Remote Attestation

Building trust in things you can’t see

N. Asokan  asokan@acm.org
Andrew Paverd  andrew.paverd@ieee.org
Acknowledgements

(including co-authors of the presenters on papers cited in this tutorial)

Tigist Abera
Ferdinand Brasser
Lucas Davi
Ghada Dessouky
Jan-Erik Ekberg
Kari Kostiainen
Ahmad Ibrahim
Patrick Koeberl

Pekka Laitinen
Thomas Nyman
Ahmad-Reza Sadeghi
Matthias Schunert
Sandeep Tamrakar
Gene Tsudik
Christian Wachsmann
Shaza Zeitouni
Outline

• Remote Attestation in Principle
  – What is remote attestation?
  – What technologies have been proposed?

• Break

• Remote Attestation in Practice
  – What technologies are being used?
  – What challenges remain?
Motivating Example
Motivating Example: IoT

The following message is received:

```json
{
    "name": temperature,
    "value": 23.5,
    "units": Celsius,
    "timestamp": 1430905326.2
}
```

What does it mean?
Motivating Example: IoT

Network adversary: read, modify, falsify communication
Motivating Example: IoT

The following message is received over an authenticated, integrity-protected communication channel:

```
{
    "name": "temperature",
    "value": 23.5,
    "units": "Celsius",
    "timestamp": 1430905326.2
}
```

What does it mean?
Motivating Example: IoT

Network adversary: read, modify, falsify communication
- authenticated, integrity-protected communication

Malware: extract secrets, change state, modify behaviour

Physical adversary: has physical access to device

{ "name": "temperature", "value": 23.5, "units": "Celsius", "timestamp": 1430905326.2 }
IoT Malware

How can we detect remote malware infestations?

Network adversary: read, modify, falsify communication
- authenticated, integrity-protected communication

Malware: extract secrets, change state, modify behaviour

Physical adversary: has physical access to device
Remote Attestation in Principle
Remote Attestation in Principle

Verifier ascertains current state and/or behaviour of prover.

What are the security requirements?
1. Authenticity
   - representation of the real state of the system
Attestation Requirements

1. Authenticity
   - representation of the real state of the system

2. “Freshness”
   - representation of the current state
Trusted Platform Module (TPM)
Authenticated Boot

- Measure and record booted components ("state")
- State can be:
  - bound to stored secrets - sealing
  - reported to external verifier - remote attestation

Authenticated boot
TPM Measurement Process

Platform Configuration Registers (PCRs) store aggregated platform “state” measurement

- Requires a root of trust for measurement (RTM)
- A given state reached ONLY via correct extension sequence
  - “PCR extension rule”

\[ H_{\text{new}} = H(H_{\text{old}} | \text{new}) \]  

- \( H_0 = 0 \)
- \( H_1 = H(0|m_1) \)
- \( H_2 = H(H(0|m_1) | m_2) \)
- \( H_3 = H(H(H(0|m_1) | m_2) | m_3) \)
**TPM Attestation Protocol**

- **Goal**: Check whether the prover is in a trustworthy state

Prover

Measure software state into PCRs

"TPM Quote" $r = \text{Sign}(SK_{\text{AIK}}, c \parallel \text{PCR-values})$

Verifier

Database of acceptable measurements

Attestation Protocol

Challenge $c$

Response $r$

**Attestation Identity Key (AIK)** is a unique keypair whose private key $(SK_{\text{AIK}})$ is **TPM-protected**
Drawbacks of TPM Attestation

- Needs additional hardware and software
- Not suitable for “anaemic” provers
- Covers only the initial loading of software
- Deals with only one prover and one verifier
- Database of acceptable measurements does not scale
Software-Based Attestation
Software-Based Attestation

- Assumes no hardware features to support attestation
  - No secrets on prover (e.g. no AIK)

\[ r = H(\text{mem}, c) \]
Software-Based Attestation

- Pioneer system
  - compute time-optimal checksum of verifier

Software-Based Attestation: Summary

Limitations of timing side channels

– verifier must know exact hardware configuration
– difficult to prove time-optimality
– assumes “adversarial silence” during attestation
– limited to “one-hop” networks
  • requires authenticated channel (e.g. physical connection)
Hybrid Attestation
Hybrid Attestation

