialization format of the Attestation Result. The format of the Evidence on the other hand is only constrained by the Attester-Verifier attestation protocol.

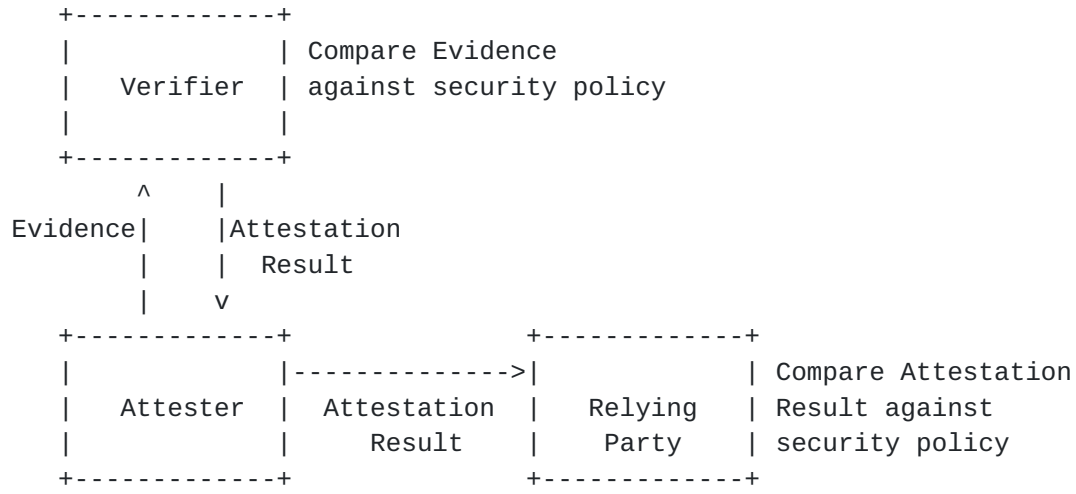

Figure 2: Passport Model

The passport model is so named because it resembles the process typically used for passports and drivers licenses, where a person applies and gets a passport or license that is issued by a government and shows information such as the person's name and birthdate. The passport or license can then be supplied to other entities to gain entrance to an airport boarding area, or age-restricted section of a bar, where the passport or license is considered sufficient because it vouches for that piece of information and is issued by a trusted authority. Thus, in this analogy, the passport issuing agency is a Verifier, the passport is an Attestation Result, and the airport security is a Relying Party.

5.2. Background-Check Model

In this model, an Attester sends Evidence to a Relying Party, which simply passes it on to a Verifier. The Verifier then compares the Evidence against its security policy, and returns an Attestation Result to the Relying Party. The Relying Party then compares the Attestation Result against its own security policy.

The resource access protocol between the Attester and Relying Party includes Evidence rather than an Attestation Result, but that Evidence is not processed by the Relying Party. Since the Evidence is merely forwarded on to a trusted Verifier, any serialization format can be used for Evidence because the Relying Party does not need a parser for it. The only requirement is that the Evidence can be _encapsulated in_ the format required by the resource access protocol between the Attester and Relying Party.

However, like in the Passport model, an Attestation Result is still consumed by the Relying Party and so the serialization format of the Attestation Result is still important. If the Relying Party is a constrained node whose purpose is to serve a given type resource using a standard resource access protocol, it already needs the parser(s) required by that existing protocol. Hence, the ability to let the Relying Party obtain an Attestation Result in the same serialization format allows minimizing the code footprint and attack surface area of the Relying Party, especially if the Relying Party is a constrained node.

