(Remotely Establishing) Trust in Keys and Software (→ Attestation)

Bernd Prünster <bernd.pruenster@a-sit.at>
A-SIT Plus GmbH
https://a-sit-plus.github.io

Terminology / Marketing Fluff

- Intel *Software Guard Extensions* (SGX) ⍰ 2021
  AMD *Secure Memory Encryption* (SME) ⚠
  Secure Encrypted Virtualization (SEV) ⚠
- *Trusted Platform Module* (TPM) 👌
- Android Key Attestation 🚀
- Apple *DeviceCheck* + *AppAttest* 👌

Why use TPM/…/AppAttest in Practice?

- *Digital Rights Management* (DRM); i.e. copy protection
- (Remotely attestable) hardware-backed key storage
  - **Ensure non-extractable private keys!**
- Ensure use of physical device
- Ensure authenticity, confidentiality, integrity (≠ CIA triad!)
- Enforce up-to-date OS + applications

Without previously enrolling remote devices!

Agenda

- High-level prerequisites
- Trusted Platform Module
- Trust models (PC vs. mobile)
- Android deep dive
- iOS not-so-deep dive
- Showcase down to the byte level

} How to remotely establish trust in unmanaged devices

High-Level Prerequisites

Prerequisites

- Hardware support
- Trust anchor
- Details depend on trust model!

Hardware Support

Why?

Software-only measures can always be circumvented!

Trust Anchor

See TLS/eID/Banking App…

More details later

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0fc40e88e1aec293ddfbbc5b82c3e294e8c0ed14

Trusted Platform Module

Trusted Platform Module Basics

- TPM is – above all – a **specification**
- TPM <1.2 broken, TPM v1 obsolete ⟶ TPM ≋ TPM v2!
- Present in virtually every PC
- Controllable by user / admin
- Secure Boot!
- Basic crypto functionality
- Standardised API

TPM Crypto Features

- RNG
- Hashing
- Key generation
- Signing
- Encyption
- Tamper-proof storage (<64kB)
- Rate limiting
- Clocks + monolithic counters
- Attestation
- Measured + Secure Boot

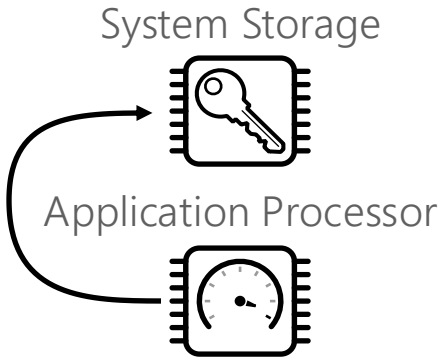
TPM Implementation Summary

- Abstraction for all things crypto
- Standardised API
- *Usually* hardware-backed

Q: How to verify HW-backing?

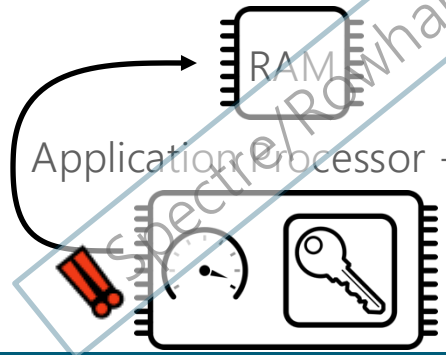
A: Check Certificate Chain!

