A Ridiculously Short Intro into Device Attestation

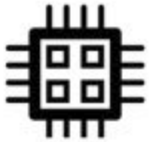
Dimitar Tomov, DesignFirst, Estonia

Ian Oliver, Nokia Bell Labs, Finland

Overview

- What is Attestation? (Ian, 10 mins)
- TPM and Measurements
  - How, What, Why
- Reporting and Quoting
  - Structure and Contained Data
- What to Attest
  - Identity, Firmware, Configuration, etc
- Rules
  - Attest, Verify, Decide
- Example (Dimitar, 10 mins)
  - Time Attestation for Network Monitoring

Measure



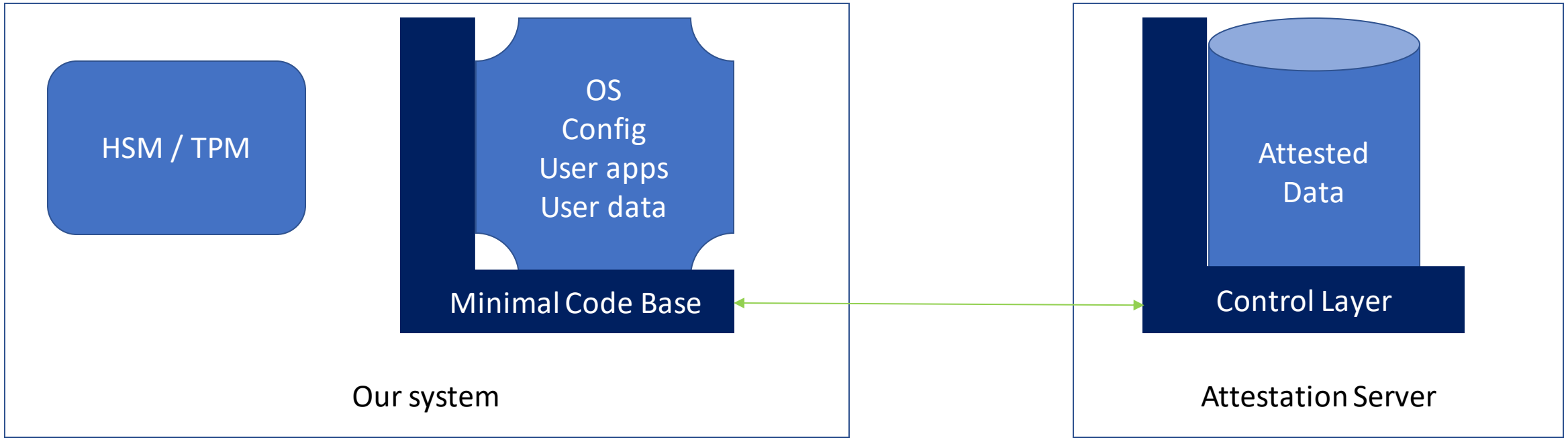
Collect, Store, Report

Verify

Decide

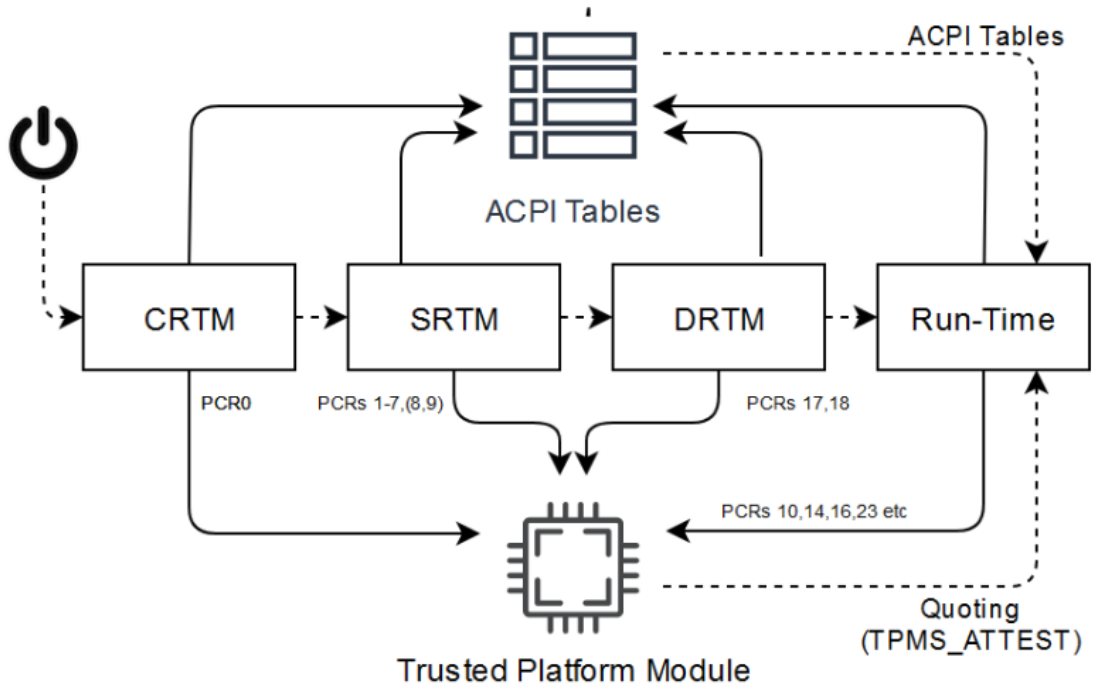
What is attestation?

- The process of providing evidence that something is true

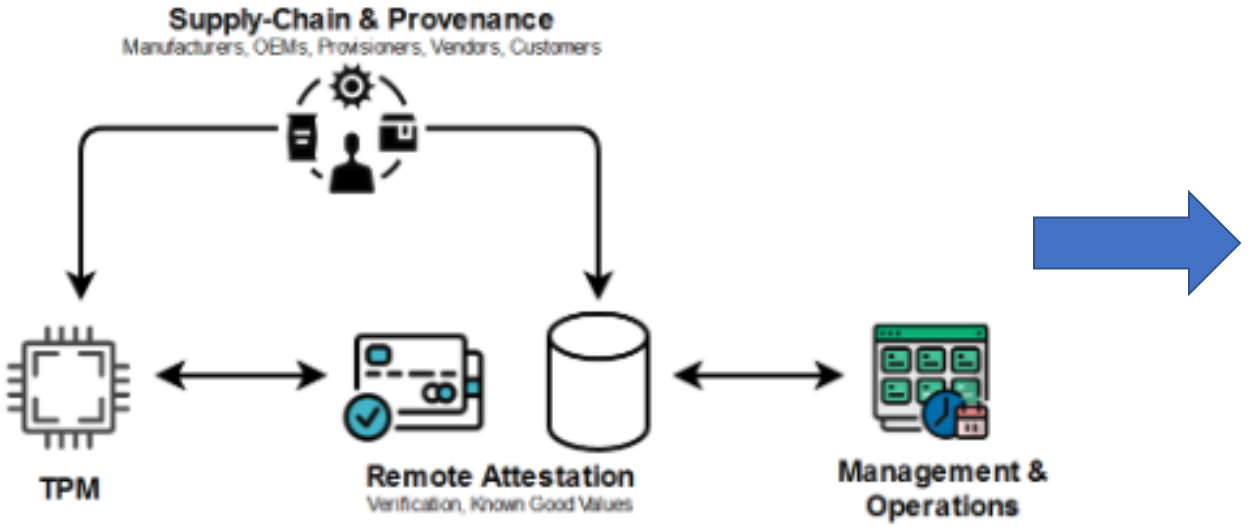


TPM and Measurements

How the **** do things boot and what gets measured?!