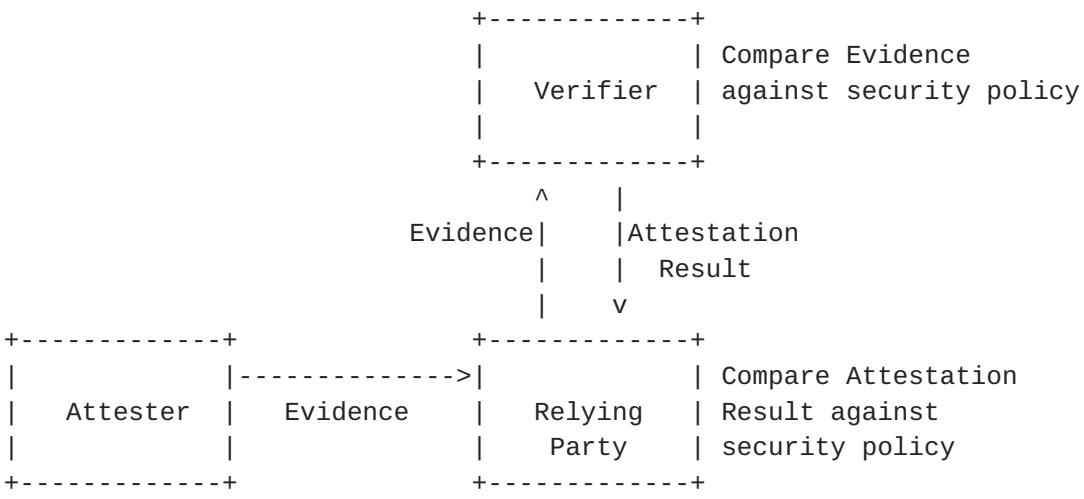

Figure 3: Background-Check Model

The background-check model is so named because it resembles the process typically used for job and loan applications, where a person fills out an application to get a job or a loan from a company, and

the company then does a background check with some other agency that checks credit history, arrest records, etc. and gives back a report on the application that is then used to help determine whether to actually offer the job or loan. Thus, in this analogy, a person asking for a loan is an Attester, the bank is the Relying Party, and a credit report agency is a Verifier.

5.2.1. Variation: Verifying Relying Party

One variation of the background-check model is a "Verifying Relying party", where the Relying Party and the Verifier on the same machine, and so there is no need for a protocol between the two.

5.2.2. Variation: Out-of-Band Evidence Conveyance

Another variation of the background-check model is shown in Figure 4, where the Verifier is still chosen by (and trusted by) the Relying Party, but the Evidence must be passed out-of-band. For example, in step 1, the Attester communicates with the Relying Party, which refers the matter to a Verifier chosen by the Relying Party in step 2. Evidence is then passed to that Verifier in step 3, e.g., either by the Relying Party providing the Attester with information about the Verifier to send Evidence to, or by the Verifier querying the Attester directly, although the latter has the problem that it only works if devices allow unsolicited inbound queries, which may be a security problem in some contexts.

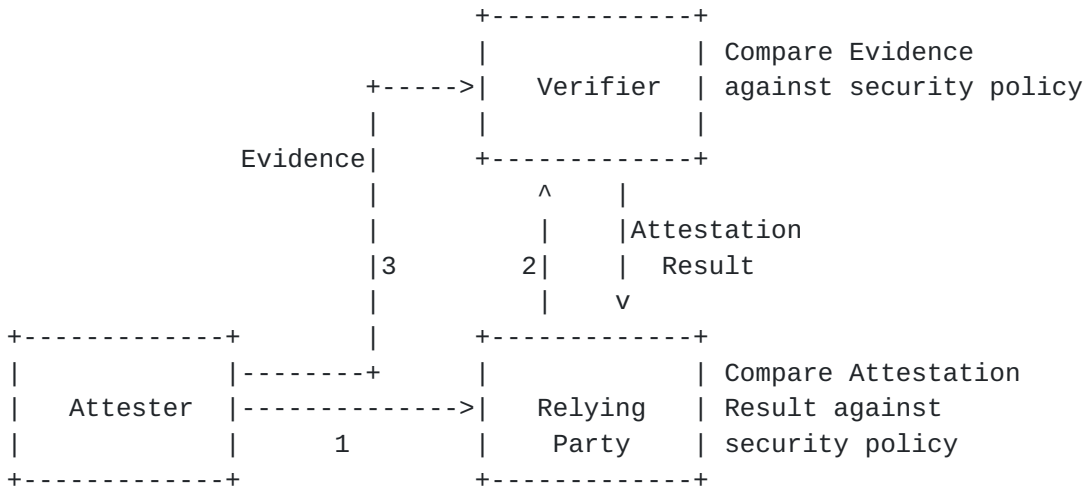

Figure 4: Out-of-Band Evidence Conveyance

5.3. Combinations

The choice of model is generally up to the Relying Party, and the same device may need to attest to different relying parties for different use cases (e.g., a network infrastructure device to gain access to the network, and then a server holding confidential data to get access to that data). As such, both models may simultaneously be in use by the same device.

