

RATS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 1, 2020

G. Fedorkow, Ed.  
Juniper Networks, Inc.  
J. Fitzgerald-McKay  
National Security Agency  
June 30, 2019

Network Device Attestation Workflow  
draft-fedorkow-rats-network-device-attestation-00

## Abstract

This document describes a workflow for network device attestation.

## 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 January 1, 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.

Internet-Draft

Network Device Attestation Workflow

June 2019

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">1.1.</a>	<a href="#">Requirements Language</a>	<a href="#">3</a>
<a href="#">1.2.</a>	<a href="#">Goals</a>	<a href="#">4</a>
<a href="#">1.3.</a>	<a href="#">Problem Description</a>	<a href="#">4</a>
<a href="#">1.4.</a>	<a href="#">Solution Requirements</a>	<a href="#">6</a>
<a href="#">1.5.</a>	<a href="#">Scope</a>	<a href="#">7</a>
<a href="#">1.5.1.</a>	<a href="#">Out of Scope</a>	<a href="#">8</a>
<a href="#">1.5.2.</a>	<a href="#">Why Remote Integrity Verification?</a>	<a href="#">8</a>
<a href="#">1.5.3.</a>	<a href="#">Network Device Attestation Challenges</a>	<a href="#">8</a>
<a href="#">1.5.4.</a>	<a href="#">Why is OS Attestation Different?</a>	<a href="#">9</a>
<a href="#">2.</a>	<a href="#">Solution Outline</a>	<a href="#">10</a>
<a href="#">2.1.</a>	<a href="#">2.1 RIV Software Configuration Attestation using TPM</a>	<a href="#">10</a>
<a href="#">2.2.</a>	<a href="#">RIV Keying</a>	<a href="#">11</a>
<a href="#">2.3.</a>	<a href="#">RIV Information Flow</a>	<a href="#">12</a>
<a href="#">2.4.</a>	<a href="#">RIV Simplifying Assumptions</a>	<a href="#">13</a>
<a href="#">2.4.1.</a>	<a href="#">DevID Alternatives</a>	<a href="#">14</a>
<a href="#">2.4.2.</a>	<a href="#">Additional Attestation of Platform Characteristics</a>	<a href="#">14</a>
<a href="#">2.4.3.</a>	<a href="#">Root of Trust for Measurement</a>	<a href="#">15</a>
<a href="#">2.4.4.</a>	<a href="#">Reference Integrity Measurements (RIMs)</a>	<a href="#">15</a>
<a href="#">2.4.5.</a>	<a href="#">Attestation Logs</a>	<a href="#">16</a>
<a href="#">3.</a>	<a href="#">Standards Components</a>	<a href="#">17</a>
<a href="#">3.1.</a>	<a href="#">Reference Models</a>	<a href="#">17</a>
<a href="#">3.1.1.</a>	<a href="#">IETF Reference Model for Challenge-Response Remote Attestation</a>	<a href="#">17</a>
<a href="#">3.2.</a>	<a href="#">RIV Workflow</a>	<a href="#">18</a>
<a href="#">3.3.</a>	<a href="#">Layering Model for Network Equipment Attester and Verifier</a>	<a href="#">20</a>
<a href="#">4.</a>	<a href="#">Conclusion</a>	<a href="#">21</a>
<a href="#">5.</a>	<a href="#">Appendix</a>	<a href="#">21</a>
<a href="#">5.1.</a>	<a href="#">Implementation Notes</a>	<a href="#">22</a>
<a href="#">5.2.</a>	<a href="#">Comparison with TCG PTS / IETF NEA</a>	<a href="#">24</a>
<a href="#">6.</a>	<a href="#">IANA Considerations</a>	<a href="#">25</a>
<a href="#">7.</a>	<a href="#">Security Considerations</a>	<a href="#">26</a>
<a href="#">8.</a>	<a href="#">Informative References</a>	<a href="#">26</a>
	<a href="#">Authors' Addresses</a>	<a href="#">29</a>

[1.](#) Introduction

There are many components to consider in fielding a trusted computing device, from operating systems to applications. Part of that is a trusted supply chain, where manufacturers can certify that the

product they intended to build is actually the one that was installed at a customer's site.

Attestation is defined here as the process of creating, conveying and appraising assertions about Platform trustworthiness characteristics,

including Roots of Trust, supply chain trust, identity, platform provenance, shielded locations, protected capabilities, software configuration, hardware configuration, platform composition, compliance to test suites, functional and assurance evaluations, etc.

The supply chain itself has many elements, from validating suppliers of electronic components, to ensuring that shipping procedures protect against tampering through many stages of distribution and warehousing. One element that helps maintain the integrity of the supply chain after manufacturing is Attestation.

Within the Trusted Computing Group context, attestation is the process by which an independent Verifier can obtain cryptographic proof as to the identity of the device in question, evidence of the integrity of software loaded on that device when it started up, and then verify that what's there is what's supposed to be there. For networking equipment, a verifier capability can be embedded in a Network Management Station (NMS), a posture collection server, or other network analytics tool (such as a software asset management solution, or a threat detection and mitigation tool, etc.). While informally referred to as attestation, this document focuses on a subset defined here as Remote Integrity Verification (RIV). RIV takes a network equipment centric perspective that includes a set of protocols and procedures for determining whether a particular device was launched with untampered software, starting from Roots of Trust. While there are many ways to accomplish attestation, RIV sets out a specific set of protocols and tools that work in environments commonly found in Networking Equipment. RIV does not cover other platform characteristics that could be attested, although it does provide evidence of a secure infrastructure to increase the level of trust in other platform characteristics attested by other means.

This profile outlines the RIV problem, and then identifies components that are necessary to get the complete attestation procedure working in a scalable solution using commercial products.

This document focuses primarily on software integrity verification using the Trusted Platform Module (TPM) as a root of trust.

### [1.1.](#) Requirements Language

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.

### [1.2.](#) Goals

The RIV attestation workflow outlined in this document is intended to meet the following high-level goals:

- o Provable Device Identity - The ability to identify a device using a cryptographic identifier is a critical prerequisite to software inventory attestation.
- o Software Inventory - A key goal is to identify the software release installed on the device, and to provide evidence of its integrity.
- o Verification - Verification of software and configuration of the device shows that the software that's supposed to be installed on there actually has been launched, without unauthorized modification.

This document itself is non-normative; the document does not define protocols, but rather identifies protocols that can be used together to achieve the goals above, and in some cases, highlights gaps in existing protocols.

### [1.3.](#) Problem Description

RIV is a procedure that assures a network operator that the equipment on their network can be reliably identified, and that untampered software of a known version is installed on each endpoint. In this context, endpoint might include the conventional endpoints like

servers and laptops, but also network equipment itself, such as routers, switches and firewalls.

RIV can be viewed as a link in a trusted supply chain, and includes three major processes:

- o Creation of Evidence is the process whereby an endpoint generates cryptographic proof (evidence) of claims about platform properties. In particular, the platform identity and its software configuration are of critical importance.

\* Platform Identity refers to the mechanism assuring the attestation verifier (typically a network administrator) that the equipment on their network can be reliably identified, and that its manufacturer is certified by a trusted authority. This certification provides the verifier with assurance that the Root of Trust elements of the device were verified by the manufacturer before the device was shipped.

