REMOTE SOFTWARE INTEGRITY VERIFICATION USING TRUSTED COMPUTING GROUP TPM

Guy Fedorkow, Juniper Networks

Jessica Fitzgerald-McKay, USG

Nov 2019

V1b

Introduction

Remote software integrity verification is a mechanism that can be used to determine the authenticity of software installed on a fielded device such as a router or firewall.

This ppt outlines work submitted as:

- draft-fedorkow-rats-network-device-attestation-01

The work is based on Trusted Computing Group document:

- TCG Remote Integrity Verification: Network Equipment Remote Attestation System
- https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline_v1r9b_pubrev.pdf

Problem Statement

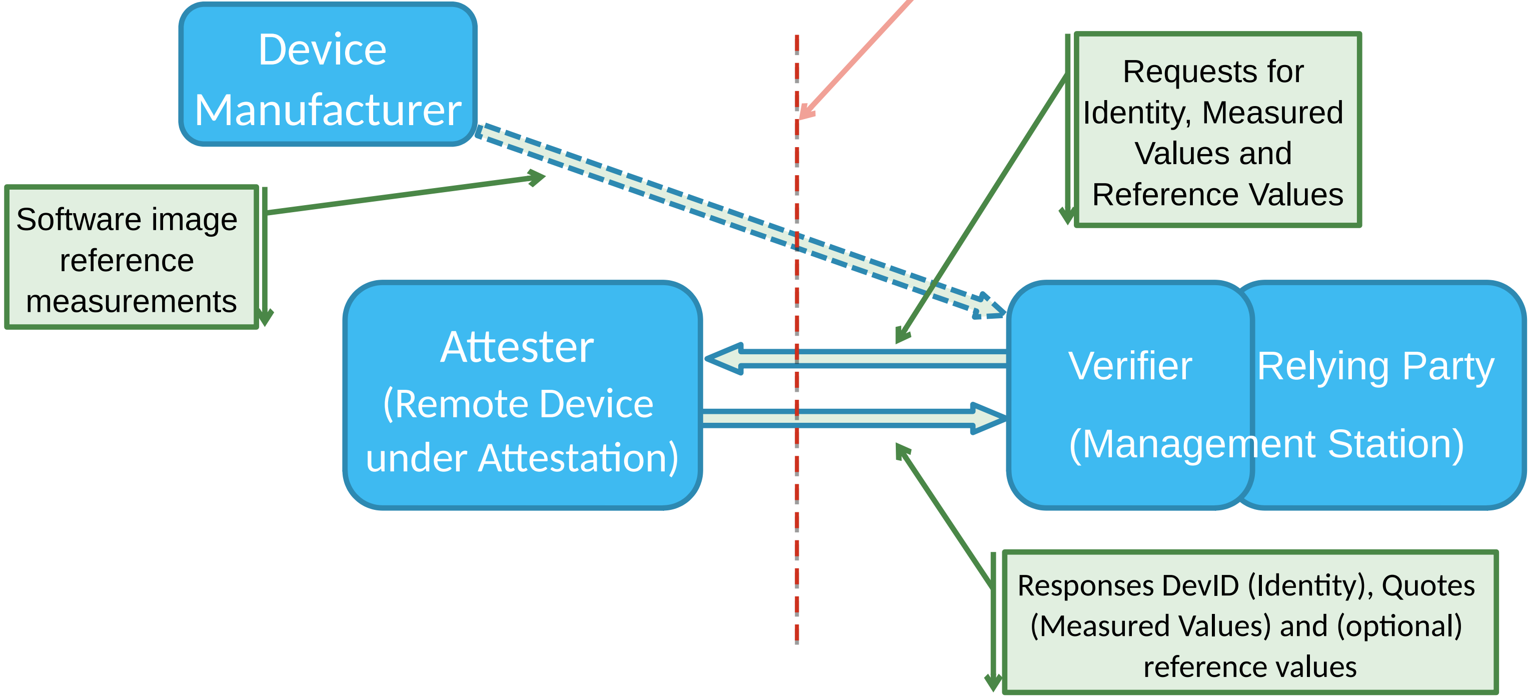
- How do you know what software is actually running on a device?
  - You could ask it, but it might not tell the truth
  - Attestation (‘measured boot’) establishes a chain of trust where each link measures the next before it starts
  - The TPM reports the results, signed by a key known only by the TPM
- A workflow must be established where the entity that wants the validation may query the device in question via standard protocols.
- Current RIV Scope:

