RATS Working Group
Internet-Draft
Intended status: Informational
Expires: 8 September 2020

H. Birkholz
Fraunhofer SIT
D. Thaler
Microsoft
M. Richardson
Sandelman Software Works
N. Smith
Intel
W. Pan
Huawei Technologies

7 March 2020

Remote Attestation Procedures Architecture
draft-ietf-rats-architecture-02

Abstract

In network protocol exchanges, it is often the case that one entity (a Relying Party) requires evidence about a remote peer to assess the peer's trustworthiness, and a way to appraise such evidence. The evidence is typically a set of claims about its software and hardware platform. This document describes an architecture for such remote attestation procedures (RATS).

Note to Readers

Discussion of this document takes place on the RATS Working Group mailing list (rats@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/rats/ (https://mailarchive.ietf.org/arch/browse/rats/).

Source for this draft and an issue tracker can be found at https://github.com/ietf-rats-wg/architecture (https://github.com/ietf-rats-wg/architecture).

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 8 September 2020.

Table of Contents

1. Introduction

In Remote Attestation Procedures (RATS), one peer (the "Attester") produces believable information about itself - Evidence - to enable a remote peer (the "Relying Party") to decide whether to consider that Attester a trustworthy peer or not. RATS are facilitated by an additional vital party, the Verifier. The Verifier appraises Evidence via Appraisal Policies and creates the Attestation Results to support Relying Parties in their decision process.

This documents defines a flexible architecture with corresponding roles and their interaction via conceptual messages. Additionally, this document defines a universal set of terms that can be mapped to various existing and emerging Remote Attestation Procedures. Common role compositions and data flows, such as the "Passport Model" and the "Background-Check Model" are illustrated to enable readers of this document to map their current and emerging solutions to the architecture provided and the corresponding terminology defined. A common terminology that provides a well-understood semantic meaning to the concepts, roles, and models in this document is vital to create semantic interoperability between solutions and across different platforms.

Amongst other things, this document is about trust and trustworthiness. Trust is a decision being made. Trustworthiness is a quality that is assessed via evidence created. This is a subtle difference and being familiar with the difference is crucial for using this document. Additionally, the concepts of freshness and trust relationships with respect to RATS are elaborated on to enable

implementers in order to choose appropriate solutions to compose their Remote Attestation Procedures.

2. Terminology

This document uses the following terms.

Appraisal Policy for Evidence: A set of rules that direct how a Verifier evaluates the validity of information about an Attester. Compare /security policy/ in [RFC4949]

Appraisal Policy for Attestation Result: A set of rules that direct how a Relying Party uses the Attestation Results regarding an Attester generated by the Verifiers. Compare /security policy/ in [RFC4949]

Attestation Result: The output 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 appraised in order to determine whether the entity is considered trustworthy, such as when deciding whether the entity is authorized to perform some operation

Endorsement: A secure statement that some entity (typically a manufacturer) vouches for the integrity of an Attester's signing capability

Endorser: An entity that creates Endorsements that can be used to help to appraise the trustworthiness of Attesters

Evidence: A set of information about an Attester that is to be appraised 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]

Relying Party Owner: An entity, such as an administrator, that is

authorized to configure Appraisal Policy for Attestation Results in a Relying Party.

Verifier: An entity that appraises the validity of Evidence about an Attester

Verifier Owner: An entity, such as an administrator, that is authorized to configure Appraisal Policy for Evidence in a Verifier

3. Reference Use Cases

This section covers a number of representative use cases for remote attestation, independent of specific solutions. 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.

Each use case includes a description, and a summary of what an Attester and a Relying Party refer to in the use case.

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. Remote attestation is desired to prevent vulnerable or compromised devices from getting access to the network and potentially harming others.

Typically, solutions start with a specific component (called a "Root of Trust") that provides device identity and protected storage for measurements. These components perform a series of measurements, and express this with Evidence as to the hardware and firmware/software that is running.

FIXME from Henk: Measurements at early stages of

Layered Attestation are NOT evidence yet.
This text does not cover that yet

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 remote 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, remote 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 assess 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

remote attestation to a 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. Architectural Overview

Figure 1 depicts the data that flows between different roles, independent of protocol or use case.