Figure 5 shows an example of a combination where Relying Party 1 uses the passport model, whereas Relying Party 2 uses an extension of the background-check model. Specifically, in addition to the basic functionality shown in Figure 3, Relying Party 2 actually provides the Attestation Result back to the Attester, allowing the Attester to use it with other Relying Parties. This is the model that the Trusted Application Manager plans to support in the TEEP architecture [I-D.ietf-teep-architecture].

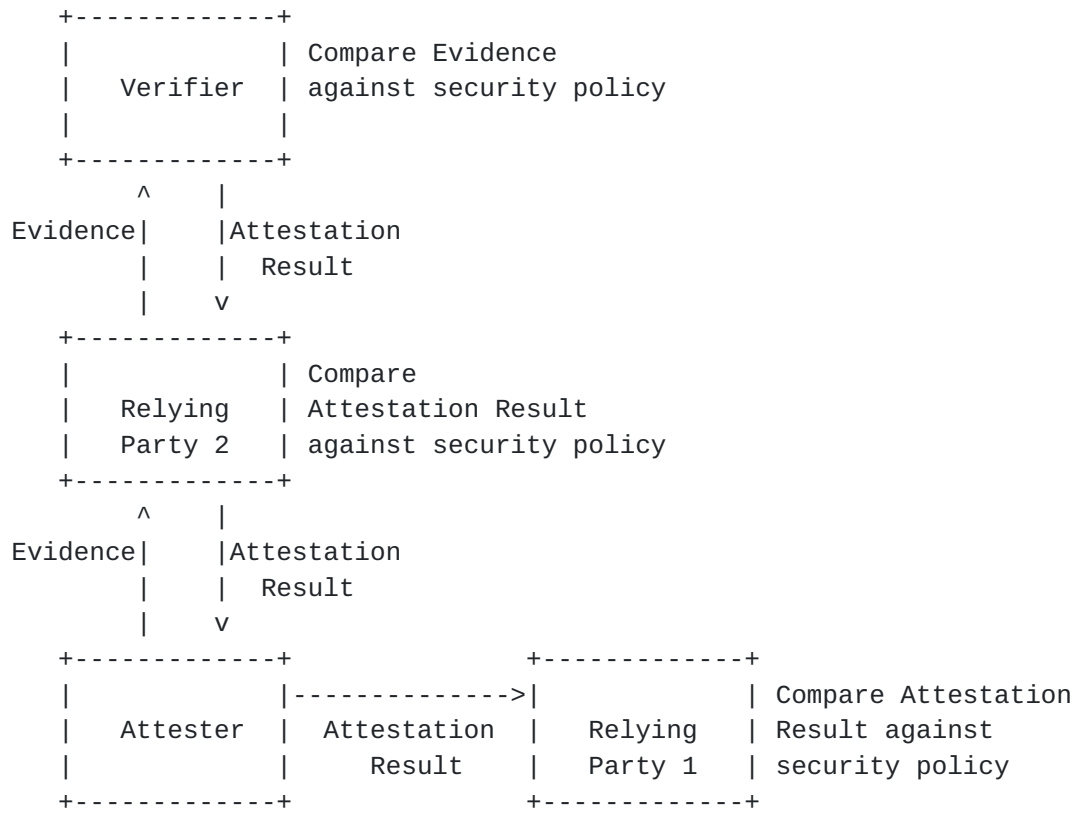

Figure 5: Example Combination

6. Trust Model

The scope of this document is scenarios for which a Relying Party trusts a Verifier that can evaluate the trustworthiness of information about an Attester. Such trust might come by the Relying Party trusting the Verifier (or its public key) directly, or might come by trusting an entity (e.g., a Certificate Authority) that the Verifier has a certificate that chains up to. The Relying Party might implicitly trust a Verifier (such as in the Verifying Relying Party combination). Or, for a stronger level of security, the Relying Party might require that the Verifier itself provide information about itself that the Relying Party can use to evaluate the health of the Verifier before accepting its Attestation Results.

In solutions following the background-check model, the Attester is assumed to trust the Verifier (again, whether directly or indirectly via a Certificate Authority that it trusts), since the Attester relies on an Attestation Result it obtains from the Verifier, in order to access resources.

