Cursul #4

Trusted Computing in Embedded Systems

3/10/2025

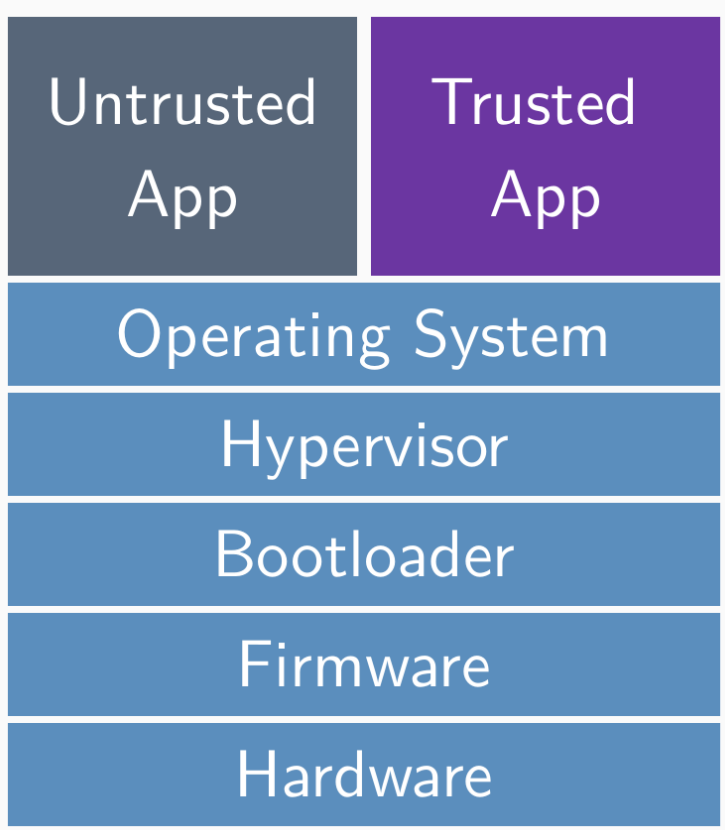
Outline

- Intro: Trusted Computing
  - Technologies (TPM, TEEs)
  - Secure Boot, Attestation & Provisioning
  - TEEs: ARM TrustZone
- Security Threat Models
- Applications & Case Studies


Trusted Computing

Trusting technology

- **TCB**: Trusted Computing Base
  - Hardware, Bootloader, OS
- Problem:
  - Bugs!
  - Kernels: ~20 mil. LoC
- Security Approach:
  - Minimize the trusted components
  - Start from a good Root of Trust (HW)
    - Chain measurements for integrity
    - Digital signatures / storage using protected keys



Trusted Computing

- "*For years Bill Gates has dreamed of finding a way to make the Chinese pay for software, TC looks like being the answer to his prayer.*"
- Trusted Computing Group:
  - TPM specification (1.x – 2009, 2.0 – 2014)
- Applications:
  - Boot integrity
  - Authentication
  - Code/data protection / isolation
  - *Walled Gardens / Security by Obscurity*

Security Functions

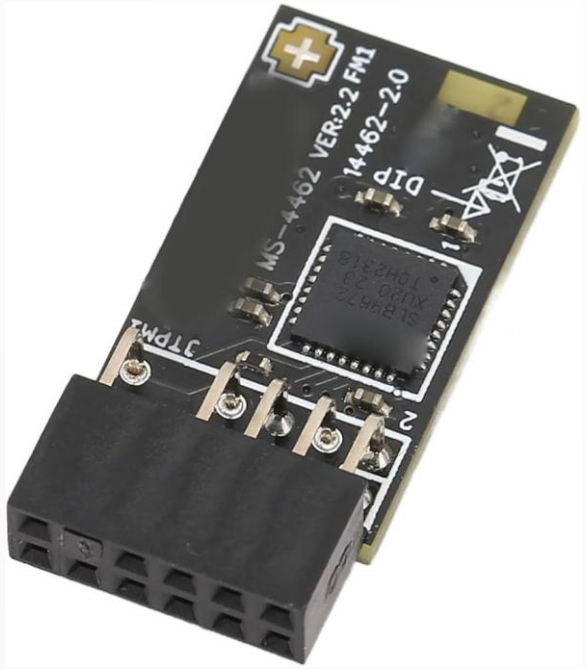
- Integrity Measurements
- Secure Storage
- Isolated execution
  - Trusted Execution Environments (TEE)
- Attestation (remote / local)
- Secure Provisioning
- Trusted Path