********** ********** **********

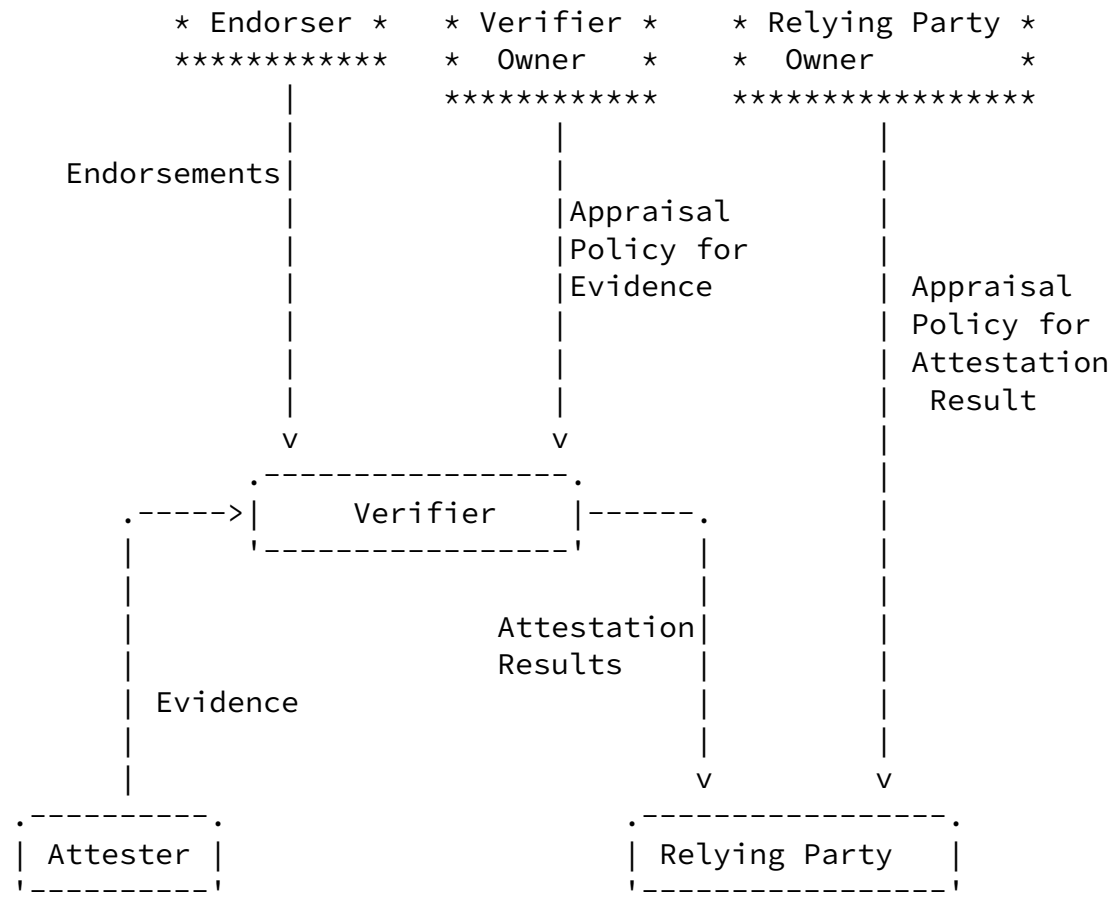


Figure 1: Conceptual Data Flow

An Attester creates Evidence that is conveyed to a Verifier.

The Verifier uses the Evidence, and any Endorsements from Endorsers, by applying an Evidence Appraisal Policy to assess the trustworthiness of the Attester, and generates Attestation Results for use by Relying Parties. The Evidence Appraisal Policy might be obtained from an Endorser along with the Endorsements, or might be obtained via some other mechanism such as being configured in the Verifier by an administrator.

The Relying Party uses Attestation Results by applying its own Appraisal Policy to make application-specific decisions such as authorization decisions. The Attestation Result Appraisal Policy might, for example, be configured in the Relying Party by an administrator.

4.1. Two Types of Environments of an Attester

An Attester consists of at least one Attesting Environment and at least one Target Environment. In some implementations, the Attesting and Target Environments might be combined. Other implementations might have multiple Attesting and Target Environments, such as in the examples described in more detail in Section 4.2 and Section 4.3. Other examples may exist, and the examples discussed could even be combined into even more complex implementations.

Claims are collected from Target Environments. That is, Attesting Environments collect the raw values and the information to be represented in claims, such as by doing some measurement of a Target Environment's code, memory, and/or registers. Attesting Environments then format the claims appropriately, and typically use key material and cryptographic functions, such as signing or cipher algorithms, to create Evidence. Examples of environments that can be used as Attesting Environments include Trusted Execution Environments (TEE), embedded Secure Elements (eSE), or Hardware Security Modules (HSM).

4.2. Layered Attestation Procedures

By definition, the Attester role takes on the duty to create Evidence. The fact that an Attester role is composed of several types of environments that can be nested or staged adds complexity to the architectural layout of how an Attester - in itself - is composed and therefore has to conduct the Claims collection in order to create believable Attestation Evidence. The following example is intended to illustrate this composition:

A very common example is elaborated on to illustrate Layered Attestation.