Reporting and Quoting




```
pi@cyberlassa:~ $ tpm2_print -t TPMS_ATTEST < q.quot
magic: ff544347
type: 8018
qualifiedSigner: 000ba45ba8b633e5da83b786edd2961c91817f64046ab2ef2e2b8f648973c924532a
extraData:
clockInfo:
  clock: 1186970272
  resetCount: 3631038553
  restartCount: 3882334833
  safe: 1
firmwareVersion: b6f4fbe4535c1b76
attested:
  quote:
    pcrSelect:
      count: 2
      pcrSelections:
        0:
          hash: 4 (sha1)
          sizeofSelect: 3
          pcrSelect: 0f0000
        1:
          hash: 11 (sha256)
          sizeofSelect: 3
          pcrSelect: 0f0000
    pcrDigest: c64cf2032fe712b95e906adfca745c9242785221b0e5a55ebdd062295062a229
```

What to Attest?

```
pi@cyberlassa:~ $ tpm2_print -t TPMS_ATTEST < q.quot
magic: ff544347
type: 8018
qualifiedSigner: 000ba45ba8b633e5da83b786edd2961c91817f64046ab2ef2e2b8f648973c924532a
extraData:
clockInfo:
  clock: 1186970272
  resetCount: 3631038553
  restartCount: 3882334833
  safe: 1
firmwareVersion: b6f4fbe4535c1b76
attested:
  quote:
    pcrSelect:
      count: 2
      pcrSelections:
        0:
          hash: 4 (sha1)
          sizeofSelect: 3
          pcrSelect: 0f0000
        1:
          hash: 11 (sha256)
          sizeofSelect: 3
          pcrSelect: 0f0000
    pcrDigest: c64cf2032fe712b95e906adfca745c9242785221b0e5a55ebdd062295062a229
```

- Quote Type/Magic
- Identity (signature and signer)
- Configuration
- Clock
- Firmware
- Nonce & Arbitrary Data

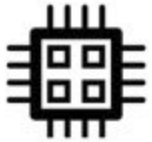
- History

Attestation Rules

TPM 2.0 Quote Rules	Attestation Timeliness Rules	History/Assurance Rules
Is it a quote?	Did the device respond to the quote request in a timely manner?	Has the device changed in any way since the last quote?
Signed and matches the qualified signer?	Did the device process the quote request in a timely manner?	...and for what properties?
Nonce + additional data correct?	Was the response consistent with network latencies (where applicable)?	Does the device verify against the selected rules for its LoA ?
Does the attested value match the known good value?	-	What set of PCRs is required for a minimum LoA?
Is the device running the correct firmware?	-	-
Has the device been rebooted?	-	-
Is the clock increasing correctly?	-	-
Was the device shutdown correctly?	-	-

Who decides if it is trusted?

CRTM/SRTM Measures from BIOS/UEFI/Firmware/ACM etc.



TPM 2.0 Root of Trust for Reporting

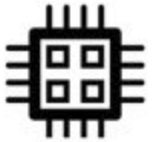
Attestation and Verification Services

End user

Example – Time attestation

- How to attest time?
  - TPM time evidences
- How to attest identity?
  - TPM key based identity
- Choosing a TPM stack(library)
- Time attestation as a timestamp
- Attestation server
- Use cases
  - Data center
  - IoT fleet