\* Software used to boot a platform can be described as a chain of measurements, starting from a Root of Trust for Measurement, that normally ends when the system software is loaded. Measurement records the identity, integrity and version of each software component registered with the TPM, so that the subsequent appraisal stage can determine whether the software installed is authentic and free of tampering. Clearly the second part of the problem, attesting the state of mutable components of a given device, is of little value without the first part, reliable identification of the device in question. By the same token, unambiguous identity of a device is necessary, but is insufficient to assure the operator of the provenance of the device through the supply chain, or that the device is configured to behave properly.

- o Conveyance of Evidence is the process of reliably transporting evidence from an endpoint to an appraiser/verifier, e.g. a management station. The transport is typically carried out via a management network. The channel must provide integrity and authenticity, and, in some use cases, may also require confidentiality.
- o Appraisal of Evidence is the process of verifying the evidence received by a verifier/appraiser from a device, and using verified evidence to inform decision making. In this context, verification means comparing the device characteristics reported as evidence with the configuration expected by the system administrator. This step can work only when there is a way to express what should be there, often referred to as golden measurements, or Reference Integrity Measurements, representing the intended configured state of an endpoint.

As a part of a trusted supply chain, RIV attestation provides two important benefits:

- o Platform Identity is the mechanism providing trusted identity can reassure network managers that the specific devices they ordered from authorized manufacturers for attachment to their network are the ones that were installed, and that they continue to be present in their network. As part of the mechanism for Platform Identity,

cryptographic proof of the identity of the manufacturer is also provided.

- o Software Configuration is the mechanism that reports the state of mutable components on the device can assure network managers that they have known, untampered software configured to run in their network.

An implementation of RIV requires three technologies

1. Identity: Platform identity can be based on IEEE 802.1AR Device Identity [[IEEE-802-1AR](#)], coupled with careful supply-chain management by the manufacturer. The DevID certificate contains a statement by the manufacturer that establishes the provenance of the device as it left the factory. Some applications with a

more-complex post-manufacture supply chain (e.g. Value Added Resellers), or with privacy concerns, may want to use an alternate mechanism for platform authentication based on TCG Platform Certificates [[Platform-Certificates](#)].

2. Platform Attestation provides evidence of configuration of software elements throughout the product lifecycle. This form of attestation can be implemented with TPM PCR, Quote and log mechanisms, which provide an authenticated mechanism to report what software actually starts up on the device each time it reboots. Note that the TPM requires separate keys for identity (DevID) and attestation (PCR Quotes0) (see [Section 2.2](#)).
3. Reference Integrity Measurements must be conveyed from the software authority (often the manufacturer for embedded systems) to the system in which verification will take place

Network operators benefit from a trustworthy attestation mechanism that provides assurance that their comprises authentic equipment, and has loaded software free of known vulnerabilities and unauthorized tampering.

#### [1.4](#). Solution Requirements

An Attestation solution must meet a number of requirements to make it simple to deploy at scale.

1. Easy to Use - This solution should work "out of the box" as far as possible, that is, with the fewest possible steps needed at the end-user's site. Eliminate complicated databases or provisioning steps that would have to be executed by the owner of a new device. Network equipment is often required to "self-configure", to reliably reach out without manual intervention to

prove its identity and operating posture, then download its own configuration. See [[RFC8572](#)] for an example of Secure Zero Touch Provisioning.

2. Multi-Vendor - This solution should identify standards-based interfaces that allow attestation to work with attestation-capable devices and verifiers supplied by different vendors in one network.

3. Scalable - The solution must not depend on choke points that limit the number of endpoints that could be evaluated in one network domain.
4. Extensible - A network equipment attestation solution needs to expand over time as new features are added. The solution must allow new features to be added easily, providing for a smooth transition and allowing newer and older architectural components to continue to work together. Further, a network equipment attestation solution and the specifications referenced here must define safe extensibility mechanisms that enable innovation without breaking interoperability.
5. Efficient - A network equipment attestation solution should, to the greatest extent feasible, continuously monitor the health and posture status of network devices. Posture measurements should be updated in real-time as changes to device posture occur and should be published to remote integrity validators. Validation reports should also be shared with their relying parties (for example, network administrators, or network analytics that rely on these reports for posture assessment) as soon as they are available.

#### [1.5.](#) Scope

This document includes a number of assumptions to limit the scope:

- o This solution is for use in non-privacy-preserving applications (for example, networking, Industrial IoT), avoiding the need for a Privacy Certificate Authority for attestation keys
- o This document applies primarily to "embedded" applications, where the device manufacturer ships the software image for the device.
- o The approach outlined in this document assumes a physical TPM.

##### [1.5.1.](#) Out of Scope



- o Run-Time Attestation: Run-time attestation of Linux or other multi-threaded operating system processes considerably expands the scope of the problem. Many researchers are working on that problem, but this document defers the run-time attestation problem.
- o Multi-Vendor Embedded Systems: Additional coordination would be needed for devices that themselves comprise hardware and software from multiple vendors, integrated by the end user.
- o Processor Sleep Modes: Embedded equipment typically does not "sleep", so sleep and hibernate modes are not considered.
- o Virtualization and Containerization: These technologies are increasingly used in embedded systems, but are not considered in this revision of the document.

#### 1.5.2. Why Remote Integrity Verification?

Remote Integrity Verification can go a long way to solving the "Lying Endpoint" problem, in which malicious software on an endpoint may both subvert the intended function, and also prevent the endpoint from reporting its compromised status. Man-in-the Middle attacks are also made more difficult through a strong focus on device identity

Attestation data can be used for asset management, vulnerability and compliance assessment, plus configuration management.

#### 1.5.3. Network Device Attestation Challenges

There have been demonstrations of attestation using TPMs for years, accompanied by compelling security reasons for adopting attestation. Despite this, the technology has not been widely adopted, in part, due to the difficulties in deploying TPM-based attestation. Some of those difficulties are:

- o Standardizing device identity. Creating and using unique device identifiers is difficult, especially in a privacy-sensitive environment. But attestation is of limited value if the operator is unable to determine which devices pass attestation validation tests, and which fail. This problem is substantially simplified for infrastructure devices like network equipment, where identity can be explicitly coded using IEEE 802.1AR, but doing so relies on adoption of 802.1AR [[IEEE-802-1AR](#)] by manufacturers and hardware system integrators.

- o Standardizing attestation representations and conveyance. Interoperable remote attestation has a fundamental dependence on vendors agreeing to a limited set of network protocols for communicating attestation data. Network device vendors will be slow to adopt the protocols necessary to implement remote attestation without a fully-realized plan for deployment.
- o Interoperability. Networking equipment operates in a fundamentally multi-vendor environment, putting additional emphasis on the need for standardized procedures and protocols.
- o Attestation evidence is complex. Operating systems used in larger embedded devices are often multi-threaded, so the order of completion for individual processes is non-deterministic. While the hash of a specific component is stable, once extended into a PCR, the resulting values are dependent on the (non-deterministic) ordering of events, so there will never be a single known-good value for some PCRs. Careful analysis of event logs can provide proof that the expected modules loaded, but it's much more complicated than simply comparing hashes.
- o Software configurations can have seemingly infinite variability. This problem is nearly intractable on PC and Server equipment, where end users have unending needs for customization and new applications. However, embedded systems, like networking equipment, are often simpler, in that there are fewer variations and releases, with vendors typically offering fewer options for mixing and matching.
- o Software updates can be complex. Even the most organized network operator may have many different releases in their network at any given time, with the result that there's never a single digest or fingerprint that indicates the software is "correct"; digests formed by hashing software modules on a device can only show the correct combination of versions for a specific device at a specific time.