Trusted Technologies

- General purpose computing:
  - **Trusted Platform Module**
  - ~~*Intel TxT, Intel SGX* – *0xD34D*~~
- Mobile & Embedded:
  - **ARM TrustZone**
  - Apple’s proprietary Secure Enclave
- Server/Cloud:
  - AMD Secure Encrypted Virtualization
  - Intel Trust Domain Extensions

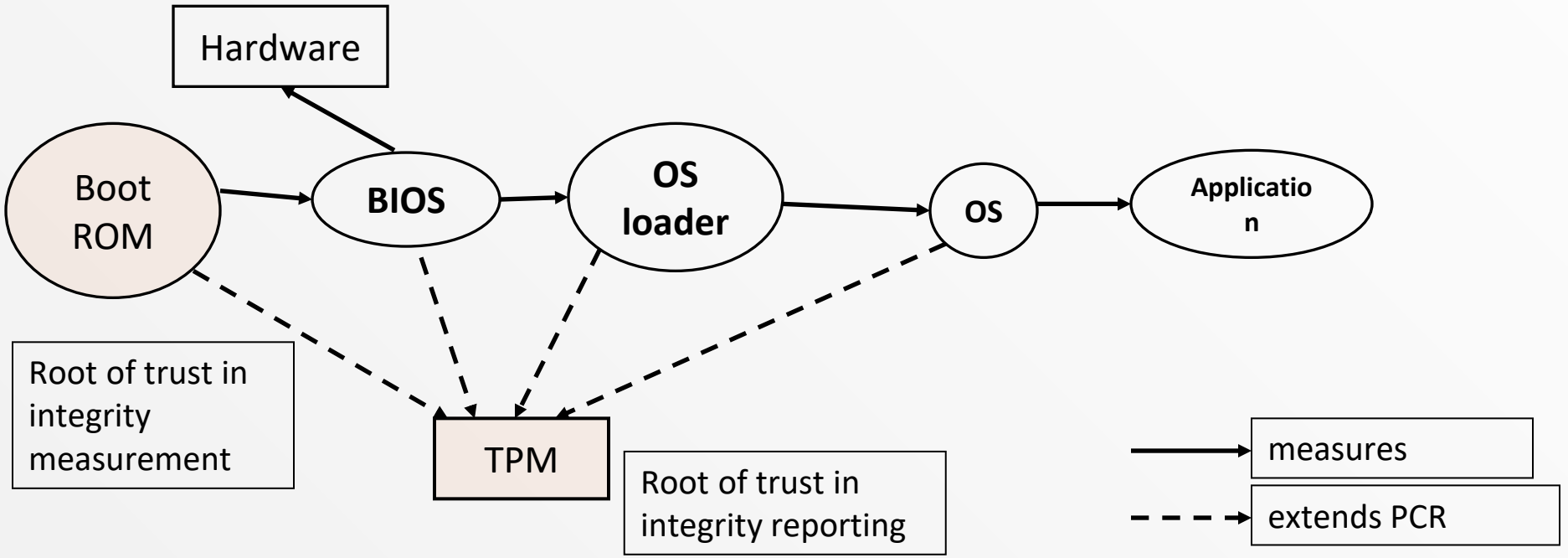
Trusted Platform Module

- Cryptographic Coprocessor (TPM v2 standard)
- Multiple Roots of Trust:
  - Root of Trust for Measurement (RTM) – hash algorithm
  - Root of Trust for Storage (RTS) – crypto ops. with Storage Root Key (SRK)
  - Root of Trust for Reporting (RTR) – identity & attestation using Endorsement Key (EK)
- Manufacturers: Infineon, Nuvoton, STMicro, Microchip etc.
  - Commonly interfaced using TPM LPC or SPI (Serial Peripheral Interface)
  - AMD + Intel CPUs have hidden microcontrollers core implementing firmware TPM