Virtual/Software TPM



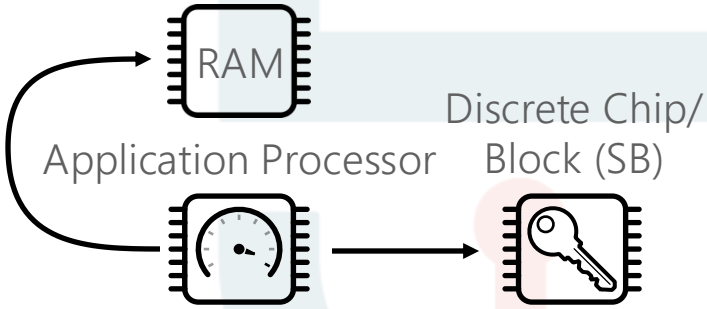
Firmware TPM

TrustZone, PTT, fTPM



Spectre/Rowhammer

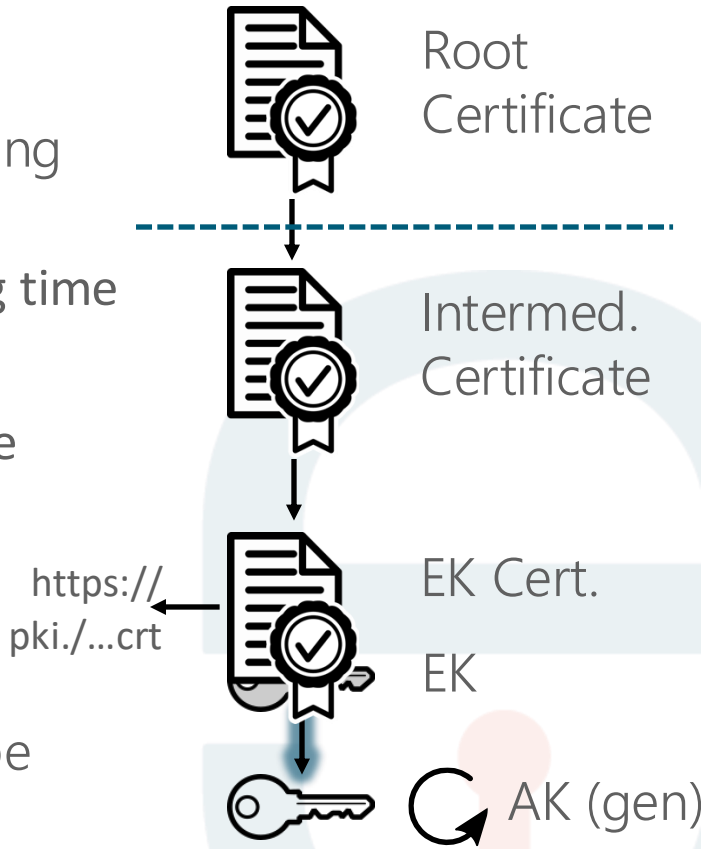
Integrated/Discrete TPM



TPM Attestation

- *Endorsement Key* (EK) burned-in at manufacturing time
  - Certificate (chain) burned-in at manufacturing time
  - Root Cert trusted out-of-band
  - URI to issuer cert encoded inside EK certificate
- *Attestation Key* (AK) generated inside TPM
  + bound to EK
- AK can sign other, freshly generated keys
- Root certificate indicates TPM manufacturer/type

  → Now we can trust the TPM!



TPM Measurements

- *Measurement* ≈ recodring Hash of Firmware/Software
  - Record hash and compare with known-good hash from register
  - Make certains keys available only if measurements check out
- Verify integrity of FW/BL/OS/…
  - *Secure Boot* → Verify UEFI against trusted keys
  - *Measured Boot* → Verify booted OS (kernel)
  - Managed devices → Prohibit using tampered devices
  - Decrypt volumes/disks only in known good state

Trust Models (PC vs. Mobile)

PC Trust Model

- OS is under attacker's control
- User wants+needs+has root access
- User is the enemy
- Devices untrusted (IO!)
- Main memory part of attack surface
- Only TPM API trustworthy
- Extending trust to OS and apps challenging

→ Realistic possibilities for secure applications?