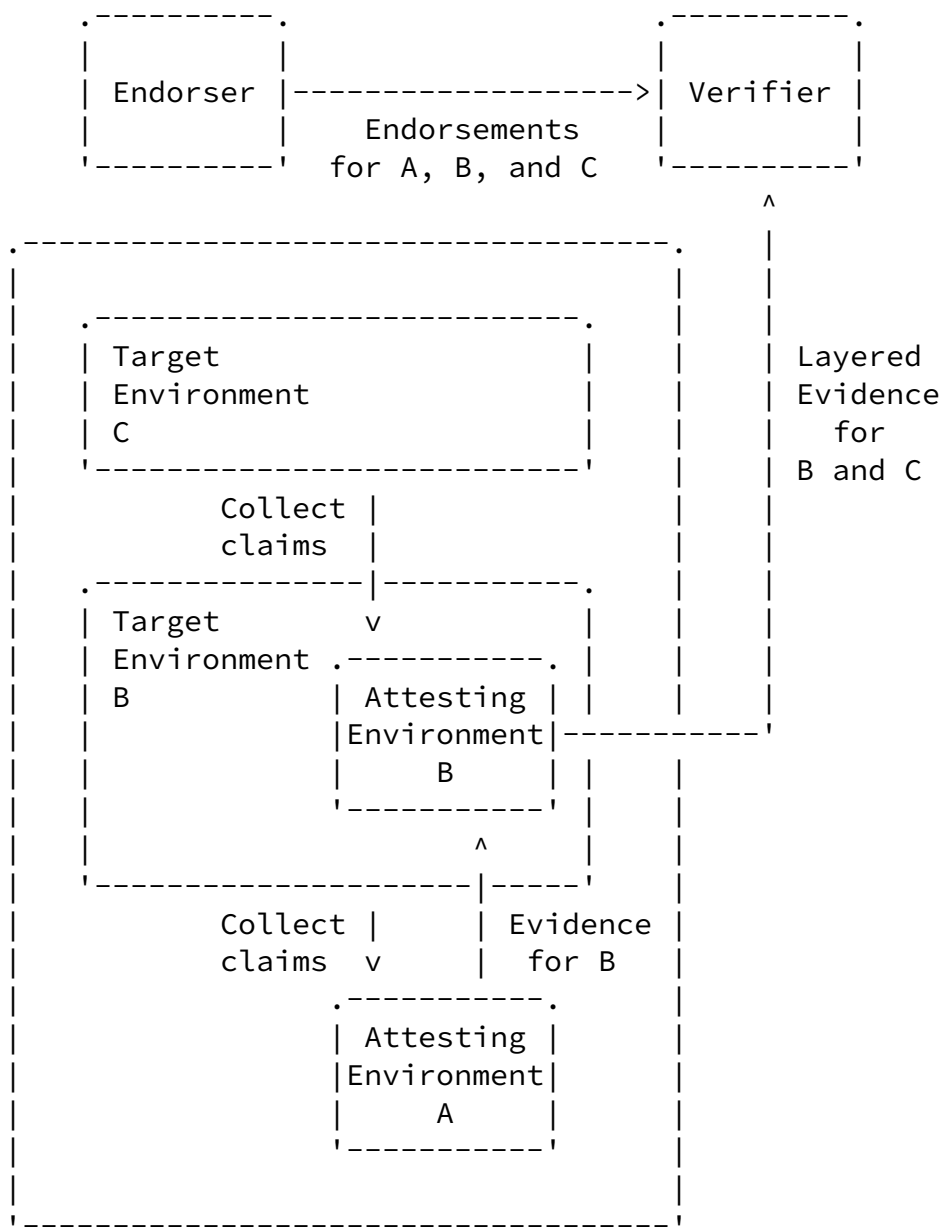


Figure 2: Layered Attester

The very first Attesting Environment has to ensure the integrity of the (U)EFI / BIOS / Firmware that initially boots up a composite device (e.g., a cell phone).

Henk: we are looking for a better term than UEFI/BIOS/Firmware

These Claims have to be measured securely. At this stage of the boot-cycle of a composite device, the Claims collected typically cannot be composed into Evidence.

The very first Attesting Environment in this example can be a hardware component that is a Static Code Root of Trust. As in any

other scenario, this hardware component is the first Attesting Environment. It collects a rather concise number of Claims about the Target Environment. The Target Environment in this example is the (U)EFI / BIOS / Firmware After the boot sequence started, the Target Environment conducts the most important and defining feature of Layered Attestation: The successfully measured environment that is the (U)EFI / BIOS / Firmware now becomes the Attesting Environment. Analogously, the Attesting Environment hands off its duty to one of its Target Environments. This procedure in Layered Attestation is called Staging.

Now, the duties have been transferred and Layered Attestation takes place. The initial Attesting Environment relinquishes its duties to the Target Environment. It is important to note that the new Attesting Environment cannot alter the content about its own measurements. If the Attesting Environment would be able to do that, Layered Attestation would become unfeasible.

In this example the duty of being the Attesting Environment is now taken over by the (U)EFI / BIOS / Firmware that was the Attested Environment before. This transfer of duty is the essential part of Layered Attestation. The (U)EFI / BIOS / Firmware now is the Attesting Environment. The next Target Environment is, in this example, a bootloader. There are potentially multiple kernels to boot, the decision is up to the bootloader. Only a bootloader with intact integrity will make an appropriate decision. Therefore, Claims about the integrity of a bootloader are now collected by the freshly appointed Attesting Environment that is the (U)EFI / BIOS / Firmware. Collected Claims have to be stored by the current Attesting Environment in a similar shielded and secured manner, so that the next Attesting Environment is not capable of altering the collection of claims stored.

Continuing with this example, the bootloader is now in charge of collecting Claims about the next execution environment. The next execution environment in this example is the kernel to be booted up. Analogously, the next transfer of duties in this Layered Attestation example occurs: The duty of being an Attesting Environment is transferred to a successfully measured kernel. In this sequence, the kernel is now collecting additional Claims and is storing them in a secure and shielded manner.

> [Henk: we might have to define what successful means in this example and beyond]

The essence of this example is a cascade of staged boot environments. Each environment (after the initial one that is a root-of-trust) has the duty of measuring its next environment before it is started.