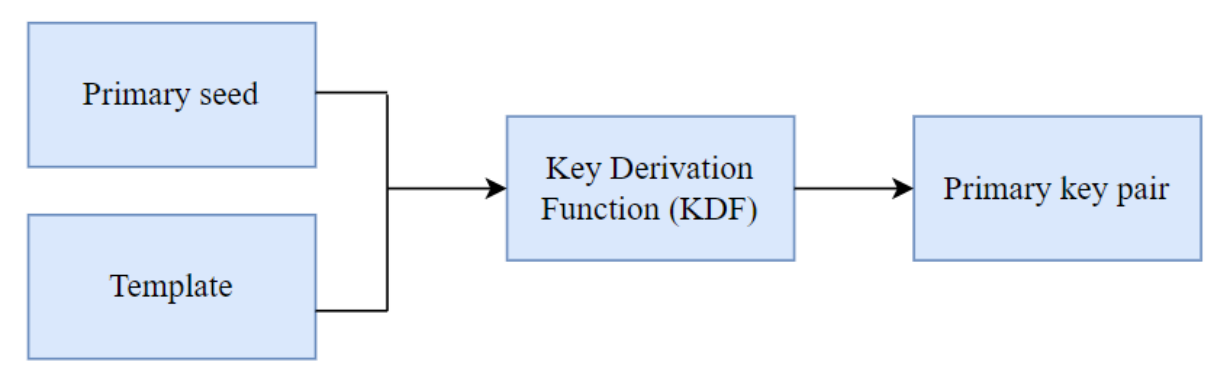
Trusted Boot

- Uses TPM's Platform Configuration Registers
- After boot, PCRs contain hash chain of booted software



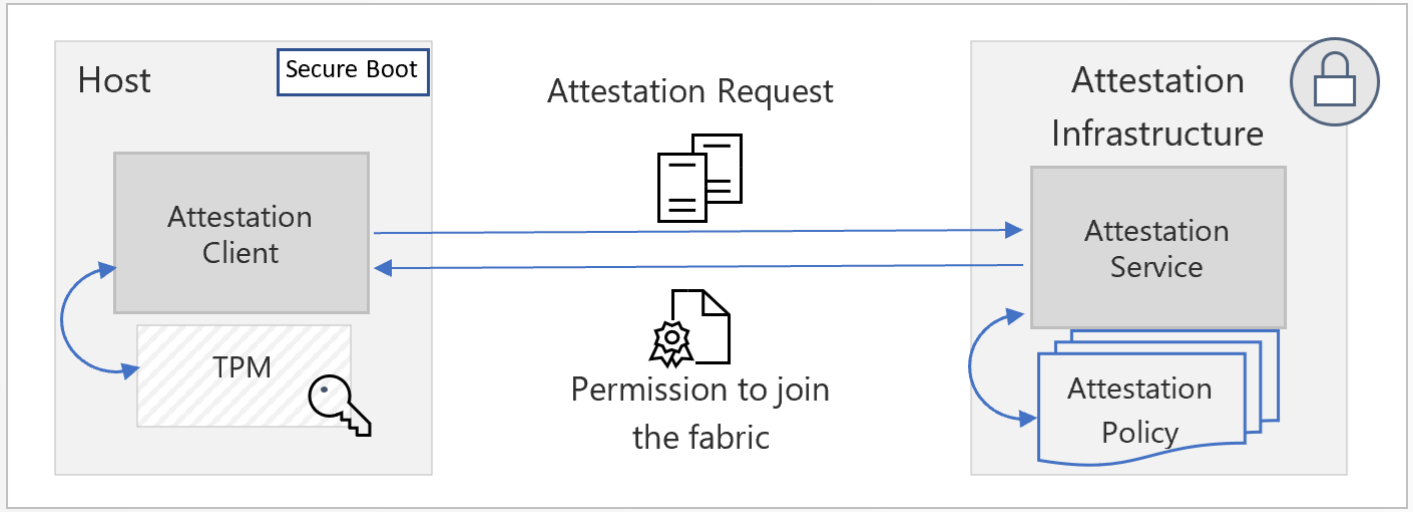
Secure Storage

- Migratable keys: Non-Volatile Memory
  - can be written / read by software if they pass integrity checks
- Non-migratable keys (sealing):
  - Keys can only be used for TPM crypto ops
  - Derived from the Storage Root Key



Attestation [1]

- System boots, measures all code & data into PCR;
- Client requests the TPM to sign using the Attestation Key;
- Client supplies report to remote service for secret provisioning.