Minimal trust anchors: small changes to hardware
Hybrid Attestation: SMART

Minimal trust anchors: small changes to hardware

Read-only Verification code, secure key storage and atomicity of execution of Verification code

Hybrid Attestation: TrustLite & TyTAN

- Execution-Aware Memory Protection Unit (EA-MPU)
  - Access control based on memory request target and origin

Hybrid Attestation: Summary

• Advantages of hybrid approaches
  – Can be used across a network / over an untrusted channel
  – Verifier need not know prover’s exact hardware configuration

• Drawbacks
  – Needs additional hardware support
  – But minimal MCU trust anchors available commercially
    • TrustZone-M (ARM v9), …
Scalability of Attestation
Scalability of Attestation

• Attestation protocols usually assume a single prover
  – but IoT scenarios may involve groups of (many) provers
Scalability of Attestation

• Device *swarms*
  – dynamic topology: nodes move within swarm
  – dynamic membership: nodes join and leave the swarm

[Image of Smart factories and Smart vehicles]
Scalability of Attestation: SEDA

SEDA: Scalable Embedded Device Attestation
- More efficient than attesting each node individually
- Can use any type of measurement process

SEDA: Scalable Embedded Device Attestation. CCS ’15
Scalability: DARPA, SANA, LISA-s

DARPA: Device Attestation Resilient to Physical Attacks
   – Absence detection to detect physical attacks

SANA: Secure & Scalable Aggregate Network Attestation
   – Optimistic Aggregate Signature (OAS) scheme

LISA-s: Lightweight Swarm Attestation schemes
   – Quality of Swarm Attestation (QoSA): binary; count; list; full

Scalability of Attestation: Summary

• Different types of schemes proposed to:
  – improve security (e.g. physical attack resilience) or
  – improve performance (e.g. optimistic aggregation) or
  – improve in functionality (e.g. QoSA)

• What are the real-world application requirements?
Run-Time Attestation
Why Run-Time Attestation?

• Traditional attestation measures binaries at load time
• Cannot capture run-time attacks
  – return-oriented programming
  – control data attacks
Run-Time Attacks

1. **Invoke Function A**

   - **Prover**
   - **Verifier**
   - **Adversary**

   - **Function A** $(i_1, i_2)$
   - **Privileged code**
Control Flow Integrity (CFI)

1. if (cond)
2. then: block A
3. else: block B
4. block C
Run-Time Attacks Without Violating CFI

1. if (cond)
2. then: block_A
3. else: block_B
4. block_C
Control-Flow Attestation (C-FLAT)

**Prover**

1. Application A
2. Execute: $\text{Exec}(A(input))$
3. Measure executed
   - CFG Path: $p = H(\text{Exec}(A(input)))$
   - Compute $r = \text{Auth}(K, c \ || \ p)$
4. Challenge: $c$
5. Response: $r$

**Verifier**

1. Generate CFG: $\text{CFG}(A(*))$
2. Measure CFG Paths: $H(\text{CFG}(A(*)))$
3. Measurement Database
4. Verify $r$

---

C-FLAT: High-Level Idea

- Cumulative Hash Value: $H_j = H(H_i, N)$, where $H_i$ is the previous hash result and $N$ is the current node.

\[
\begin{align*}
H_1 &= H(0, N_1) \\
H_2 &= H(H_1, N_2) \\
H_3 &= H(H_1, N_3) \\
H_4 &= H(H_2, N_4) \text{ or } H_4 = H(H_3, N_4)
\end{align*}
\]

$\rho = H_4$
Handling Loops

- Different loop paths/iterations → many valid hash values
  - Our approach: treat loops as separate sub-graphs

\[
H_1 = H(0, N_1) \\
H_2 = H(0, N_2) \\
H_3 = H(H_2, N_3) \\
H_3' = H(H_2', N_3) \\
\ldots \\
H_4 = H(H_1, N_2) \\
H_5 = H(H_4, N_4) \\
\ldots \\
\]

\[
p = H_5, \langle H_1, \{H_3, \#H_3\} \rangle
\]

\(H_x\) different for each loop iteration

Proof-of-Concept Implementation

- Bare-metal prototype on Raspberry Pi 2
  - Single-purpose program instrumented using binary-rewriting
  - Runtime Monitor written in ARM assembler
  - Measurement Engine isolated in TrustZone-A Secure World

Source: https://github.com/control-flow-attestation/c-flat/blob/master/samples/syringe/syringe-auth.txt