Therefore, creating a layered boot sequence and correspondingly enabling Layered Attestation.

4.3. Composite Device

A Composite Device is an entity composed of multiple sub-entities such that its trustworthiness has to be determined by the appraisal of all these sub-entities.

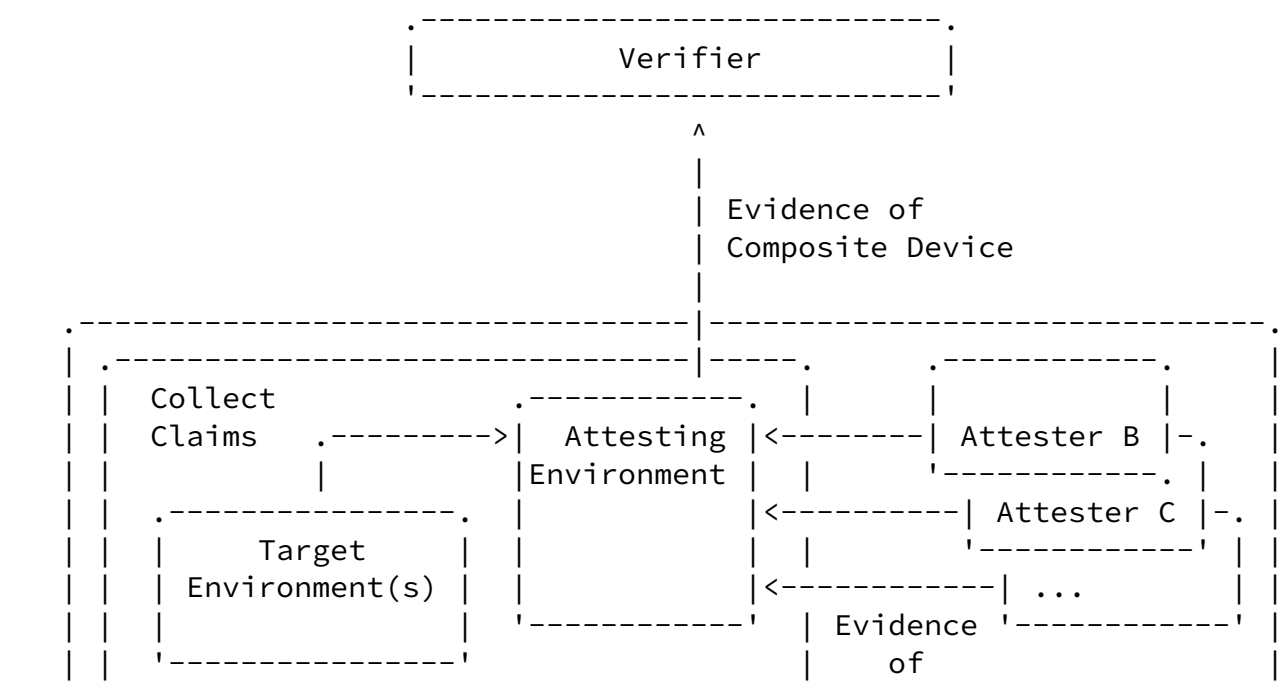
Each sub-entity has at least one Attesting Environment collecting the claims from at least one Target Environment, then this sub-entity generates Evidence about its trustworthiness. Therefore each sub-entity can be called an Attester. Among all the Attesters, there may be only some which have the ability to communicate with the Verifier while others do not.

For example, a carrier-grade router is consists of a chassis and multiple slots. The trustworthiness of the router depends on all its slots' trustworthiness. Each slot has an Attesting Environment such as a TEE collecting the claims of its boot process, after which it generates Evidence from the claims. Among these slots, only a main slot can communicate with the Verifier while other slots cannot. But other slots can communicate with the main slot by the links between them inside the router. So the main slot collects the Evidence of other slots, produces the final Evidence of the whole router and

conveys the final Evidence to the Verifier. Therefore the router is a Composite Device, each slot is an Attester, and the main slot is the transiting Attester.

Another example is a multi-chassis router composed of multiple single carrier-grade routers. The multi-chassis router provides higher throughput by interconnecting multiple routers and can be logically treated as one router for simpler management. Among these routers, there is only one main router that connects to the Verifier. Other routers are only connected to the main router by the network cables, and therefore they are managed and appraised via this main router's help. So, in this case, the multi-chassis router is the Composite Device, each router is an Attester and the main router is the lead Attester.

Figure 3 depicts the conceptual data flow for a Composite Device.



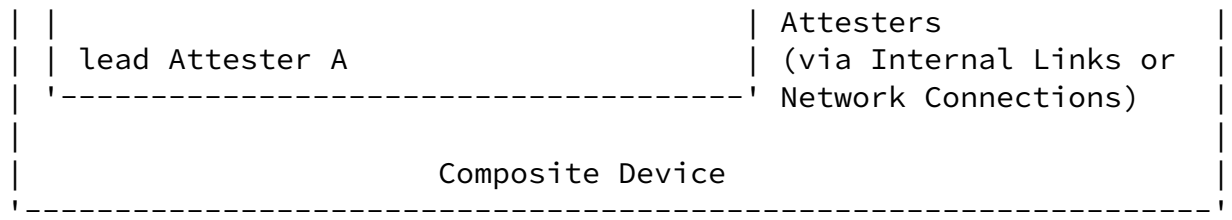


Figure 3: Conceptual Data Flow for a Composite Device

In the Composite Device, each Attester generates its own Evidence by its Attesting Environment(s) collecting the claims from its Target Environment(s). The lead Attester collects the Evidence of all other Attesters and then generates the Evidence of the whole Composite Attester.