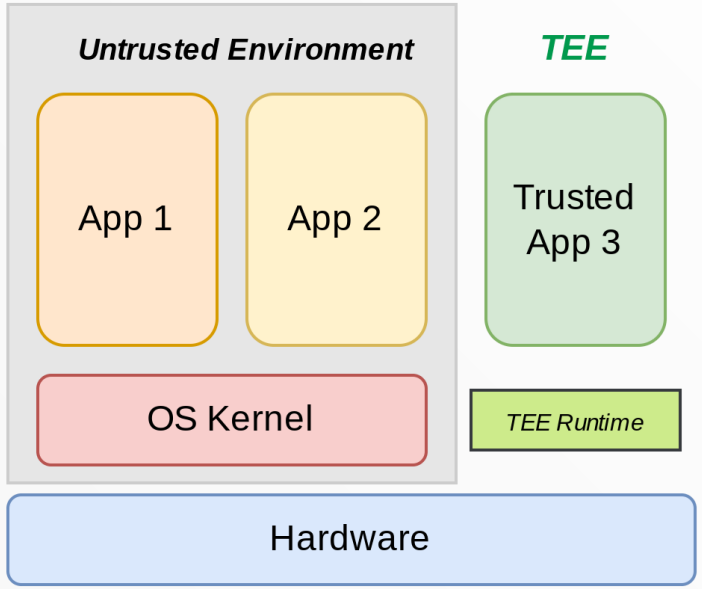


[1] https://tpm2-software.github.io/tpm2-tss/getting-started/2019/12/18/Remote-Attestation.html

Trusted Execution

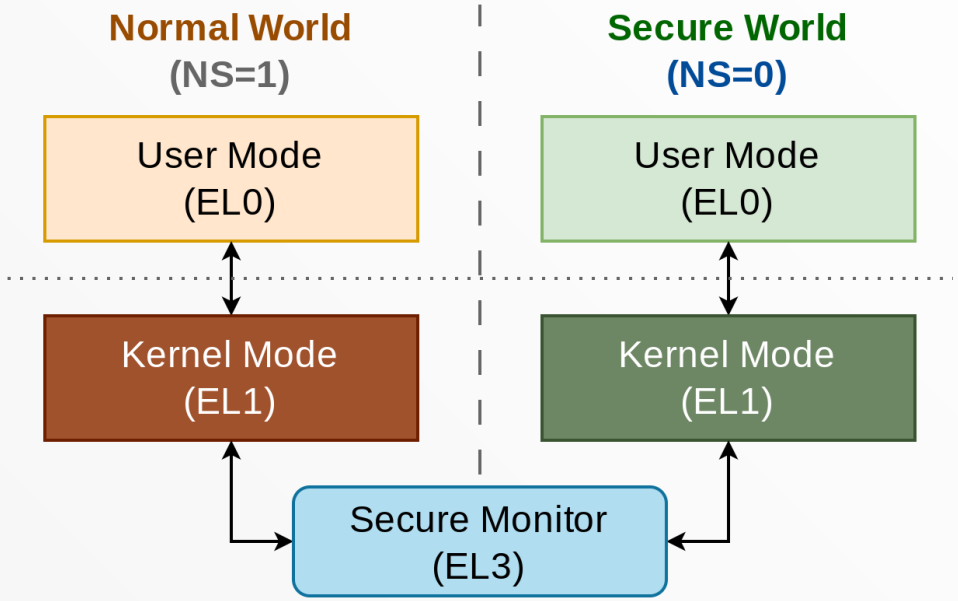
Trusted Execution Environments

- Isolate memory & execution state
  - guaranteed protection even from higher privileged attackers
- Implementation approaches:
  - Software-based de-privileging (e.g., binary instrumentation) – performance & complexity issues
  - Hardware-assisted (e.g., CPU architectural extensions / dedicated processing core)
- HW platforms: Intel TxT, ARM TrustZone, Intel SGX (RIP), AMD SEV, Intel TDX



ARM TrustZone

- RE: user / kernel space – ARM Exception Levels
- Separation: Secure / Non-Secure Worlds
- Components:
  - Secure Monitor (BL31: ATF / TF-A);
  - Secure Micro-OS (BL32: OP-TEE);
  - Trusted Applications (EL0 – userspace);
- Applications: smartphones (virtual cards, password managers etc.), trusted IoT devices (e.g., governmental / military infrastructure) etc.