Periodically attest



Tamper-proofed HW time

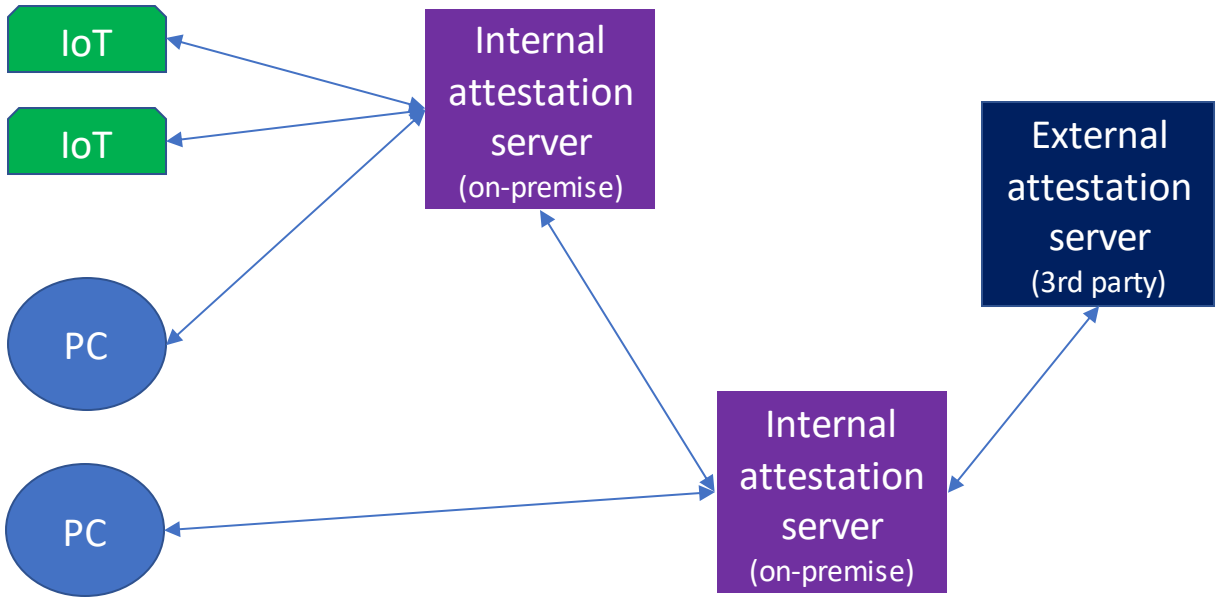
Verify

Decide

Motivation

- Establish identity & temporal trust across devices & systems
  - Attesting the TPM time and clock is a special case
    - Root of Trust for Reporting is the TPM
    - The data being attested is physically internal to the TPM (hint: Vs. Data is fed to the PCRs)
  - Periodic attestation

Equipment or device type	Importance	Sampling period (of attestation)
Critical infrastructure	High	1 minute
User stations, Nodes in the field	Medium	10 minutes
Everything else	Low	1 hour



GetTime vs Quote(PCRs)

- Data is fed to the PCR – How to guarantee what is being measured?

```
trm@DFDEV2:~/wolfTPM$ sudo examples/pcr/reset 16
Demo how to reset a PCR (clear PCR value)
wolfTPM2_Init: success
Trying to reset PCR16...
TPM2_PCR_Reset success
PCR16 digest:
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
        00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
trm@DFDEV2:~/wolfTPM$ sudo examples/pcr/extend 16 /usr/bin/zip2
Demo how to extend data into a PCR (TPM2.0 measurement)
wolfTPM2_Init: success
Hash to be used for measurement:
8C8E79CCF51C391E840FBC24E86E1FD261AC80516EBB52F09A28E594FBA25844
TPM2_PCR_Extend success
PCR16 digest:
        2b bd 54 ae 08 5b 59 ef 90 42 d5 ca 5d df b5 b5 | +.T..[Y..B..]...
        74 3a 26 76 d4 39 37 eb b0 53 f5 82 67 6f b4 aa | t:&v.97..S..go..
trm@DFDEV2:~/wolfTPM$
```

GetTime vs Quote(PCRs)

- TPM2_Quote gives an evidence
- But who creates the evidence?
- We need secure environment
- We need secure application

  Vs

- TPM2_GetTime gives an evidence
- The evidence is created completely internally to the TPM
- No need of secure environment