None of these issues are insurmountable, but together, they've made deployment of attestation a major challenge. The intent of this document is to outline an attestation profile that's simple enough to deploy, while yielding enough security to be useful.

#### 1.5.4. Why is OS Attestation Different?

Even in embedded systems, adding Attestation at the OS level (e.g. Linux IMA, Integrity Measurement Architecture [[IMA](#)]) increases the

number of objects to be attested by one or two orders of magnitude,

involves software that's updated and changed frequently, and introduces processes that complete in unpredictable order.

TCG and others (including the Linux community) are working on methods and procedures for attesting the operating system and application software, but standardization is still in process.

## [2.](#) Solution Outline

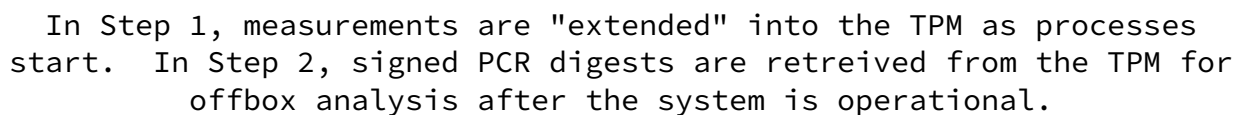
### [2.1.](#) 2.1 RIV Software Configuration Attestation using TPM

RIV Attestation is a process for determining the identity of software running on a specifically-identified device. Remote Attestation is broken into two phases, shown in Figure 1:

- o During system startup, measurements (i.e., hashes computed as fingerprints of files) are extended, or cryptographically folded, into the TPM. Entries are also added to an informational log. The measurement process generally follows the Chain of Trust model used in Measured Boot, where each stage of the system measures the next one before launching it.
- o Once the device is running and has operational network connectivity, a separate, trusted server (called a Verifier in this document) can interrogate the network device to retrieve the logs and a copy of the digests collected by hashing each software object, signed by an attestation private key known only to the TPM.

The result is that the Verifier can verify the device's identity by checking the certificate corresponding to the TPM's attestation private key, and can validate the software that was launched by comparing digests in the log with known-good values, and verifying their correctness by comparing with the signed digests from the TPM.

It should be noted that attestation and identity are inextricably linked; signed evidence that a particular version of software was loaded is of little value without cryptographic proof of the identity of the device producing the evidence.



## 2.2. RIV Keying

TPM 1.2 and TPM 2.0 have a variety of rules separating the functions of identity and attestation, allowing for use-cases where software

configuration must be attested, but privacy must be maintained.

To accommodate these rules in an environment where device privacy is not normally a requirement, the TCG Guidance for Securing Network Equipment [[NetEq](#)] suggests using separate keys for Identity (i.e., DevID) and Attestation (i.e., signing a quote of the contents of the PCRs).

In this case, the device manufacturer should provision an Initial Attestation Key (IAK) and x.509 certificate that parallels the IDevID, with the same device ID information as the IDevID certificate (i.e., the same Subject Name and Subject Alt Name, even though the key pairs are different). This allows a quote from the device, signed by the IAK, to be linked directly to the device that provided it, by examining the corresponding IAK certificate.

Inclusion of an IAK by a vendor does not preclude a mechanism whereby an Administrator can define Local Attestation Keys (LAKs) if desired.

### [2.3.](#) RIV Information Flow

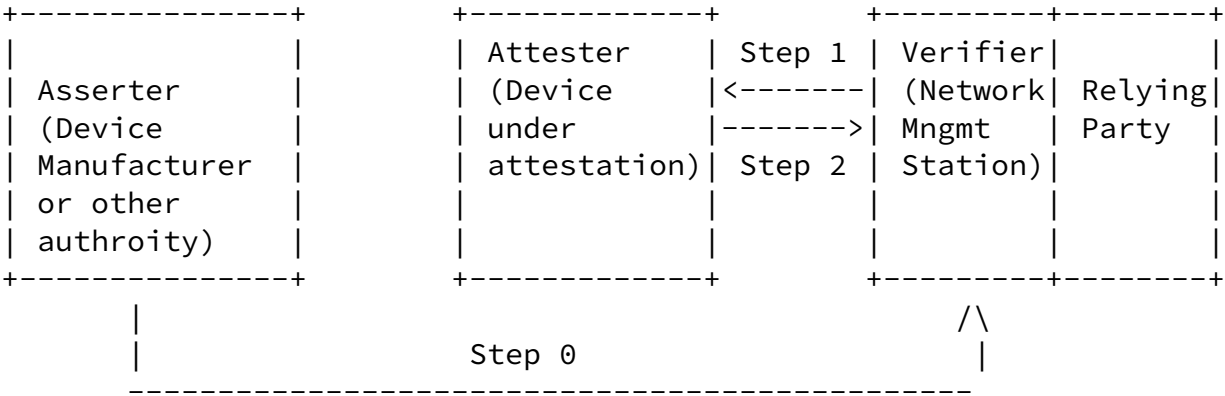
RIV workflow for networking equipment is organized around a simple use-case, where a network operator wishes to verify the integrity of software installed in specific, fielded devices. This use-case implies several components:

1. A Device (e.g. a router or other embedded device, also known as an Attester) somewhere and the network operator wants to examine its boot state.
2. A Verifier (which might be a network management station) somewhere separate from the Device that will retrieve the information and analyze it to pass judgement on the security posture of the device.
3. A Relying Party, which has access to the Verifier to request attestation and to act on results.
4. This document assumes that signed Reference Integrity Measurements (RIMs) (aka "golden measurements") can either be created by the device manufacturer and shipped along with the

device as part of its software image, or alternatively, could be obtained a number of other ways (direct to the verifier from the manufacturer, from a third party, from the owner's observation of what's thought to be a "known good system", etc.). Retrieving RIMs from the device itself allows attestation to be done in systems which may not have access to the public internet, or by other devices that are not management stations per-se (e.g., a peer device). If reference measurements are obtained from multiple sources, the Verifier may need to evaluate the relative level of trust to be placed in each source in case of a discrepancy.

These components are illustrated in Figure 2.

A more-detailed taxonomy of terms is given in [I-D.birkholz-rats-architecture]



In Step 0, The Asserter (the device manufacturer) provides a Software Image accompanied by Reference Integrity Measurements (RIMs) to the Attester (the device under attestation) signed by the asserter. In Step 1, the Verifier (Network Management Station), on behalf of a Relying Party, requests Identity, Measurement Values (and possibly

RIMs) from the Attester. In Step 2, the Attester responds to the request by providing a DevID, Quotes (measured values), and optionally RIMs, signed by the Attester.