An entity can take on multiple RATS roles (e.g., Attester, Verifier, Relying Party, etc.) at the same time. The combination of roles can be arbitrary. For example, in this Composite Device scenario, the entity inside the lead Attester can also take on the role of a Verifier, and the outside entity of Verifier can take on the role of a Relying Party. After collecting the Evidence of other Attesters, this inside Verifier verifies them using Endorsements and Appraisal Policies (obtained the same way as any other Verifier), to generate Attestation Results. The inside Verifier then sends the Attestation Results of other Attesters, whether in the same conveyance protocol as the Evidence or not, to the outside Verifier.

In this situation, the trust model described in Section 6 is also suitable for this inside Verifier.

5. Topological Models

There are multiple possible models for communication between an Attester, a Verifier, and a Relying Party. This section includes some reference models, but this is not intended to be a restrictive list, and other variations may exist.

5.1. Passport Model

In this model, an Attester sends Evidence to a Verifier, which compares the Evidence against its Appraisal 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 Appraisal Policy.

There are three ways in which the process may fail. First, the Verifier may refuse to issue the Attestation Result due to some error in processing, or some missing input to the Verifier. The second way in which the process may fail is when the resulting Result is examined by the Relying Party, and based upon the Appraisal Policy, the result does not pass the policy. The third way is when the Verifier is unreachable.

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 remote attestation protocol.

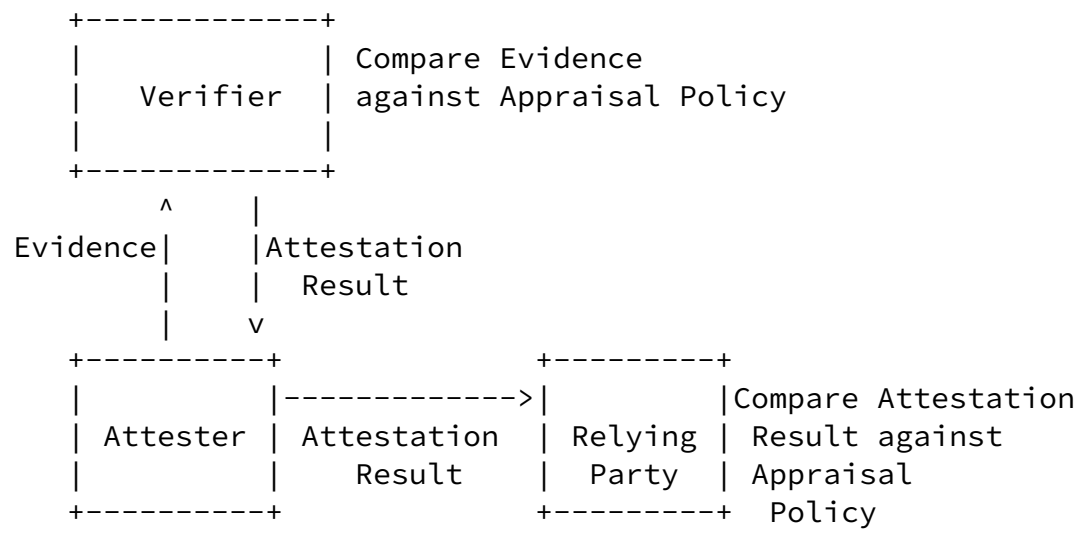


Figure 4: Passport Model

The passport model is so named because of its resemblance to how nations issue passports to their citizens. The nature of the Evidence that an individual needs to provide to its local authority is specific to the country involved. The citizen retains control of

the resulting passport document and presents it to other entities when it needs to assert a citizenship or identity claim, such as an airport immigration desk. The passport is considered sufficient because it vouches for the citizenship and identity claims, and it is issued by a trusted authority. Thus, in this immigration desk analogy, the passport issuing agency is a Verifier, the passport is an Attestation Result, and the immigration desk 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 Appraisal Policy, and returns an Attestation Result to the Relying Party. The Relying Party then compares the Attestation Result against its own appraisal 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.