Android Platform Security Model[1]

- TPM-style hardware crypto
- Verified Boot
- Strong sandboxing (app isolation)
- Strict permission enforcement
- No root access for users
- Attestation for device + keys + OS + apps!
- Elevate OS + app to trusted computing base (TCB) through attestation
- Implementation details → *Compatibility Definition Documents*[2]

Apple: Same same, but different

[1]https://arxiv.org/abs/1904.05572

[2]https://source.android.com/docs/compatibility/cdd

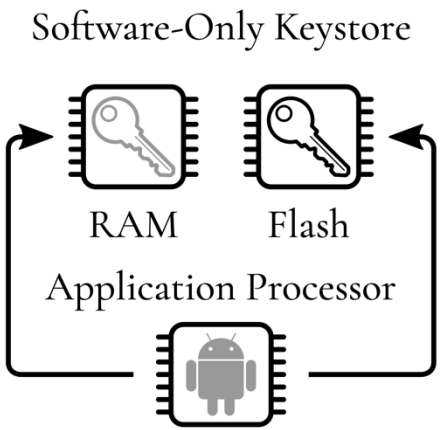
Android Deep Dive

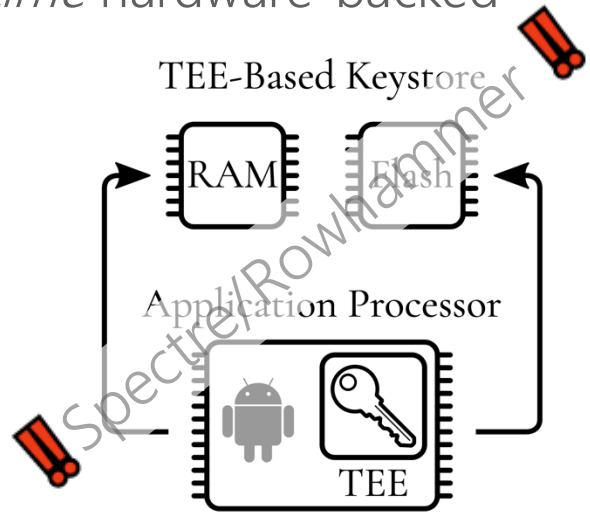
Android Keystore

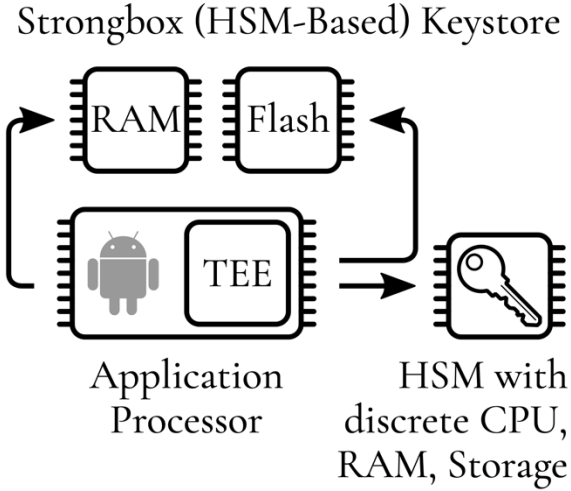
- Abstraction for all things crypto
- *Java Cryptography Architecture* (JCA) API
- *Most of the time* hardware-backed

Q: How to verify HW-backing?