Secure Monitor

- Secure Monitor – arbiter between the two contexts:
  - Receives requests from either Secure / Non-Secure world (SMC, similar to system calls)
  - Saves execution context (CPU flags & registers);
  - Copies call parameters between the two memory contexts
  - Jumps to execute the requested service (to either Normal **OR** Trusted kernel);
  - Copy return values back & restore execution context
- Receives interrupt requests from secure peripherals and dispatches them to the trusted kernel

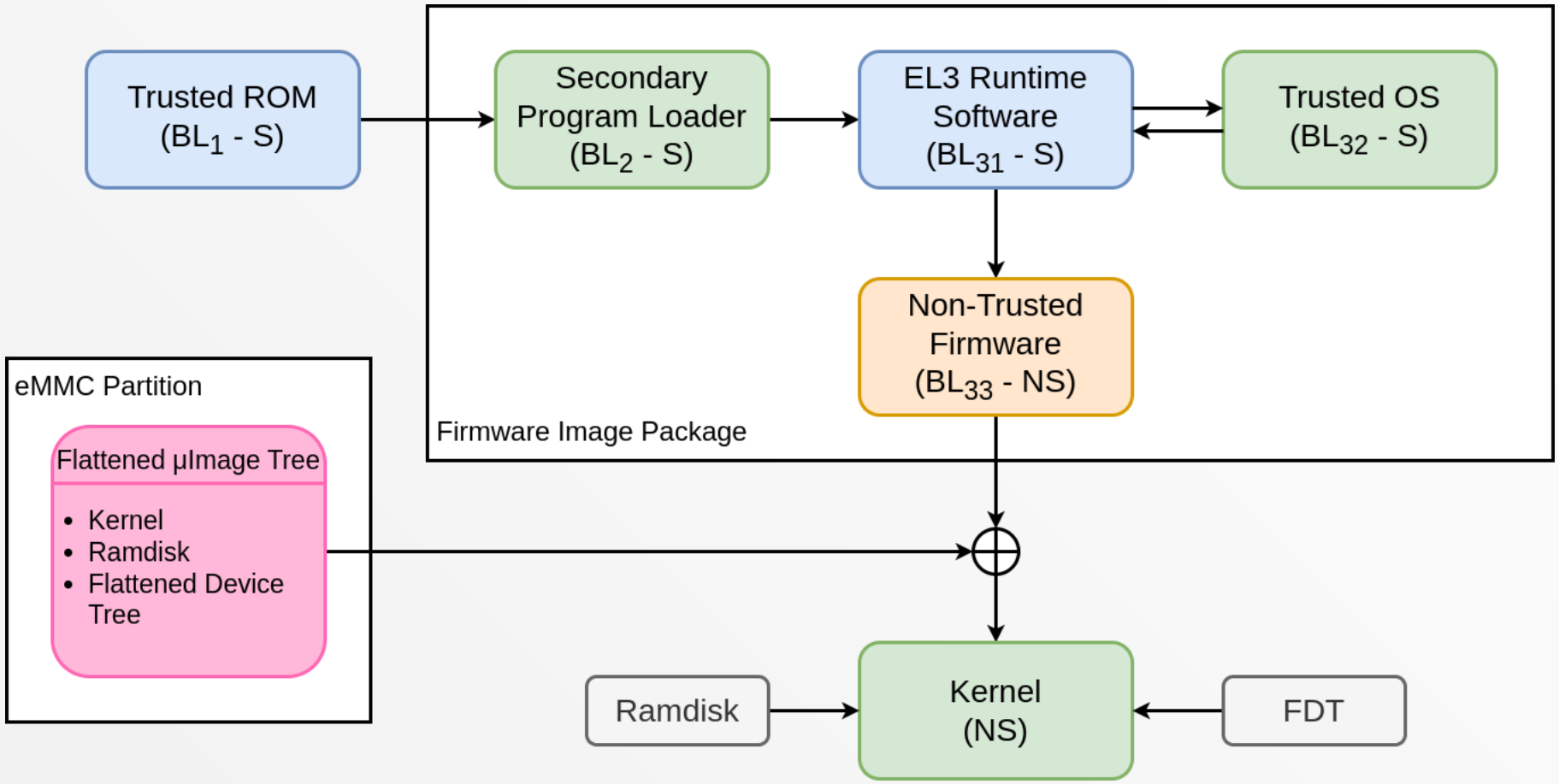
OP-TEE

- Open-source Portable TEE (mainly for ARM TrustZone)
  - https://optee.readthedocs.io/en/latest/general/about.html
  - TEE specifications by Global Platform
- Components:
  - Secure OS
  - Normal (Linux) OS drivers (in mainline!)
  - Trusted libraries (e.g., cryptographic routines)
  - Host OS libraries + client (for interacting with trusted apps)
  - Trusted Application Software Development Kit + examples!
- Trusted kernel size: 244KB!

Trusted Applications

- Split code
  - Trusted Application
  - Untrusted components (under a Linux host)
- Rich OS (Linux) implements almost all drivers + services (networking, filesystem, scheduling etc.)
- Host program calls TA routine when it needs to process sensitive data
- TAs may system call (SMC) everytime it needs its system calls