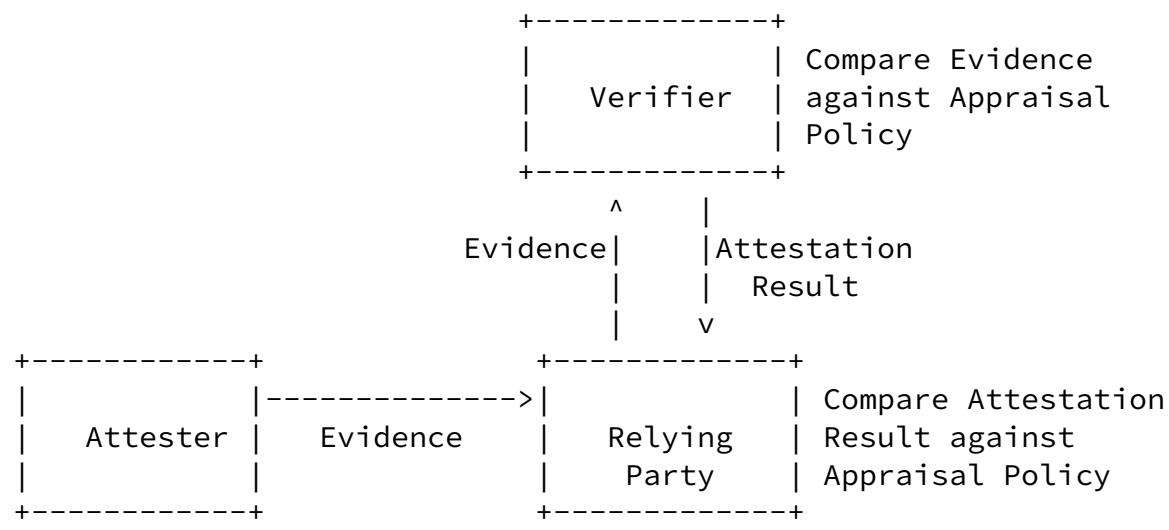


Figure 5: Background-Check Model

The background-check model is so named 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. Thus, in this analogy, a prospective volunteer is an Attester, the organization is the Relying Party, and a former employer or government agency that issues a report is a Verifier.

5.3. Combinations

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

It is also worth pointing out that the choice of model is generally up to the Relying Party, and the same device may need to create Evidence for different Relying Parties and 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 6 shows another 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 5, 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].

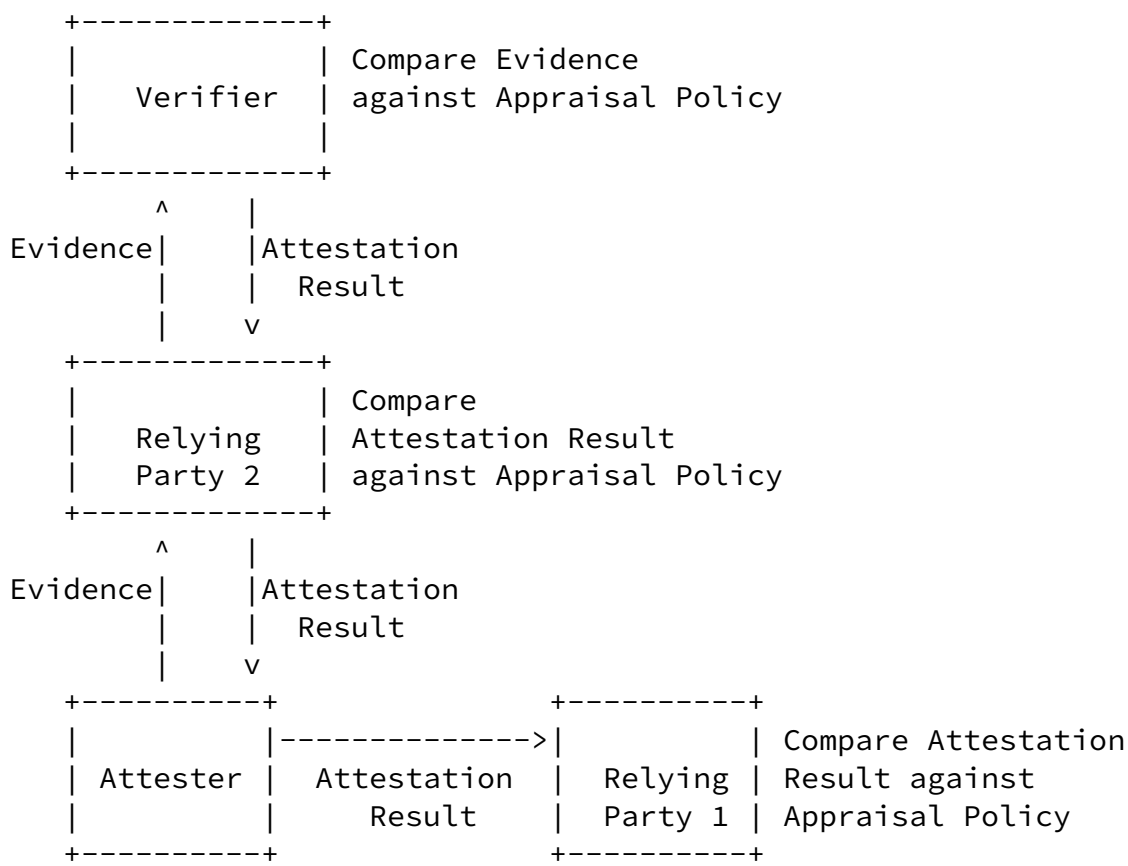


Figure 6: Example Combination

6. Trust Model

The scope of this document is scenarios for which a Relying Party trusts a Verifier that can appraise 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 is in the Verifier's certificate chain. 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 assess the trustworthiness of the Verifier before accepting its Attestation Results.