Proof-of-Concept Implementation

Source code at https://github.com/control-flow-attestation/c-flat

LO-FAT

- Low-Overhead Control Flow Attestation in Hardware
  - Same security guarantees as C-FLAT
  - No performance overhead
  - No need for software instrumentation

- Utilizes existing IP building blocks
  - Branch filter used for detecting repeated paths
  - Hash engine for compressing attestation evidence

- Proof-of-concept implementation of main components
  - Targeting RISC-V SoC ("Pulpino")

Run-Time Attestation: Summary

• How can we scale control flow attestation?
  – Better ways to encode/aggregate measurements?
  – Faster/simpler purpose-built hash functions?
  – Attestation of properties rather than measurements?
    • From attestation to checking compliance with a (dynamic) policy?
Property-Based Attestation
Property-Based Attestation

Attest properties of interest instead of program binaries
• scalability of maintaining acceptable measurements

Use a trusted third party to convert from binary evidence to properties

Mid-Point Review: Attestation in Principle

- TPM attestation
- Software-based attestation
  - Pioneer
- Hybrid attestation
  - SMART
  - TrustLite & TyTAN
- Scalable attestation
  - SEDA
  - SANA & LISAs
- Control-Flow Attestation
  - C-FLAT
  - LO-FAT
- Property-based attestation

Which of these are:
1. “Paperware”
2. Testable
3. Deployed
A well-known scientist (some say it was Bertrand Russell) once gave a public lecture on astronomy. He described how the earth orbits around the sun and how the sun, in turn, orbits around the center of a vast collection of stars called our galaxy. At the end of the lecture, a little old lady at the back of the room got up and said: "What you have told us is rubbish. The world is really a flat plate supported on the back of a giant tortoise." The scientist gave a superior smile before replying, "What is the tortoise standing on?" "You're very clever, young man, very clever," said the old lady. "But it's tortoises all the way down!"

- Stephen Hawking, in A Brief History of Time
Remote Attestation in Practice
TPM Attestation

• Where are TPMs used?

• Where is TPM attestation used?

• Main challenge: verifier database scalability
  – Very large number of software packages
  – Frequently changing due to updates
  – Therefore: very hard to maintain whitelists

• Other challenges?
Property-Based Attestation in MirrorLink

- MirrorLink allows use of smartphone services in vehicles
- Car head-unit must enforce driver distraction regulations

http://www.mirrorlink.com

Content Attestation in MirrorLink

- Head unit only allows some types of content while driving
  - Needs to know what content it is asked to render
- **Content Attestation**
  - Defined using TPM structures (part of MirrorLink standard)
  - Initially implemented using On-board Credentials (an early TEE)

MirrorLink Data Attestation

Privacy in Attestation
Privacy in TPM Attestation

- (Recall) Prover provides **TPM-signed quotes** to verifiers

See also [Intel Pentium III Processor Serial Number controversy](https://en.wikipedia.org/wiki/Intel_Pentium_III_Processor_Serial_Number_controversy) (1999)
Privacy in TPM Attestation

• Solution: use different attestation key pairs

  – **Endorsement Key (EK)**
    • One EK per TPM
    • Certified by manufacturer
    
    \[\text{Used to prove this is a real TPM}\]

  – **Attestation Identity Key (AIK)**
    • (Virtually) unlimited number of AIKs
    • Certified by a Privacy CA or through Direct Anonymous Attestation (DAA)
    
    \[\text{Used during attestation}\]
Privacy Certificate Authority (PCA)
Privacy Certificate Authority (PCA)

Verifier 1

From: Clark Kent
Sign(SK_{AIK1}, PCRs),
Cert(PCA, PK_{AIK1})

Verifier 2

From: Superman
Sign(SK_{AIK2}, PCRs),
Cert(PCA, PK_{AIK2})

Prover

TPM1

Compromise

Collude
Direct Anonymous Attestation

• Mechanism of certifying AIKs without a trusted third party

• Based on group signature schemes
  – Secure in random oracle model with strong RSA and decisional Diffie-Hellman assumptions
  – Prover controls linkability between signatures
  – Revocation of anonymity intentionally not possible

• Rogue TPMs can be excluded only if private key is known

Direct Anonymous Attestation

**DAA_join:** Protocol between TPM and DAA issuer (e.g. manufacturer) through which TPM obtains a DAA key.

**DAA_sign:** TPM signs an AIK using its DAA key.

**DAA_verify:** Protocol through which TPM proves to a verifier that it has a valid DAA signature on AIK (without revealing DAA key).