Figure 2: RIV Reference Configuration for Network Equipment

See [Section 3.1.1](#) for more narrowly defined terms related to Attestation

## [2.4.](#) RIV Simplifying Assumptions

This document makes the following simplifying assumptions to reduce complexity:

- o The product to be attested is shipped with an IEEE 802.1AR DevID and an Initial Attestation Key (IAK) with certificate. The IAK cert contains the same identity information as the DevID (specifically, the same Subject Name and Subject Alt Name, signed by the manufacturer), but it's a type of key that can be used to sign a TPM Quote. This convention is described in TCG Guidance for Securing Network Equipment [[NetEq](#)]. For network equipment, which is generally non-privacy-sensitive, shipping a device with both an IDevID and an IAK already provisioned substantially simplifies initial startup. Privacy-sensitive applications may use the TCG Platform Certificate and additional procedures to install identity credentials on the platform after manufacture. (See [Section 2.3.1](#) below for the Platform Certificate alternative)
- o The product is equipped with a Root of Trust for Measurement, Root of Trust for Storage and Root of Trust for Reporting that is

capable of conforming to the TCG Trusted Attestation Protocol (TAP) Information Model [[TAP](#)].

- o The vendor will ship Reference Integrity Measurements (i.e., known-good measurements) in the form of signed CoSWID tags [[I-D.ietf-sacm-coswid](#)], [[SWID](#)], as described in TCG Reference Integrity Measurement Manifest [[RIM](#)].

### [2.4.1.](#) DevID Alternatives

Some situations may have privacy-sensitive requirements that preclude

shipping every device with an Initial Device ID installed. In these cases, the IDevID can be installed remotely using the TCG Platform Certificate [[Platform-Certificates](#)].

Some security-sensitive administrators may want to install their own identity credentials to certify platform identity and attestation results. IEEE 802.1AR [[IEEE-802-1AR](#)] allows for both Initial Device Identity credentials, installed by the manufacturer, or Local Device Identity credentials installed by the administrator of the platform. TCG TPM 2.0 Keys documents [[Platform-DevID-TPM-2.0](#)] and [[PC-Client-BIOS-TPM-2.0](#)] specifies analogous Initial and Local Attestation Keys (IAK and LAK), and contains figures showing the relationship between IDevID, LDevID, IAK and LAK keys.

Platform administrators are free to use any number of criteria to judge authenticity of a platform before installing local identity keys, as part of an on-boarding process. The TCG TPM 2.0 Keys document [[Platform-DevID-TPM-2.0](#)] also outlines procedures for creating Local Attestation Keys and Local Device IDs (LDevIDs) rooted in the manufacturer's IDevID as a check to reduce the chances that counterfeit devices are installed in the network.

Note that many networking devices are expected to self-configure (aka Zero Touch Provisioning). Current standardized zero-touch mechanisms such as [[RFC8572](#)] assume that identity keys are already in place before network on-boarding can start.

#### [2.4.2.](#) Additional Attestation of Platform Characteristics

The Platform Attribute Credential [[Platform-Certificates](#)] can also be used to convey additional information about a platform from the manufacturer or other entities in the supply chain. While outside the scope of RIV, the Platform Attribute Credential can deliver information such as lists of serial numbers for components embedded in a device or security assertions related to the platform, signed by the manufacturer, system integrator or value-added-reseller.

#### [2.4.3.](#) Root of Trust for Measurement

The measurements needed for attestation require that the device being attested is equipped with a Root of Trust for Measurement, i.e., some



trustworthy mechanism that can take the first measurement in the chain of trust required to attest that each stage of system startup is verified, and a Root of Trust for Reporting to report the results [[Roots-of-Trust](#)].

While there are many complex aspects of a Root of Trust, two aspects that are important in the case of attestation are:

- o The first measurement taken by the Root of Trust for Measurement, and stored in the Root of Trust for Storage, is presumed to be correct.
- o There must not be a way to reset the RTM without re-entering the RTS code.

The first measurement can't be checked by a code that's been previously checked by something further back up the chain (it's the first, after all); if that first measurement can be subverted, none of the remaining measurements can be trusted. (See [[NIST-SP-800-155](#)])

#### [2.4.4.](#) Reference Integrity Measurements (RIMs)

Much of attestation focuses on collecting and transmitting 'evidence' in the form of PCR measurements and attestation logs. But the critical part of the process is enabling the verifier to decide whether the measured hashes are "the right ones" or not.

While it must be up to network administrators to decide what they want on their networks, the software supplier should supply the Reference Integrity Measurements, (aka Golden Measurements or "known good" hash digests) that may be used by a verifier to determine if evidence shows known good, known bad or unknown software configurations.

In general, there are two kinds of reference measurements:

1. Measurements of early system startup (e.g., BIOS, boot loader, OS kernel) are essentially single threaded, and executed exactly once, in a known sequence, before any results could be reported. In this case, while the method for computing the hash and extending relevant PCRs may be complicated, the net result is that the software (more likely, firmware) vendor will have one known good PCR value that "should" be present in the PCR after

the box has booted. In this case, the signed reference measurement simply lists the expected hash for the given version.

2. Measurements taken later in operation of the system, once an OS has started (for example, Linux IMA[16.]), may be more complex, with unpredictable "final" PCR values. In this case, the Verifier must have enough information to reconstruct the expected PCR values from logs and signed reference measurements from a trusted authority.

In both cases, the expected values can be expressed as signed CoSWID tags, but the SWID structure in the second case is somewhat more complex. An example of how CoSWIDs could be incorporated into a reference manifest can be found in the IETF Internet-Draft "A SUIT Manifest Extension for Concise Software Identifiers" [[I-D.birkholz-suit-coswid-manifest](#)].

The TCG has done exploratory work in defining formats for reference integrity manifests under the working title TCG Reference Integrity Manifest [[RIM](#)].

#### [2.4.5.](#) Attestation Logs

Quotes from a TPM can provide evidence of the state of a device at the time the quote was requested, but to make sense of the quote in most cases an event log of what software modules contributed which values to the quote during startup must also be provided. The log needs not be secured, but it is essential that the logs contain enough information to exactly reconstruct the state of whatever went into the quote (e.g., PCR values).

TCG has defined several event log formats:

- o Legacy BIOS event log (TCG PC Client Specific Implementation Specification for Conventional BIOS, [Section 11.3](#)[PC-Client-BIOS-TPM-1.2])
- o UEFI BIOS event log (TCG EFI Platform Specification for TPM Family 1.1 or 1.2, [Section 7](#) [[EFI](#)])
- o Canonical Event Log [[Canonical-Event-Log](#)]

It should be noted that a given device might use more than one event log format (e.g., a UEFI log during initial boot, switching to Canonical Log when the host OS launches).

The TCG SNMP Attestation MIB [[SNMP-Attestation-MIB](#)] will support any

record-oriented log format, including the three TCG-defined formats,

but it currently leaves figuring out which log(s) are in what format up to the Verifier.