Note: In both cases an action is needed when a fresh evidence is not received. It would be a form of denial-of-service(DoS) attack.

```
trm@DFDEV2:~/wolfTPM$ sudo examples/pcr/quote 16
Incorrect arguments
Expected usage:
./examples/pcr/quote [pcr] [filename]
* pcr is a PCR index between 0-23 (default 16)
* filename for saving the TPMS_ATTEST structure to a file
Demo usage without parameters, generates quote over PCR16 and
saves the output TPMS_ATTEST structure to "quote.blob" file.
trm@DFDEV2:~/wolfTPM$ sudo examples/pcr/quote 16 report.signed
Demo of generating signed PCR measurement (TPM2.0 Quote)
wolfTPM2_Init: success
TPM2_CreatePrimary: 0x80000000 (314 bytes)
wolfTPM2_CreateEK: Endorsement 0x80000000 (314 bytes)
TPM2_ReadPublic Handle 0x81000200: pub 282, name 34, qualifiedName 34
wolfTPM2_CreateSRK: Storage 0x81000200 (282 bytes)
TPM2_StartAuthSession: sessionHandle 0x3000000
TPM2_Create key: pub 280, priv 212
TPM2_Load Key Handle 0x80000002
wolfTPM2_CreateAndLoadAIK: AIK 0x80000002 (280 bytes)
TPM2_Quote: success
TPM with signature attests (type 0x8018):
        TPM signed 1 count of PCRs
        PCR digest:
        71 da 71 c2 ee dd 1f fd 1f 62 f4 a1 ad 2f 63 a8 | q.q......b.../c.
        11 92 53 b5 c6 91 9d a7 68 2f 5d 19 7c 29 28 a2 | ..S.....h/].|)(.
        TPM generated signature:
        7c e5 03 0b 0d 33 e6 ae 6b 99 bf 57 8a 35 02 57 | |....3..k..W.5.W
        bf 49 26 ed ed 90 eb 90 e7 3a 0f c9 40 5a 70 95 | .I&......:..@Zp.
        e0 23 47 ec 2e c4 78 fb b4 ef bb 62 02 ea 18 95 | .#G...x....b....
        db 35 23 9b 41 1b 19 90 c6 b9 81 5e 8c fc 29 b6 | .5#.A......^..).
        e6 03 13 25 66 a3 ea c2 5e ef 38 c4 75 25 e5 ed | ...%f...^.8.u%..
        c8 8b 39 72 19 95 34 fd 2a a5 8f 50 50 6b 28 41 | ..9r..4.*..PPk(A
        7d 62 e0 af 71 95 33 1b 96 6c f5 3e 00 fe 42 61 | }b..q.3..l.>..Ba
        89 e5 b5 88 ba eb 03 9f 3b 58 e8 76 85 c5 2a 43 | ........;X.v..*C
        10 19 f2 23 07 ba f3 c5 32 29 64 73 a9 5c 47 6a | ...#....2)ds.\Gj
        98 02 eb f2 87 43 45 05 5c 97 ec 89 d6 89 b9 34 | .....CE.\......4
        59 ef 72 08 2d dd e8 74 a8 24 a8 e0 00 56 ba ee | Y.r.-..t.$...V..
        8b 74 50 3b 82 87 52 df ae f9 99 12 f7 18 54 b5 | .tP;..R.......T.
        d7 c2 88 c5 23 07 29 46 32 67 3f 93 81 ca e3 88 | ....#.)F2g?.....
        3f f7 2a 41 43 70 5d 34 c1 34 4e f4 4c d0 00 40 | ?.*ACp]4.4N.L..@
        88 c1 90 5d b8 00 94 4b 58 4a ff 7c cb 21 92 d2 | ...]...KXJ.|.!..
        8d a9 7d 56 68 ec 84 6b 0d 9f a6 80 39 31 01 a0 | ..}Vh..k....91..
TPM2_FlushContext: Closed handle 0x80000002
TPM2_FlushContext: Closed handle 0x80000000
trm@DFDEV2:~/wolfTPM$
```