  “*Things* that have a TPM, use YANG and don’t Sleep”

  - Compatibility with existing TPM practice is critical
  - Addition of protocol suites other than YANG (e.g. IIoT) would extend the scope

RIV Information Fl

RIV addresses traffic crossing the boundary to the remote device


Device Manufacturer
Software image reference measurements
Requests for Identity, Measured Values and Reference Values
Attester (Remote Device under Attestation)
Verifier
Relying Party
(Management Station)
Responses DevID (Identity), Quotes (Measured Values) and (optional) reference values

Security Considerations: Attacks Against RIV

Bad Stuff Can Happen:

- Keys may be compromised
- A counterfeit device may attempt to impersonate (spoof) a known authentic device
- Man-in-the-middle attacks may be used by a counterfeit device to attempt to deliver responses that originate in an actual authentic device
- Replay attacks may be attempted by a compromised device.

Defense Against Key Compromise

- RIV depends on keys generated inside the TPM for both Identity and Attestation
  - Secret keys cannot be extracted from the TPM*

- Certificate provenance must be managed carefully by device manufacturer
  - i.e., be careful of what you sign.

* Unless you find a bug or mount a very challenging physical attack

Defense Against Counterfeit Devices

- RIV depends on IEEE 802.1AR Device ID credentials protected by the TPM
  - Manufacturers should provision Initial Device Identity (IDevID)
  - Device Owners can (optionally) provision Local Device ID (LDevID)

- Pre-provisioned IDevID allows for Secure Zero Touch Provisioning (e.g. RFC 8572)

Defense Against Man-in-the-Middle Attacks

- Identity depends on difficult-to-forge DevID
- Software Attestation evidence is collected *and signed* inside the TPM