### [3.](#) Standards Components

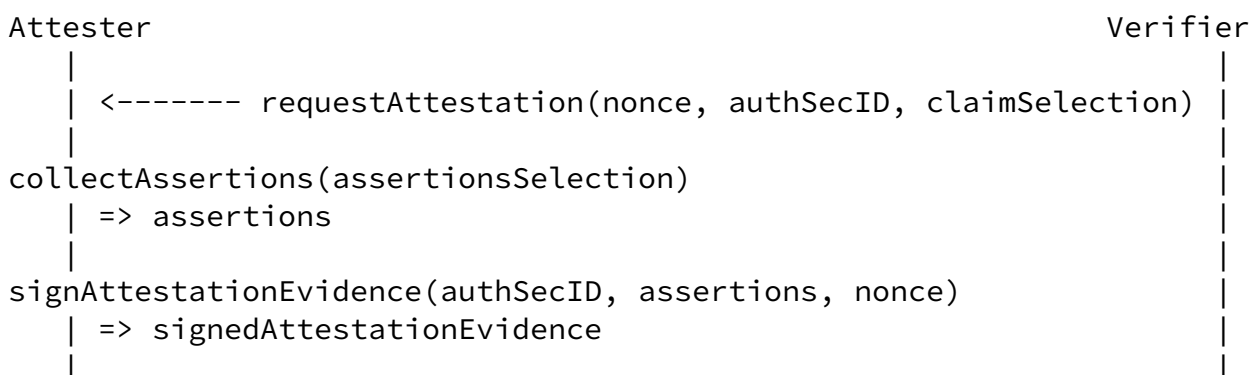
#### [3.1.](#) Reference Models

##### [3.1.1.](#) IETF Reference Model for Challenge-Response Remote Attestation

Initial work at IETF defines remote attestation as follows:

- o The Reference Interaction Model for Challenge-Response-based Remote Attestation is based on the standard roles defined in [\[I-D.birkholz-rats-architecture\]](#):
- o
  - \* **Attester:** The role that designates the subject of the remote attestation. A system entity that is the provider of evidence takes on the role of an Attester.
  - \* **Verifier:** The role that designates the system entity and that is the appraiser of evidence provided by the Attester. A system entity that is the consumer of evidence takes on the role of a Verifier.

The following diagram illustrates a common information flow between a Verifier and an Attester, specified in [\[I-D.birkholz-rats-reference-interaction-model\]](#):



```

| signedAttestationEvidence -----> |
| verifyAttestationEvidence(signedAttestationEvidence, refassertions) |
|                                     attestationResult <= |
|

```

Figure 3: IETF Attestation Information Flow

The RIV approach outlined in this document aligns with the RATS reference model.

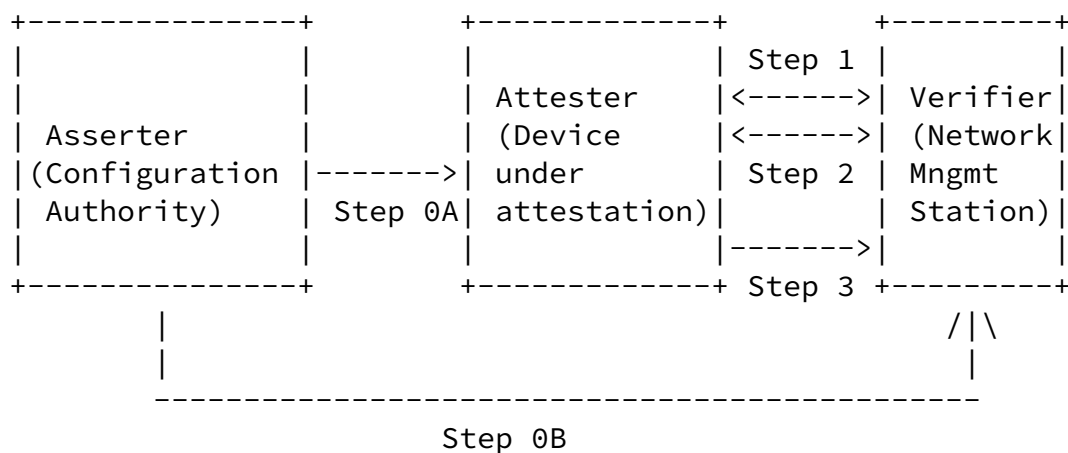
### [3.2.](#) RIV Workflow

The overall flow for an attestation session is shown in Figure 4. In this diagram:

- o Step 0, positioning of the signed reference measurements, may happen in two ways:
  - o Step 0A below shows a verifier obtaining reference measurements directly from a software configuration authority, whether it's the vendor or another authority chosen by the system administrator. The reference measurements are signed by the Asserter (i.e., the software configuration authority).
  - o - Or - Step 0B, the reference measurements, signed by the Asserter, may be distributed as part of software installation, long before the attestation session begins. Software installation is usually vendor-dependent, so there are no standards involved in this step. However, the verifier can use the same protocol to obtain the reference measurements from the device as it would have used with an external reference authority
- o In Step 1, the Verifier initiates an attestation session by opening a TLS session, validated using the DevID to prove that the connection is attesting the right box.
- o In Step 2, measured values are retrieved from the Attester's TPM using a YANG or SNMP interface that implements the TCG TAP model (e.g. YANG Module for Basic Challenge-Response-based Remote Attestation Procedures

[[I-D.birkholz-yang-basic-remote-attestation](#)]).

- o In Step 3, the Attester also delivers a copy of the signed reference measurements, using Software Inventory YANG module based on Software Identifiers [[I-D.birkholz-yang-swid](#)].



Either CoSWID-encoded reference measurements are signed by a trusted authority and retrieved directly prior to attestation (as shown in Step 0A), or CoSWID-encoded reference measurements are signed by the device manufacturer, installed on the device by a proprietary installer, and delivered during attestation (as shown in Step 0B). In Step 1, the Verifier initiates a connection for attestation. The Attester's identity is validated using DevID with TLS. In Step 2, a nonce, quotes (measured values) and measurement log are conveyed via TAP with a protocol-specific binding (e.g. SNMP). Logs are sent in the Canonical Log Format In Step 3, CoSWID-encoded reference measurements are retrieved from the Attester using the YANG ([[I-D.birkholz-yang-swid](#)]). .

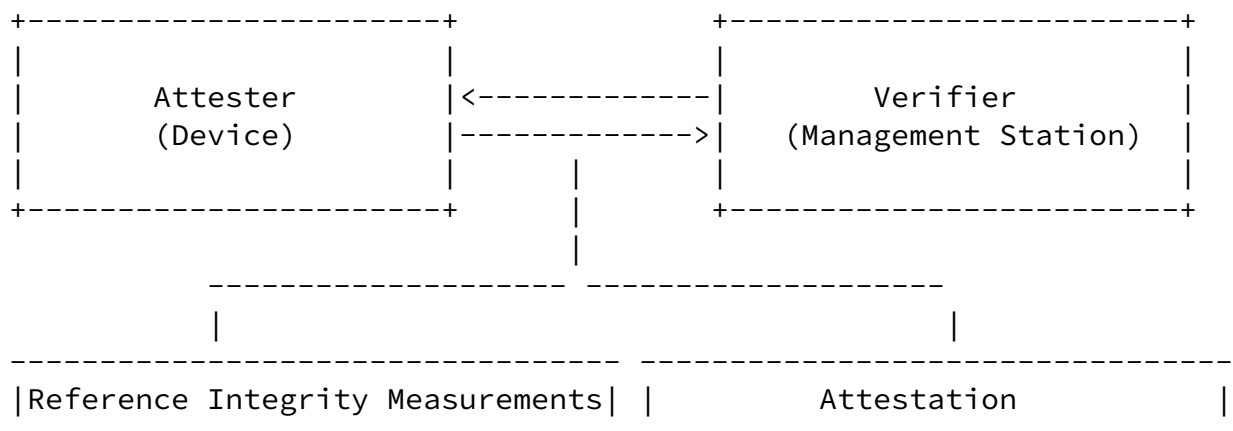
Figure 4: RIV Protocol and Encoding Summary