The Endorser and Verifier Owner may need to trust the Verifier before giving the Endorsement and Appraisal Policy to it. Such trust can also be established directly or indirectly, implicitly or explicitly. One explicit way to establish such trust may be the Verifier first

acts as an Attester and creates Evidence about itself to be consumed by the Endorser and/or Verifier Owner as the Relying Parties. If it is accepted as trustworthy, then they can provide Endorsements and Appraisal Policies that enable it to act as a Verifier.

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 appraise the trustworthiness 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 appraise the trustworthiness 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.

In some scenarios, Evidence might contain sensitive information such as Personally Identifiable Information. Thus, an Attester must trust entities to which it sends Evidence, to not reveal sensitive data to unauthorized parties. The Verifier might share this information with other authorized parties, according rules that it controls. In the background-check model, this Evidence may also be revealed to Relying Party(s).

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 its appraisal in isolation would require Appraisal 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 (e.g., 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, an appraisal procedure can be conducted based on Appraisal Policies that may not be specific to the device instance, but merely specific to the manufacturer providing the Endorsement. For example, an Appraisal Policy might simply check that devices from a given manufacturer have information matching a set of known-good reference values, or an Appraisal Policy might have a set of more complex logic on how to appraise the validity of information.

However, while an Appraisal 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 an Appraisal 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 remote 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 Appraisal 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 Appraisal 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 on 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 the appraisal of vendor-specific information in Evidence. This allows a Relying Party's Appraisal 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 Appraisal Policy.

8. Claims Encoding Formats

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

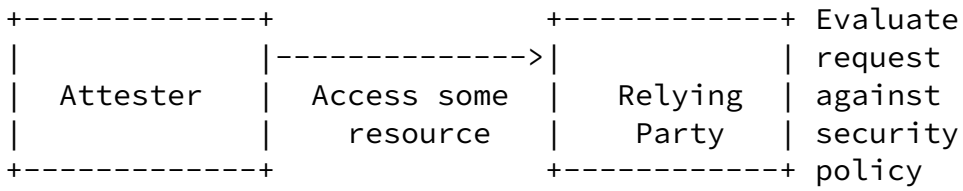


Figure 7: Typical Resource Access

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.

To enable remote attestation to be added to existing protocols, enabling a higher level of assurance against malware for example, it is important that information needed for appraising 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, remote 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 remote attestation related information in an encoding-agnostic way, and allowing multiple encoding formats (CWT, JWT, X.509, etc.) that encode the same information into the claims format needed by the Relying Party.

The following diagram illustrates that Evidence and Attestation Results might each have multiple possible encoding formats, so that they can be conveyed by various existing protocols. It also motivates why the Verifier might also be responsible for accepting Evidence that encodes claims in one format, while issuing Attestation Results that encode claims in a different format.

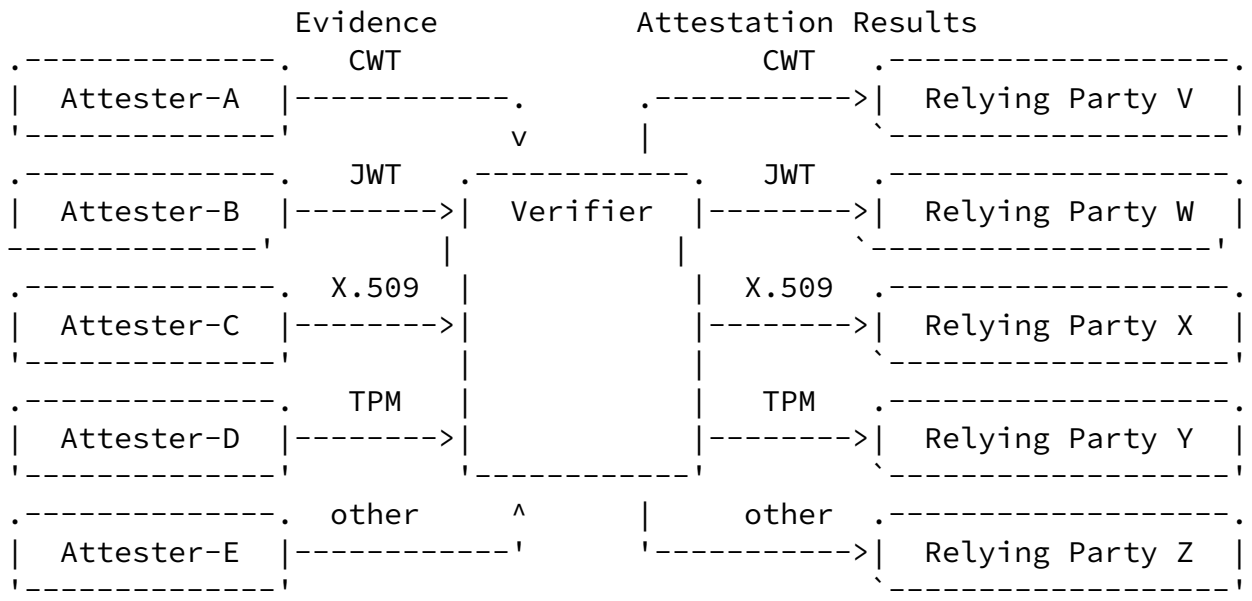


Figure 8: Multiple Attesters and Relying Parties with Different Formats

9. Freshness

It is important to prevent replay attacks where an attacker replays old Evidence or an old Attestation Result that is no longer correct. To do so, some mechanism of ensuring that the Evidence and

Attestation Result are fresh, meaning that there is some degree of assurance that they still reflect the latest state of the Attester, and that any Attestation Result was generated using the latest Appraisal Policy for Evidence. There is, however, always a race condition possible in that the state of the Attester, and the Appraisal Policy for Evidence, might change immediately after the Evidence or Attestation Result was generated. The goal is merely to narrow the time window to something the Verifier (for Evidence) or Relying Party (for an Attestation Result) is willing to accept.