  - Results must be checked / authenticated, since anyone untrusted could become malicious at any time

ARMv8 Secure Boot Process



Case Studies

Real World Applications & Vulnerabilities

T.C. Applications (1)

- Mobile phones (ARM):
  - Android: Trusty TEE + proprietary solutions (Samsung Knox, Huawei's iTrustee 😂 etc.)
  - Apple Secure Enclave: SoC co-processor with L4 microkernel
  - Google Titan M (TPM-like, but enhanced / non-standard)
- Embedded / IoT / Industrial:
  - ARMv8 Cortex-A firmwares must bundle ARM Trusted Firmware
  - ARMv8 Cortex-M: started introducing TrustZone for microcontrollers, too! (e.g., STM32L5, RP2350)

T.C. Threat Model

- Adversaries: remote / local (HID, physical etc.)
- Attacks:
  - software vulnerabilities (e.g., memory corruption)
  - communication interface attacks (MitM / replay)
  - side channels
  - hardware attacks (Evil Maid, tampering, destructive key extraction etc.)

Software Jailbreaks

- Local Privilege Escalation + Defeating Secure Boot 🥹
- Examples:
  - iOS (16.x): Fugu15/Dopamine (untethered, uses 4 exploits [1])
  - Kindle: WinterBreak [1] (2025, JS app privilege escalation :)))
  - Android Rooters
- Keeps getting harder with Secure Boot :(

[1] https://theapplewiki.com/wiki/Fugu15
[2] https://kindlemodding.org/jailbreaking/WinterBreak/