---

Privacy in TPM Attestation

• (Recall) Prover provides TPM-signed quotes along with the full list of executed software to verifiers

Concern 1: Infer private information from installed apps
  – Possibility for profiling/discrimination

Concern 2: Track users through “software fingerprints”
  – Negates use of DAA or Privacy CA
Attestation in Trusted Execution Environments (TEEs)
Intel Software Guard Extensions (SGX)

Objective
- Protect a small amount of code and data against all other software (including the OS)

Mechanism
- Processor-enforced isolated execution environment: enclave

Enclave features
- Secure storage (sealing)
- Secure provisioning (remote attestation)
Intel Software Guard Extensions (SGX)

- Enclave runs in user process
- Enclave memory encrypted before leaving CPU boundary
- Ensures confidentiality and integrity of enclave data

F. McKeen, et al. *Innovative Instructions and Software Model for Isolated Execution*, HASP 2013
SGX Remote Attestation

• Verifier database scalability
  – Only enclave code and configuration are attested

• Privacy
  – Limited amount of code attested
  – Enhanced Privacy ID (EPID)
Enhanced Privacy ID (EPID)

A DAA scheme with enhanced revocation capabilities

- Same privacy guarantees as DAA
  - Also assumes random oracle model with strong RSA and decisional Diffie-Hellman assumptions

- Improved revocation capabilities
  - Revocation possible even if private key not publically known

SGX *Local Attestation*

- Enclave1 sending message to Enclave2
- Authenticated by CPU

![Diagram of SGX Local Attestation]

---

*I. Anati, S. Gueron, S. Johnson, V. Scarlata. Innovative Technology for CPU Based Attestation and Sealing, HASP 2013*
SGX Remote Attestation

I. Anati, S. Gueron, S. Johnson, V. Scarlata. Innovative Technology for CPU Based Attestation and Sealing, HASP 2013
Kinibi on ARM TrustZone
Kinibi on ARM TrustZone

- Single hardware-enforced TrustZone TEE per platform

- Kinibi trusted OS:
  - manages trusted applications
  - isolates them from each other

S. Tamrakar, J-E Ekberg, P. Laitinen, *On Rehoming the Electronic ID to TEEs*, TrustCom 2015
Remote Attestation in Kinibi

ARM TrustZone hardware extensions

\[
m_{\text{attest}} : \text{Enc}(K_{\text{TEE}_i}, \text{UUID}_{\text{TA}}, h(\text{TA}), c, \ldots)
\]

S. Tamrakar, J-E Ekberg, P. Laitinen, *On Rehoming the Electronic ID to TEEs*, TrustCom 2015
Common use case: Key Attestation
Key Attestation

How to attest that a key is protected by hardware?
  – Must also prevent linkability between keys

• **TPM\_CertifyKey** command
  – Non-migratable TPM key certified using AIK
  – Subject Key Attestation Evidence (SKAE) extension X.509 cert.
  – Now [supported by Windows Server](#) (Feb 2017)

• Using normal **SGX attestation**
  – Verifier checks that enclave generated key securely
Android Keystore Attestation

• Available from Android 7.0 (API level 24) onwards
  – But currently few devices with hardware-backed attestation

• Keystore produces an attestation certificate for key pair
  – Standard X.509, signed by on-device attestation key

• On-device attestation keys
  – Injected into device during manufacture
  – Signed by device manufacturer or Google
  – Injected in batches: “minimum 10,000 devices per key”
Summary: Attestation in Practice

• TPM-based Attestation

• MirrorLink Data Attestation Protocol

• Privacy in Attestation

• Attestation in TEEs
  – SGX
  – TrustZone

• Key Attestation
Open Challenges
Open Challenges

• Physical adversary
Open Challenges

- End-to-end attestation

Traditional attestation based on virtualized hardware (e.g. virtual TPM)

Software-based or Hybrid attestation

Traditional hardware-based attestation
Open Challenges

• Attestation Servers?
Open Challenges

• Attestation of the Cloud?
Conclusions

• Increasing need for remote attestation

• Various schemes proposed, developed, deployed

• Building deployable attestation schemes is challenging