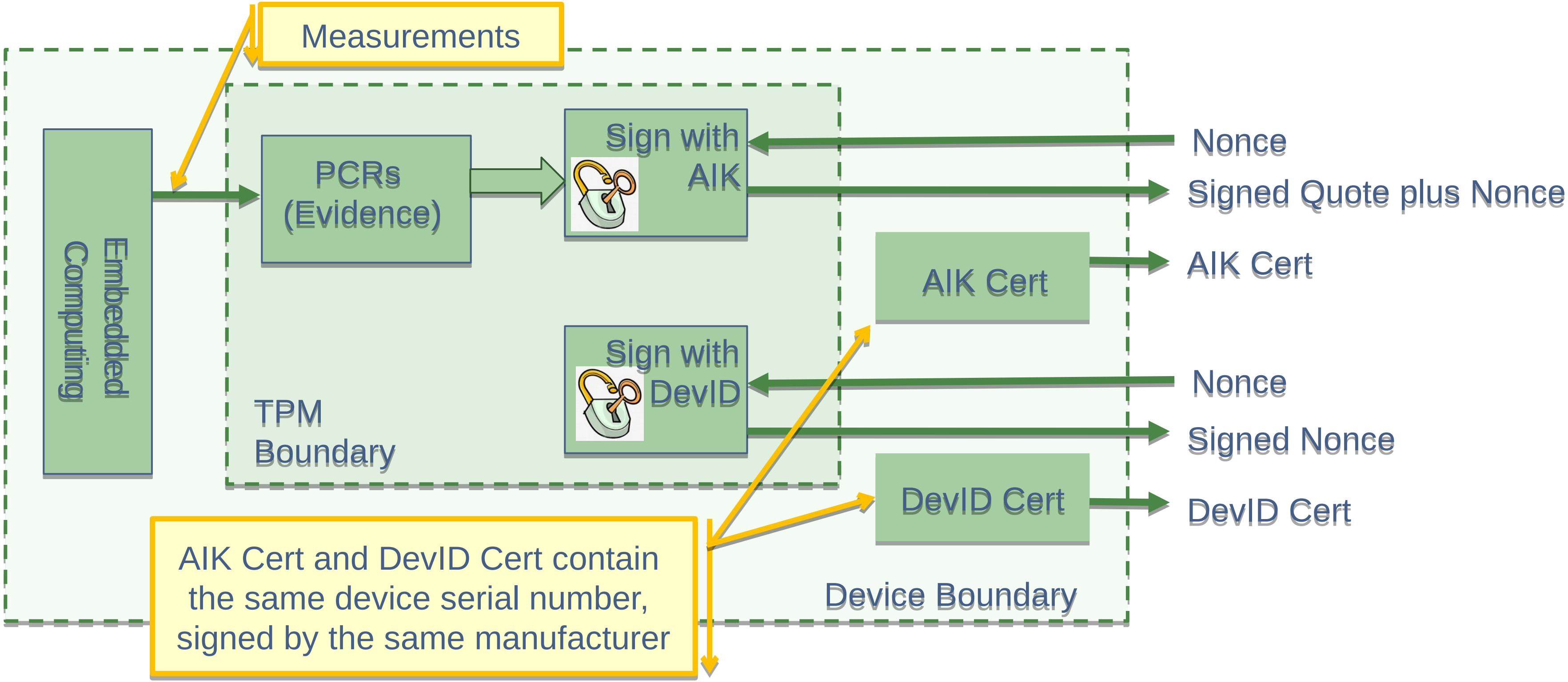
  …using an Attestation key that can't be used to sign anything else

Implications:

- RIV calls for Separate DevID and Attestation keys
- The Attestation Certificate must include the same Identity information as the IDevID*
- Signed TPM data structures must be transmitted without modification

* See Asokan Man in the Middle Attack, RFC 6813

RIV Man-in-the-Middle Defense



AIK Cert and DevID Cert contain the same device serial number, signed by the same manufacturer

Defense Against Replay Attacks

- All exchanges to prove Identity or provide software integrity evidence ("TPM quotes") contain Nonces that prove freshness
  - Verifier or Relying Party makes up the Nonce
  - Signed TPM data structures contain the Nonce, returning proof to the Verifier or Relying Party