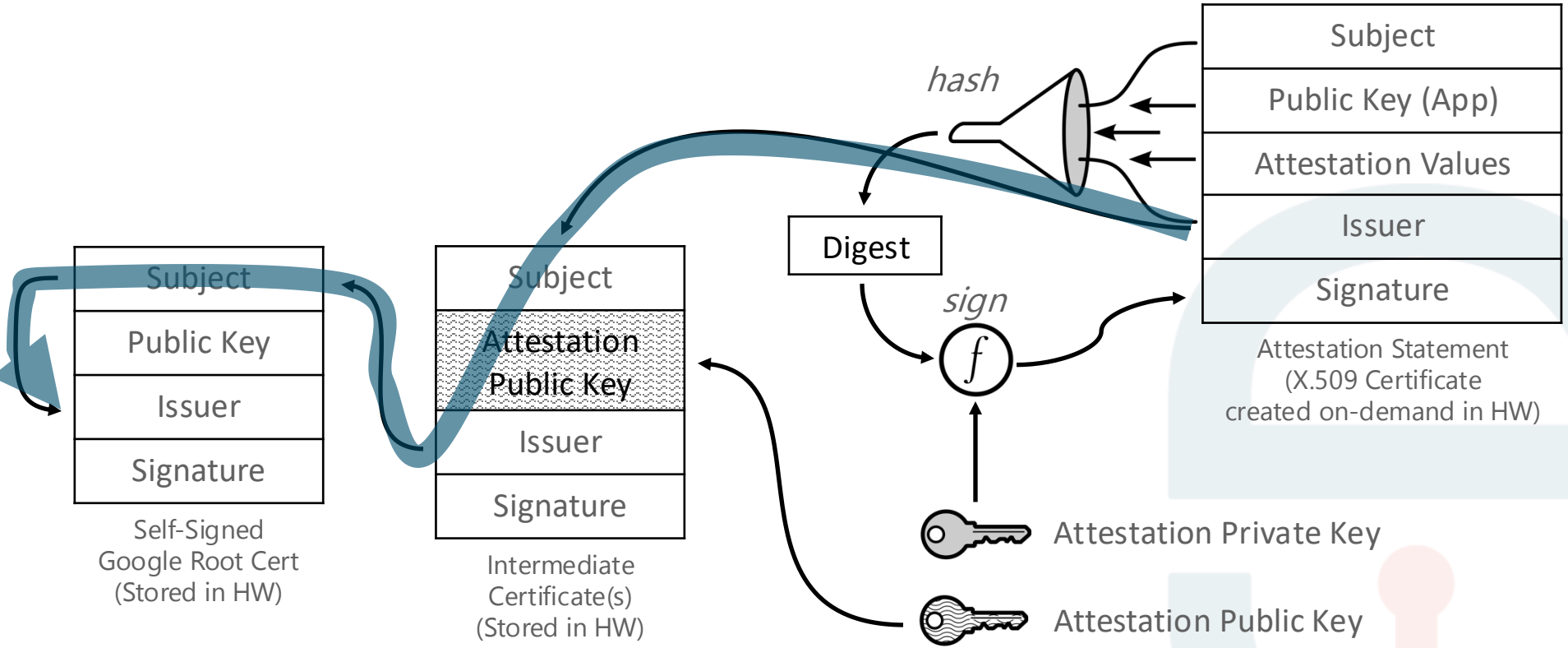
A: Check Certificate Chain!