There are two common approaches to providing some assurance of freshness. The first approach is that a nonce is generated by a remote entity (e.g., the Verifier for Evidence, or the Relying Party for an Attestation Result), and the nonce is then signed and included along with the claims in the Evidence or Attestation Result, so that the remote entity knows that the claims were signed after the nonce was generated.

A second approach is to rely on synchronized clocks, and include a signed timestamp (e.g., using [I-D.birkholz-rats-tuda]) along with the claims in the Evidence or Attestation Result, so that the remote entity knows that the claims were signed at that time, as long as it has some assurance that the timestamp is correct. This typically requires additional claims about the signer's time synchronization mechanism in order to provide such assurance.

In either approach, it is important to note that the actual values in claims might have been generated long before the claims are signed. If so, it is the signer's responsibility to ensure that the values are still correct when they are signed. For example, values might have been generated at boot, and then used in claims as long as the signer can guarantee that they cannot have changed since boot.

10. 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 information about the type of the device and the firmware/software that the device is running. This information might be particularly interesting to many attackers. For example, knowing that a device is running a weak version of firmware provides a way to aim attacks better.

Evidence and Attestation Results data structures are expected to support integrity protection encoding (e.g., COSE, JOSE, X.509) and optionally might support confidentiality protection (e.g., COSE, JOSE). Therefore, if confidentiality protection is omitted or unavailable, the protocols that convey Evidence or Attestation Results are responsible for detailing what kinds of information are disclosed, and to whom they are exposed.

11. Security Considerations

Any solution that conveys information used for security purposes, whether such information is in the form of Evidence, Attestation Results, Endorsements, or Appraisal Policy, needs to support end-to-end integrity protection and replay attack prevention, and often also needs to support additional security protections. For example, additional means of authentication, confidentiality, integrity, replay, denial of service and privacy protection are needed in many use cases. Section 9 discusses ways in which freshness can be used in this architecture to protect against replay attacks.

To assess the security provided by a particular Appraisal 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 Appraisal Policy was itself obtained securely. As such, if Appraisal Policies for a Relying Party or for a Verifier can be configured via a network protocol, the ability to create Evidence about the integrity of the entity providing the Appraisal Policy needs to be considered.

The security of conveyed information may be applied at different layers, whether by a conveyance protocol, or an information encoding format. This architecture expects attestation messages (i.e., Evidence, Attestation Results, Endorsements and Policies) are end-to-end protected based on the role interaction context. For example, if an Attester produces Evidence that is relayed through some other entity that doesn't implement the Attester or the intended Verifier roles, then the relaying entity should not expect to have access to the Evidence.

12. IANA Considerations

This document does not require any actions by IANA.

13. Acknowledgments

Special thanks go to Joerg Borchert, Nancy Cam-Winget, Jessica Fitzgerald-McKay, Thomas Fossati, Diego Lopez, Laurence Lundblade, Wei Pan, Paul Rowe, Hannes Tschofenig, Frank Xia, and David Wooten.

14. Contributors

Thomas Hardjono created older versions of the terminology section in collaboration with Ned Smith. Eric Voit provided the conceptual separation between Attestation Provision Flows and Attestation Evidence Flows. Monty Wisemen created the content structure of the first three architecture drafts. Carsten Bormann provided many of the motivational building blocks with respect to the Internet Threat Model.

15. References

15.1. Normative References

[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>.

15.2. Informative References

[I-D.birkholz-rats-tuda]
Fuchs, A., Birkholz, H., McDonald, I., and C. Bormann, "Time-Based Uni-Directional Attestation", Work in Progress, Internet-Draft, draft-birkholz-rats-tuda-01, 11 September 2019, <http://www.ietf.org/internet-drafts/draft-birkholz-rats-tuda-01.txt>.

[I-D.ietf-teep-architecture]
Pei, M., Tschofenig, H., Thaler, D., and D. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", Work in Progress, Internet-Draft, draft-ietf-teep-architecture-06, 8 February 2020,

<http://www.ietf.org/internet-drafts/draft-ietf-teep-architecture-06.txt>.

[OPCUA] OPC Foundation, "OPC Unified Architecture Specification, Part 2: Security Model, Release 1.03", OPC 10000-2 , 25 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>.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
64295 Darmstadt
Germany

Email: henk.birkholz@sit.fraunhofer.de

Dave Thaler
Microsoft
United States of America

Email: dthaler@microsoft.com

Michael Richardson
Sandelman Software Works
Canada

Email: mcr+ietf@sandelman.ca

Ned Smith
Intel Corporation
United States of America

Email: ned.smith@intel.com

Wei Pan
Huawei Technologies

Email: william.panwei@huawei.com