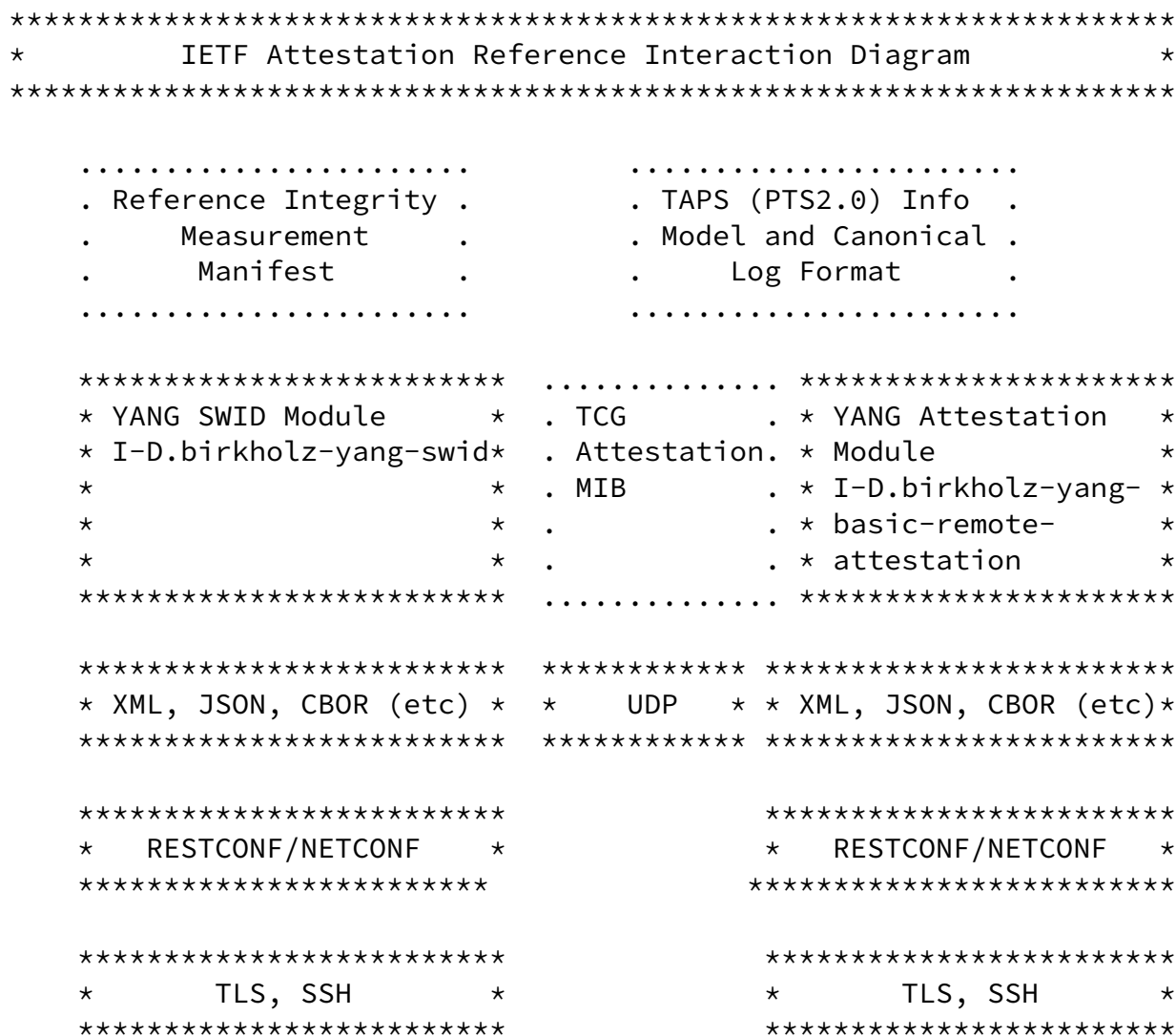
The following components are used:

1. TPM Keys are configured according to [[Platform-DevID-TPM-2.0](#)], [[PC-Client-BIOS-TPM-1.2](#)], or [[Platform-ID-TPM-1.2](#)]
2. Measurements of bootable modules are taken according to TCG PC Client [[PC-Client-BIOS-TPM-2.0](#)] and Linux IMA [[IMA](#)]
3. Device Identity is managed by IEEE 802.1AR certificates [[IEEE-802-1AR](#)], with keys protected by TPMs.
4. Quotes are retrieved according to TCG TAP Information Model [[TAP](#)]
5. Reference Integrity Measurements are encoded as CoSWID tags, as defined in the TCG RIM document [[RIM](#)], compatible with NIST IR 8060 [[NIST-IR-8060](#)] and the IETF CoSWID draft [[I-D.ietf-sacm-coswid](#)]. Reference measurements are signed by the device manufacturer.

### [3.3](#). Layering Model for Network Equipment Attester and Verifier

Retrieval of identity and attestation state uses one protocol stack, while retrieval of Reference Measurements uses a different set of protocols. Figure 5 shows the components involved.





IETF documents are captured in boxes surrounded by asterisks. TCG documents are shown in boxes surrounded by dots. The IETF Attestation Reference Interaction Diagram, Reference Integrity Measurement Manifest, TAPS Information Model and Canonical Log Format, and both YANG modules are works in progress. Information Model layers describe abstract data objects that can be requested, and the corresponding response SNMP is still widely used, but the industry is transitioning to YANG, so in some cases, both will be required. TLS Authentication with TPM has been shown to work; SSH authentication using TPM-protected keys is not as easily done [as of 2019]

Figure 5: RIV Protocol Stacks

## [4.](#) Conclusion

TCG technologies can play an important part in the implementation of Remote Integrity Verification. Standards for many of the components needed for implementation of RIV already exist:

- o Platform identity can be based on IEEE 802.1AR Device identity, coupled with careful supply-chain management by the manufacturer.
- o Complex supply chains can be certified using TCG Platform Certificates [[Platform-Certificates](#)]
- o The TCG TAP mechanism can be used to retrieve attestation evidence. Work is needed on a YANG model for this protocol.
- o Reference Measurements must be conveyed from the software authority (e.g., the manufacturer) to the system in which verification will take place. IETF CoSWID work forms the basis for this, but new work is needed to create an information model and YANG implementation.

Gaps still exist for implementation in Network Equipment (as of May 2019):

- o Coordination of YANG model development with the IETF is still needed
- o Specifications for management of signed Reference Integrity Measurements must still be completed

## [5.](#) Appendix

### [5.1.](#) Implementation Notes

Table 1 summarizes many of the actions needed to complete an Attestation system, with links to relevant documents. While documents are controlled by a number of standards organizations, the



implied actions required for implementation are all the responsibility of the manufacturer of the device, unless otherwise noted.