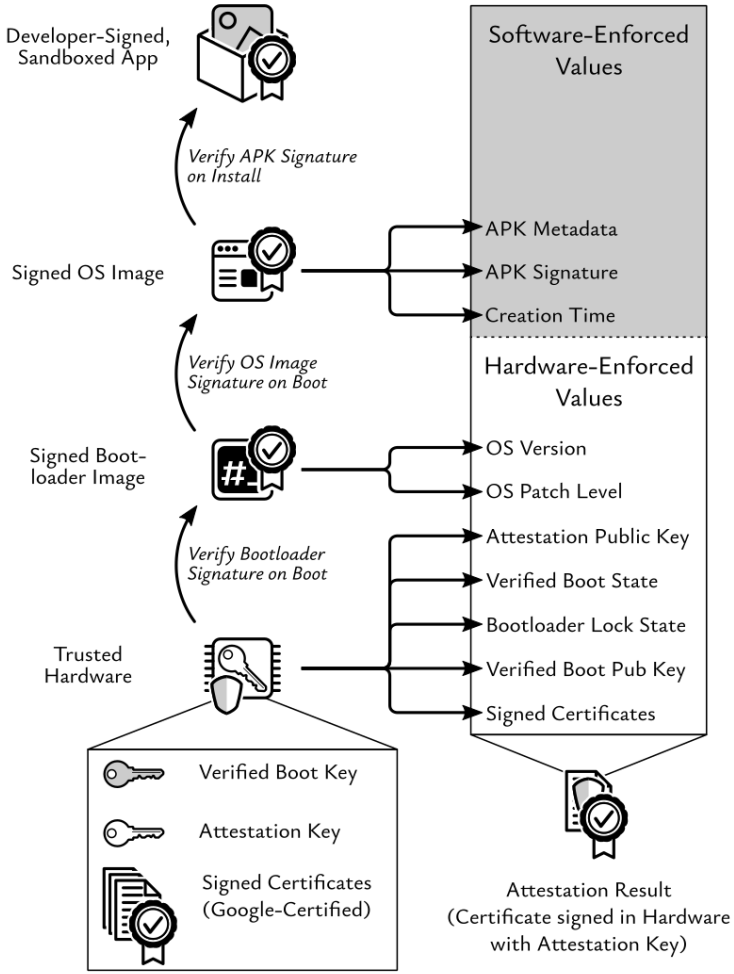




Android Attestation Chain of Trust

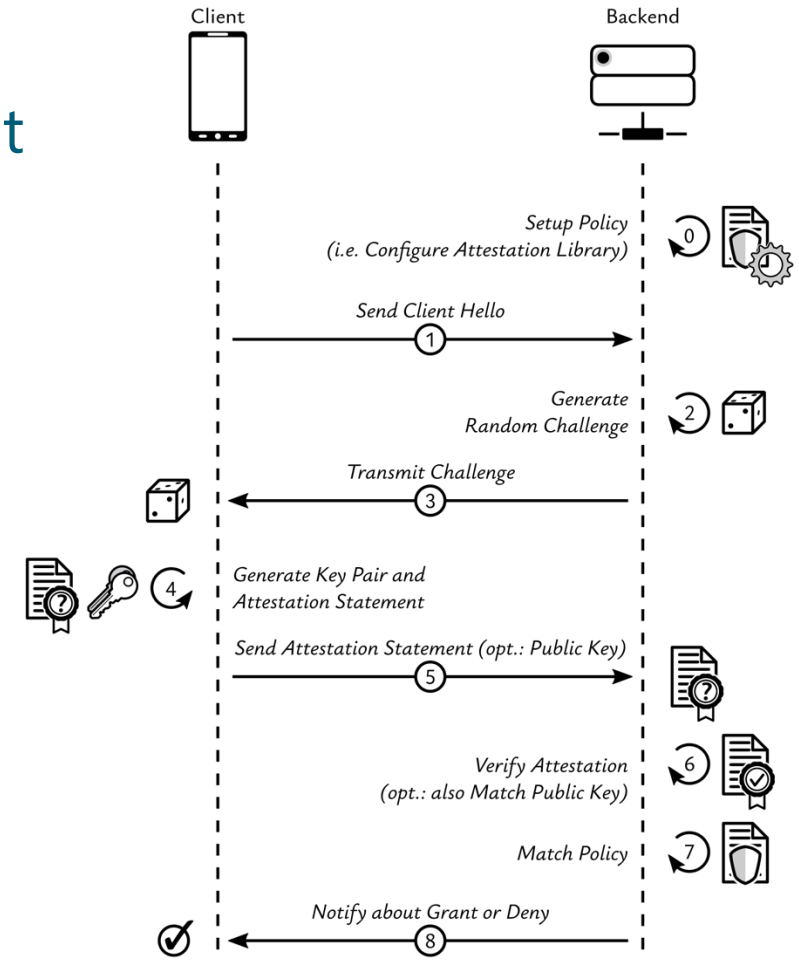


From Verified Boot to Trusted Applications



Creating an Attestation Statement

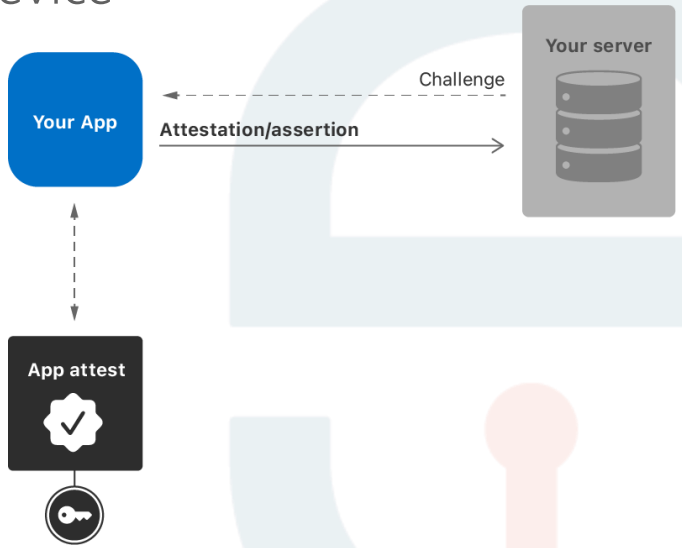
https://github.com/
a-sit-plus/warden



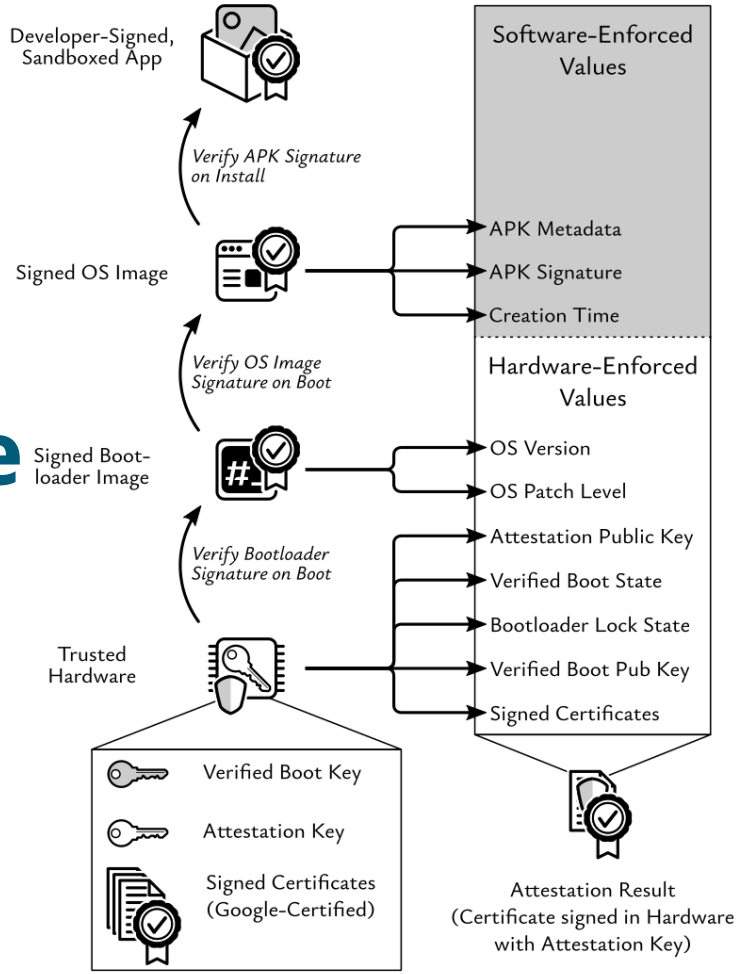
Apple AppAttest/DeviceCheck

- Device connects to Apple infrastrucutre
- Apple servers do undocumented attestation magic
- Apple server issues attestation statement to device
- Device sends statement to service
- Also dependent on cryptographic hardware
- Cannot use attestation key!

Controlled and run by Apple →



Demo Time



https://source.android.com/docs/security/features/keystore/attestation#schema