The Verifier trusts (or more specifically, the Verifier's security policy is written in a way that configures the Verifier to trust) a manufacturer, or the manufacturer's hardware, so as to be able to evaluate the health of that manufacturer's devices. In solutions with weaker security, a Verifier might be configured to implicitly trust firmware or even software (e.g., a hypervisor). That is, it might evaluate the health of an application component, or operating system component or service, under the assumption that information provided about it by the lower-layer hypervisor or firmware is true. A stronger level of security comes when information can be vouched for by hardware or by ROM code, 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.

7. Conceptual Messages

7.1. Evidence

Today, Evidence tends to be highly device-specific, since the information in the evidence often includes vendor-specific information that is necessary to fully describe the manufacturer and model of the device including its security properties, the health of the device, and the level of confidence in the correctness of the information. Evidence is typically signed by the device (whether by hardware, firmware, or software on the device), and evaluating it in isolation would require security policy to be based on device-specific details (e.g., a device public key).

7.2. Endorsements

An Endorsement is a secure statement that some entity (typically a manufacturer) vouches for the integrity of the device's signing capability. For example, if the signing capability is in hardware, then an Endorsement might be a manufacturer certificate that signs a public key whose corresponding private key is only known inside the device's hardware. Thus, when Evidence and such an Endorsement are used together, evaluating them can be done against security policy that may not be specific to the device instance, but merely specific to the manufacturer providing the Endorsement. For example, a security policy might simply check that devices from a given manufacturer have information matching a set of known-good reference values, or a security policy might have a set of more complex logic on how to evaluate the validity of information.

However, while a security policy that treats all devices from a given manufacturer the same may be appropriate for some use cases, it would be inappropriate to use such a security policy as the sole means of authorization for use cases that wish to constrain _which_ compliant devices are considered authorized for some purpose. For example, an enterprise using attestation for Network Endpoint Assessment may not wish to let every healthy laptop from the same manufacturer onto the network, but instead only want to let devices that it legally owns onto the network. Thus, an Endorsement may be helpful information in authenticating information about a device, but is not necessarily sufficient to authorize access to resources which may need device-specific information such as a public key for the device or component or user on the device.

7.3. Attestation Results

Attestation Results may indicate compliance or non-compliance with a Verifier's security policy. A result that indicates non-compliance can be used by an Attester (in the passport model) or a Relying Party (in the background-check model) to indicate that the Attester should not be treated as authorized and may be in need of remediation. In some cases, it may even indicate that the Evidence itself cannot be authenticated as being correct.

An Attestation Result that indicates compliance can be used by a Relying Party to make authorization decisions based on the Relying Party's security policy. The simplest such policy might be to simply authorize any party supplying a compliant Attestation Result signed by a trusted Verifier. A more complex policy might also entail comparing information provided in the result against known-good reference values, or applying more complex logic using such information.

Thus, Attestation Results often need to include detailed information about the Attester, for use by Relying Parties, much like physical passports and drivers licenses include personal information such as name and date of birth. Unlike Evidence, which is often very device- and vendor-specific, Attestation Results can be vendor-neutral if the Verifier has a way to generate vendor-agnostic information based on evaluating vendor-specific information in Evidence. This allows a Relying Party's security policy to be simpler, potentially based on standard ways of expressing the information, while still allowing interoperability with heterogeneous devices.

Finally, whereas Evidence is signed by the device (or indirectly by a manufacturer, if Endorsements are used), Attestation Results are signed by a Verifier, allowing a Relying Party to only need a trust relationship with one entity, rather than a larger set of entities, for purposes of its security policy.

8. Security Considerations

To evaluate the security provided by a particular security policy, it is important to understand the strength of the Root of Trust, e.g., whether it is mutable software, or firmware that is read-only after boot, or immutable hardware/ROM.

It is also important that the security policy was itself obtained securely. As such, if security policy in a Relying Party or Verifier can be configured via a network protocol, the ability to attest to the health of the client providing the security policy needs to be considered.

9. IANA Considerations

This document does not require any actions by IANA.

10. Acknowledgements

Some content in this document came from drafts by Michael Richardson, Henk Birkholz, and Ned Smith, and from the IETF RATS Working Group Charter.

11. Informative References

[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.

[OPCUA] OPC Foundation, "OPC Unified Architecture Specification, Part 2: Security Model, Release 1.03", Global Platform GPD_SPE_009, November 2015, <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-2-security-model/>.

[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>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

Author's Address

Dave Thaler
Microsoft

EMail: dthaler@microsoft.com