Choosing a TPM stack(library)

- Mature stacks vs New stacks

TPM stack	Interface(s)	Attestation example	Embedded Systems use
Infineon/Intel TSS	TCG spec. ESAPI, (soon) FAPI	No. Separate project, "CHARRA" by Fraunhofer	Yes for Linux-based systems
IBM TSS	Own rich API (ESAPI like)	Yes. "IBM open-source attestation server(ACS)"	Yes for Linux-based systems
New **Google Go-TPM**	1:1 TPM commands + mild layer on top	Yes. "Go-Attestation"	Needs Golang for non-Linux embedded system
New **WolfSSL WolfTPM**	Own rich API (wrappers) 1:1 TPM commands	Yes. Signed timestamp and local attestation	Baremetal and Linux-based

How to attest time?

- Trust the only IC in your system with physical tamper protection
- Use a standard TPM2.0 command TPM2_GetTime

- Get signed evidence of
  - Built-in hardware time
    - Current uptime of the TPM since the **last** power-on
  - Built-in hardware clock
    - Total time the TPM has **ever** been on
  - Reset counter
    - **How many times** the system has been rebooted since a TPM clear (i.e. provisioning)

Time attestation as a timestamp

```
trm@DFDEV2:~/wolfTPM$ sudo examples/timestamp/signed_timestamp
[sudo] password for trm:
TPM2 Demo of generating signed timestamp from the TPM
wolfTPM2_Init: success
TPM2_ReadClock: success
TPM2_CreatePrimary: 0x80000000 (314 bytes)
wolfTPM2_CreateEK: Endorsement 0x80000000 (314 bytes)
TPM2_ReadPublic Handle 0x81000200: pub 282, name 34, qualifiedName 34
wolfTPM2_CreateSRK: Storage 0x81000200 (282 bytes)
TPM2_StartAuthSession: sessionHandle 0x3000000
TPM2_policySecret success
TPM2_Create key: pub 280, priv 207
TPM2_Load Key Handle 0x80000002
wolfTPM2_CreateAndLoadAIK: AIK 0x80000002 (280 bytes)
wolfTPM2_GetTime: success
TPM with signature attests (type 0x8019):
        TPM uptime since last power-up(in ms): 161855302
        TPM clock, total time the TPM has been on(in ms): 4957468013
        Reset Count: 17
        Restart Count: 0
        Clock Safe: 1
        Firmware Version(vendor specific): 0x1000300010000
TPM2_FlushContext: Closed handle 0x80000002
TPM2_FlushContext: Closed handle 0x80000000
trm@DFDEV2:~/wolfTPM$
```

- TPM uptime since last power-up
  - **44 hours 57 minutes**
- Total time the TPM has been on
  - **57 days 8 hours 52 minutes**
- Reset count
  - **17 power cycles**

What is in the TPM signed time evidence?

- Standard TPM-generated attestation block with
  - TCG defined data structure called TPMS_TIME_ATTEST_INFO
  - TPMT_SIGNATURE holding the signature over the data