Component	Controlling Specification
<ul style="list-style-type: none"> <li>Make a Secure execution environment <ul style="list-style-type: none"> <li>o Attestation depends entirely on a secure root of trust for measurement.</li> <li>o Refer to TCG Root of Trust for Measurement.</li> <li>o NIST SP 800-193 also provides guidelines on Roots of Trust</li> </ul> </li> </ul>	TCG RoT UEFI.org
<ul style="list-style-type: none"> <li>Get a TPM properly provisioned as described in TCG documents.</li> </ul>	TCG TPM DevID TCG Platform Certificate
<ul style="list-style-type: none"> <li>Put a DevID or Platform Cert in the TPM <ul style="list-style-type: none"> <li>o Install an Initial Attestation Key at the same time so that Attestation can work out of the box</li> <li>o Equipment suppliers and owners may want to implement Local Device ID as well as Initial Device ID</li> </ul> </li> </ul>	TCG TPM DevID TCG Platform Certificate ----- IEEE 802.1AR
<ul style="list-style-type: none"> <li>Connect the TPM to the TLS stack <ul style="list-style-type: none"> <li>o Use the DevID in the TPM to authenticate TAP connections, identifying the device</li> </ul> </li> </ul>	Vendor TLS stack (This action is simply configuring TLS to use the DevID as its trust anchor.)
<ul style="list-style-type: none"> <li>Make CoSWID tags for BIOS/Loader/LKernel objects <ul style="list-style-type: none"> <li>o Add reference measurements into SWID tags</li> <li>o Manufacturer should sign the SWID tags</li> <li>o This should be covered in a new TCG Reference Integrity Manifest document <ul style="list-style-type: none"> <li>- IWG should define the literal SWID format</li> </ul> </li> </ul> </li> </ul>	IETF CoSWID ISO/IEC 19770-2 NIST IR 8060 TagVault SWID Tag Signing Guidance -----

<ul style="list-style-type: none"> <li>- IWG should evaluate whether IETF SUIT is a suitable manifest when multiple SWID tags are involved</li> <li>- There could be a proof-of-concept project to actually make sample SWID tags (a gap might appear in the process)</li> </ul>	TCG RIM
<p>Package the SWID tags with a vendor software release</p> <ul style="list-style-type: none"> <li>o A tag-generator plugin could help (i.e., a plugin for common development environments. NIST has something that plugs into Maven Build Environment)</li> </ul>	<p>There is no need to specify where the tags are stored in a vendor OS, as long as there is a standards-based mechanism to retrieve them.</p> <p>TCG RIM</p>
<p>BIOS SWIDs might be hard to manage on an OS disk-- maybe keep them in the BIOS flash?</p> <ul style="list-style-type: none"> <li>o Maybe a UEFI Var? Would its name have to be specified by UEFI.org?</li> <li>o How big is a BIOS SWID tag? Do we need to use a tag ID instead of an actual tag?</li> <li>o Note that the presence of Option ROMs turns the BIOS reference measurements into a multi-vendor interoperability problem</li> </ul>	TCG RIM
<p>Use PC Client measurement definitions as a starting point to define the use of PCRs (although Windows OS is rare on Networking Equipment)</p>	<p>TCG PC Client BIOS</p> <p>There have been proposals for non-PC-Client allocation of PCRs, although no specific document exists yet.</p>
<p>Use TAP to retrieve measurements</p> <ul style="list-style-type: none"> <li>o Map TAP to SNMP</li> <li>o Map to YANG</li> <li>o Complete Canonical Log Format</li> </ul>	<p>TCG SNMP MIB</p> <p>YANG Module for Basic Attestation</p>

Internet-Draft

Network Device Attestation Workflow

June 2019

	TCG Canonical Log Format
Posture Collection Server (as described in IETF SACMs ECP) would have to request the attestation and analyze the result The Management application might be broken down to several more components:	
o A Posture Manager Server which collects reports and stores them in a database	
o One or more Analyzers that can look at the results and figure out what it means.	

Figure 6: Component Status

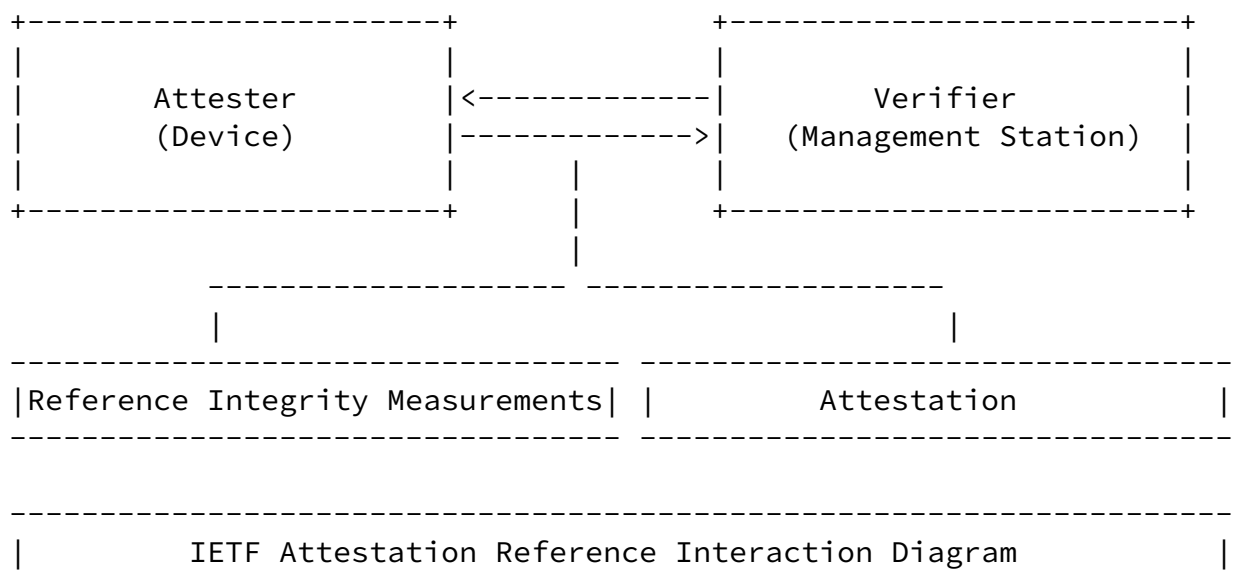
## 5.2. Comparison with TCG PTS / IETF NEA

Some components of an Attestation system have been implemented for end-user machines such as PCs and laptops. Figure 7 shows the corresponding protocol stacks.

Internet-Draft

Network Device Attestation Workflow

June 2019



.....	
. No Existing .	TAPS (PTS2.0) Info Model and
. Reference Integrity .	Canonical Log Format
. Measurement Manifest.	
. Protocols Exist .	-----
. .	
. .	
. .	YANG Attestation     IETF NEA
. .	Module     Msg and
. .	I-D.birkholz-yang-     Attrib.
. .	basic-remote-     for PA-
. .	attestation     TNC
. .	-----
. .	XML, JSON, CBOR     PT-TLS
. .	-----   (for
. .	-----   endpoint
. .	NETCONF, RESTCONF,     mcachines

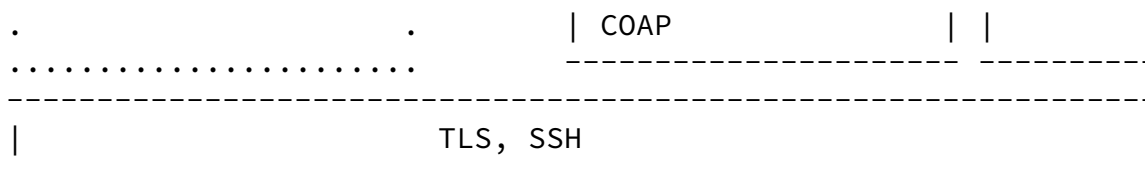


Figure 7: Attestation for End User Computers

## 6. IANA Considerations

This memo includes no request to IANA.