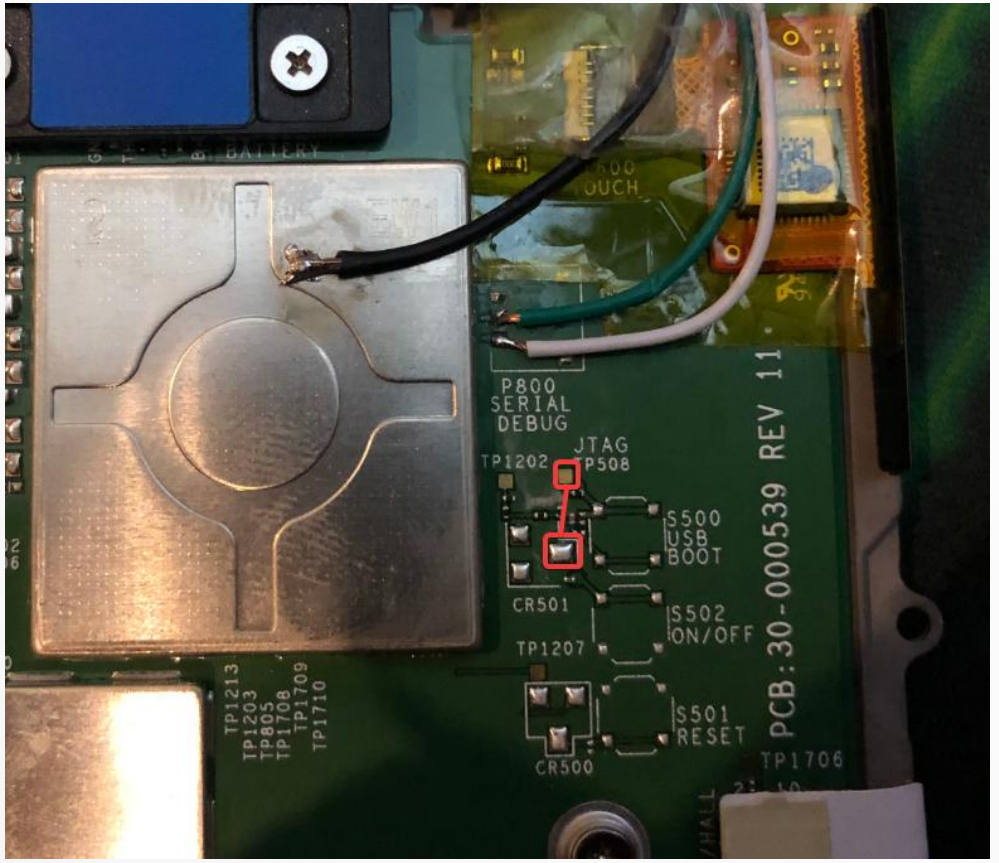
Firmware vulnerabilities

- TPM firmware vulnerabilities:
  - Infineon TPMs: CVE-2017-15361 (weak RSA key pair generation)
  - TPM 2.0 modules: CVE-2023-1017 (out-of-bounds read) and CVE-2023-1018 (out-of-bounds write)
- Titan M [1]:
  - Reversed proprietary comm protocol, found buffer overflow via fuzzing => ROP exploit!
- Apple Secure Enclave & T2 chips (2020 exploit [2])
  - demoed by China pwn team, unknown details :(

[1] https://blog.quarkslab.com/attacking-titan-m-with-only-one-byte.html
[2] https://www.gizmochina.com/2020/08/03/apple-secure-enclave-exploit/

Hardware Jailbreaks

- Routers:
  - identify UART RX/TX pins
  - cancel bootloader
- Kindle Popcorn [1]
  - Freescale i.MX uses bootloader enabled on BLx pins state



[1] https://kindlemodding.org/jailbreaking/Legacy/Popcorn

Side Channel Attacks

- Hardware behavior can give out extra information!
  - Extract crypto parameters (key)
- Cache / timing, EM / power analysis, acoustic etc.
- Speculative execution: Spectre / Meltdown
- Mitigation: software (hardware is too costly!)
  - Constant time/power algorithms
  - Cache invalidation
- External cryptoprocessors (Apple Secure Element, Google Titan M) are immune to those!

Physical Attacks

- Evil Maid
  - TPM: trigger RESET, start over with clean PCRs; applications: extract full-disk encryption keys from laptops
- Firmware readout, debug interface exploits etc.
- Fault Injection:
  - Manipulate inputs / switch off power at specific moments in time to alter firmware execution flow
  - Electromagnetic fault injection (focused EMP)
  - Lasers! (w/o sharks)

MCU Readout Protection

- STM32's RDP: disable debug ports & firmware download
  - Bypassable by applying 3.3V to BOOT0 pin
  - SWD usable while device is under reset
- Nordic (NRF52x) RDP bypass [2]
  - debug allowed, all memory access returns 0, breakpoints OK
  - => reuse ROM code to read all protected flash
- ChipWhisperer (https://chipwhisperer.readthedocs.io/)
  - tool/framework for discovering side channel attack & power glitching attacks on chips

[1] https://medium.com/@LargeCardinal/how-to-bypass-debug-disabling-and-crp-on-stm32f103-7116e7abb546
[2] https://www.emproof.com/bypassing-readout-protection-in-nordic-semiconductor-microcontrollers/

Secure OTP memory: RP2350 MCU

- Raspberry PI's new ARMv8-M microcontroller
- Has small "antifuse" OTP for storing key & read protection
- Security through transparency: RP2350 Hacking Challenge results are in [1]:
  - Power fault injection => guard read retention => debug re-enable
  - Inject voltage => instruction skip => critical secure boot check bypassed
  - Laser Fault Injection (:
  - Focused Ion Beam / Passive Voltage Contrast key extraction 🤖

[1] https://www.raspberrypi.com/news/security-through-transparency-rp2350-hacking-challenge-results-are-in/