TPMS_TIME_ATTEST_INFO

firmware version

TPMS_TIME_INFO

time

TPMS_CLOCK_INFO

Clock

ResetCount

RestartCount

Safe

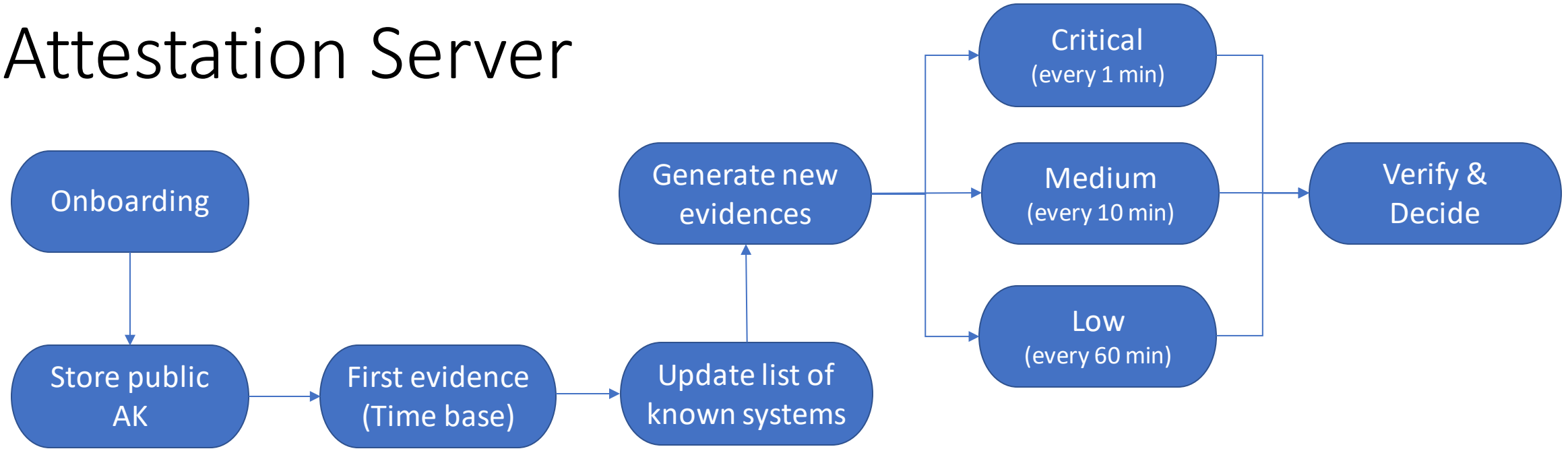
How to attest identity?

- Use a standard TPM2.0 command TPM2_Create

- Create asymmetric key pair known as “Attestation Key”(AK)
  - Private part can be used only by the TPM that created the AK
  - Public part naturally used to verify the evidence signature and decrypt
  - Possible to have a certificate authority and have rolling AK
  - Possible to have anonymous attestation for privacy reasons

NB: The AK is a key generated from the TPM that cannot be migrated between TPMs. Internally, the TPM can use AK only for signing specific TPM-generated structures. No other keys have this property. Therefore, the EK and AK are effectively a unique identity for that TPM.

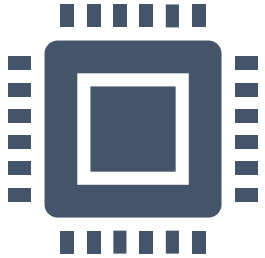
Attestation Server



Decision making

- Slicing between trusted and untrusted information
  - TPM attested data: TPM time, TPM clock, TPM reset counter
  - On-premise network monitoring data
  - Third party data from cloud monitoring

Use cases



Data center

Limited trust in the HW and software vendor

Allows to verify the maintenance periods

Allows to verify the network and monitoring data

End users and customers can have digital trust in their rented or cohosted servers.

IoT Fleet

Improves security for Edge devices with high risk of physical tampering

Helps protect maintenance and battery indicator

Generating rich attestation data on IoT devices is not possible or it is expensive

Set of trust qualities needed

TLDR

- A system can have **trusted qualities** instead of being 100% trusted
- Cross-referencing different types of attestation data can provide evidence for trusted qualities.
- The decision of whether a device is trusted is not responsibility of the attestor and verifier – these just gather and check the evidence.
- TPM time attestation can be trusted without trust in the system.
- Multiple attestation servers (including external attestation servers) are useful for cross-checking attested data.

Contact us for more information

Dimitar Tomov, DesignFirst

- https://tpm.dev
- Twitter *@tomov_eu*

Ian Oliver, Nokia Bell Labs

- https://www.bell-labs.com/usr/ian.oliver
- Twitter *@i_j_oliver*