## 7. Security Considerations

TBD

## 8. Informative References

### [Canonical-Event-Log]

Trusted Computing Group, "DRAFT Canonical Event Log Format Version: 1.0, Revision: .12", October 2018.

### [EFI]

Trusted Computing Group, "TCG EFI Platform Specification for TPM Family 1.1 or 1.2, Specification Version 1.22, Revision 15", January 2014, <<https://trustedcomputinggroup.org/wp-content/uploads/EFI-Protocol-Specification-rev13-160330final.pdf>>.

### [Firmware-Profile]

Trusted Computing Group, "Trusted Computing Group, PC Client Specific Platform Firmware Profile Specification Family "2.0", Level 00 Revision 1.03 Version 51", June 2019, <<https://trustedcomputinggroup.org/pc-client-specific-platform-firmware-profile-specification/>>.

### [I-D.birkholz-rats-architecture]

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

[I-D.birkholz-rats-reference-interaction-model]

Birkholz, H. and M. Eckel, "Reference Interaction Model for Challenge-Response-based Remote Attestation", [draft-birkholz-rats-reference-interaction-model-00](#) (work in progress), March 2019.

[I-D.birkholz-suit-coswid-manifest]

Birkholz, H., "A SUIT Manifest Extension for Concise Software Identifiers", [draft-birkholz-suit-coswid-manifest-00](#) (work in progress), July 2018.

[I-D.birkholz-yang-basic-remote-attestation]

Birkholz, H., Eckel, M., Bhandari, S., Sulzen, B., Voit, E., and G. Fedorkow, "YANG Module for Basic Challenge-Response-based Remote Attestation Procedures", [draft-birkholz-yang-basic-remote-attestation-01](#) (work in progress), October 2018.

[I-D.birkholz-yang-swid]

Birkholz, H., "Software Inventory YANG module based on Software Identifiers", [draft-birkholz-yang-swid-02](#) (work in progress), October 2018.

[I-D.ietf-sacm-coswid]

Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", [draft-ietf-sacm-coswid-11](#) (work in progress), June 2019.

[IEEE-802-1AR]

Seaman, M., "802.1AR-2018 - IEEE Standard for Local and Metropolitan Area Networks - Secure Device Identity, IEEE Computer Society", August 2018.

[IMA]

and , "Integrity Measurement Architecture", June 2019, [<https://sourceforge.net/p/linux-ima/wiki/Home/>](https://sourceforge.net/p/linux-ima/wiki/Home/).

[NetEq]

Trusted Computing Group, "TCG Guidance for Securing Network Equipment", January 2018, [<https://trustedcomputinggroup.org/wp-content/uploads/](https://trustedcomputinggroup.org/wp-content/uploads/)

[TCG Guidance for Securing NetEq 1.0r29.pdf](#)>.

[NIST-IR-8060]

National Institute for Standards and Technology, "Guidelines for the Creation of Interoperable Software Identification (SWID) Tags", April 2016, <<https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8060.pdf>>.

[NIST-SP-800-155]

National Institute for Standards and Technology, "BIOS Integrity Measurement Guidelines (Draft)", December 2011, <[https://csrc.nist.gov/csrc/media/publications/sp/800-155/draft/documents/draft-sp800-155\\_dec2011.pdf](https://csrc.nist.gov/csrc/media/publications/sp/800-155/draft/documents/draft-sp800-155_dec2011.pdf)>.

[PC-Client-BIOS-TPM-1.2]

Trusted Computing Group, "TCG PC Client Specific Implementation Specification for Conventional BIOS, Specification Version 1.21 Errata, Revision 1.00", February 2012, <[https://www.trustedcomputinggroup.org/wp-content/uploads/TCG\\_PCClientImplementation\\_1-21\\_1\\_00.pdf](https://www.trustedcomputinggroup.org/wp-content/uploads/TCG_PCClientImplementation_1-21_1_00.pdf)>.

[PC-Client-BIOS-TPM-2.0]

Trusted Computing Group, "PC Client Specific Platform Firmware Profile Specification Family "2.0", Level 00 Revision 1.04", June 2019, <<https://trustedcomputinggroup.org/pc-client-specific-platform-firmware-profile-specification>>.

[Platform-Certificates]

Trusted Computing Group, "DRAFT: TCG Platform Attribute Credential Profile, Specification Version 1.0, Revision 15, 07 December 2017", December 2017.

[Platform-DevID-TPM-2.0]

Trusted Computing Group, "DRAFT: TPM Keys for Platform

[Platform-ID-TPM-1.2]

Trusted Computing Group, "TPM Keys for Platform Identity for TPM 1.2, Specification Version 1.0, Revision 3", August 2015, <[https://trustedcomputinggroup.org/wp-content/uploads/TPM\\_Keys\\_for\\_Platform\\_Identity\\_v1\\_0\\_r3\\_Final.pdf](https://trustedcomputinggroup.org/wp-content/uploads/TPM_Keys_for_Platform_Identity_v1_0_r3_Final.pdf)>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

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

[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>>.

[RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", [RFC 8348](#), DOI 10.17487/RFC8348, March 2018, <<https://www.rfc-editor.org/info/rfc8348>>.

[RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", [RFC 8572](#), DOI 10.17487/RFC8572, April 2019, <<https://www.rfc-editor.org/info/rfc8572>>.

[RIM] Trusted Computing Group, "DRAFT: TCG Reference Integrity Manifest", June 2019.

[Roots-of-Trust]

Trusted Computing Group, "TCG Roots of Trust Specification", October 2018, <<https://trustedcomputinggroup.org/wp-content/uploads/>



[SNMP-Attestation-MIB]

Trusted Computing Group, "DRAFT: SNMP MIB for TPM-Based Attestation, Specification Version 0.8, Revision 0.02", May 2018.

[SWID]

The International Organization for Standardization/ International Electrotechnical Commission, "Information Technology Software Asset Management Part 2: Software Identification Tag, ISO/IEC 19770-2", October 2015, <<https://www.iso.org/standard/65666.html>>.

[TAP]

Trusted Computing Group, "DRAFT: TCG Trusted Attestation Protocol (TAP) Information Model for TPM Families 1.2 and 2.0 and DICE Family 1.0, Version 1.0, Revision 0.29", October 2018.

Authors' Addresses

Guy Fedorkow (editor)  
Juniper Networks, Inc.  
US

Email: [gfedorkow@juniper.net](mailto:gfedorkow@juniper.net)

Jessica Fitzgerald-McKay  
National Security Agency  
US

Email: [jmfitz2@nsa.gov](mailto:jmfitz2@nsa.gov)