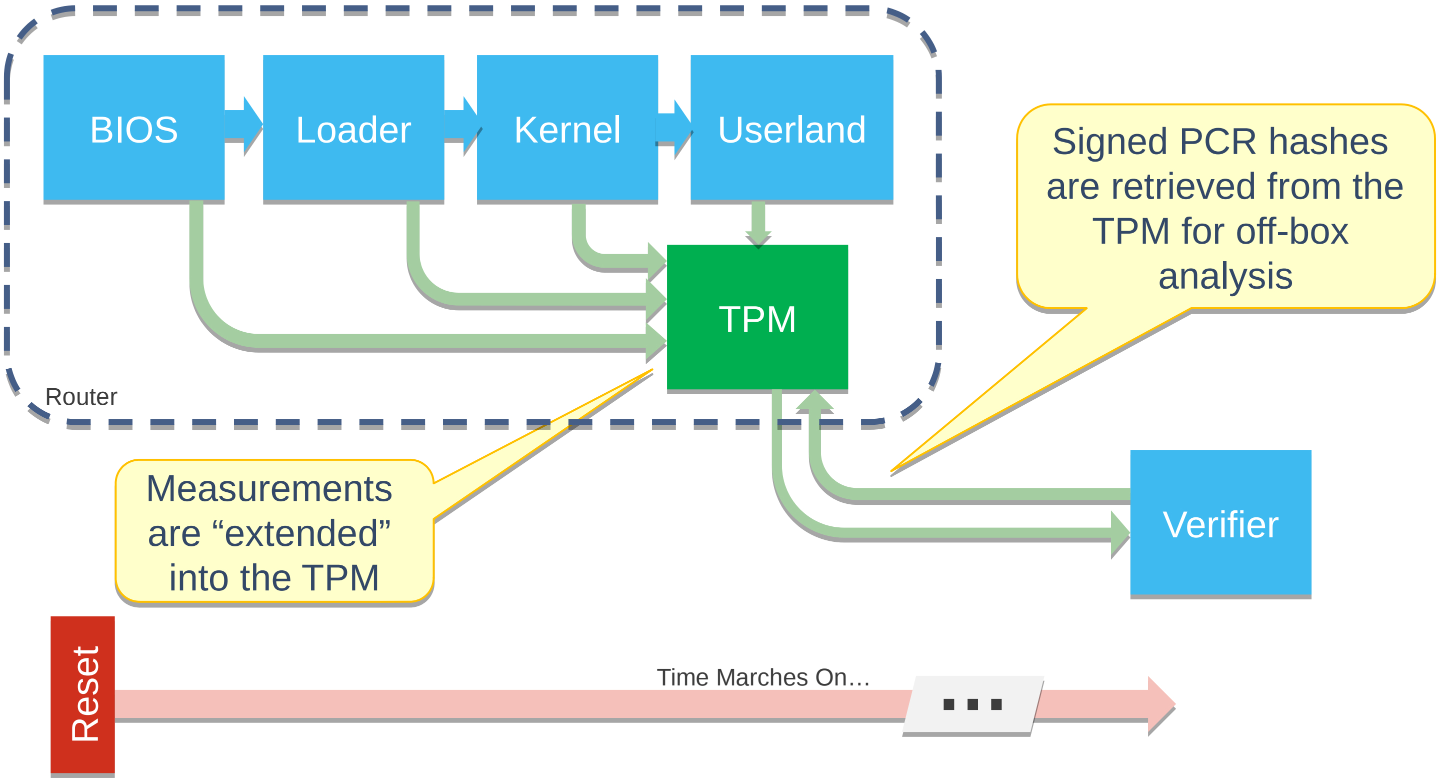
  - TUDA (draft-birkholz-rats-tuda-01) provides an alternate way to prevent replay of quotes
    - not currently in the YANG model (draft-birkholz-rats-basic-yang-module-01)

Conclusion

- TPM rules provide difficult-to-forge evidence of identity and software integrity
- But that means the TPM data structures must be transmitted unmodified.

BACKUP

TCG Attestation Information Flow



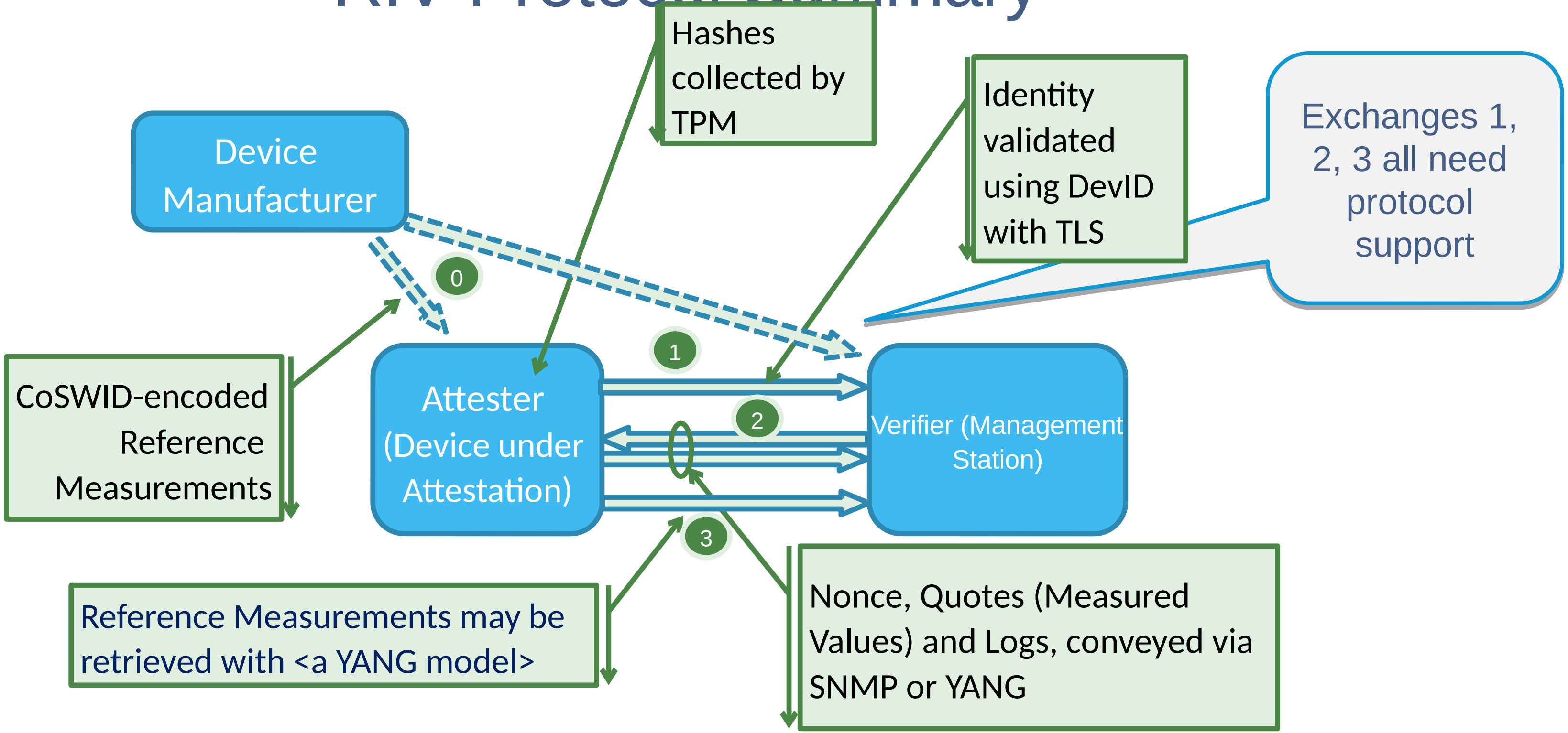
Signed PCR hashes are retrieved from the TPM for off-box analysis

Measurements are “extended” into the TPM

What’s So Hard about This?

- Device Health Attestation is dependent on strong device identity
  - No point in attesting the state of a box if you don’t know which one it is!
- It’s inherently multivendor
  - A single vendor can collect the measurements, but to be useful, someone off-box has to ask for the results and evaluate them
- Software configurations are (almost) infinitely variable.
  - Determining if a chain of hashes is “good” or not is harder than “if (a==b)”
  - Common Multi-threaded OSs don’t promise deterministic ordering, complicating hash chain analysis

RIV Protocol Summary



Hashes collected by TPM

Device Manufacturer

Identity validated using DevID with TLS

Exchanges 1, 2, 3 all need protocol support

CoSWID-encoded Reference Measurements

Attester (Device under Attestation)

Verifier (Management Station)

Reference Measurements may be retrieved with <a YANG model>

Nonce, Quotes (Measured Values) and Logs, conveyed via SNMP or YANG

Figure 4: RIV Protocol and Encoding Summary