KU LEUVEN

Software integrity checks on open platforms

Performing process attestation on user-controlled ARM TrustZone devices

Oberon Swings

KU Leuven

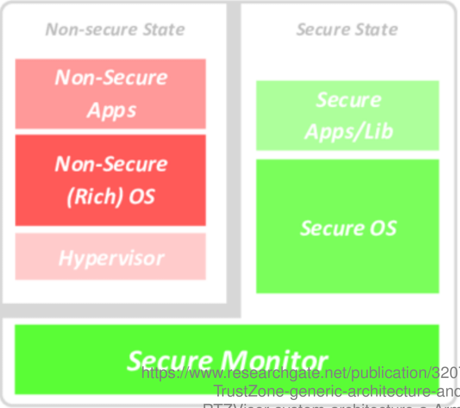
June 30, 2022

Outline

Outline

Context

- Smartphones
  - Usable everywhere
  - Sensitive data
  - Hardware security
- ARM TrustZone
  - Normal World
  - Secure World



https://www.researchgate.net/publication/32072148
TrustZone-generic-architecture-and-
RTZVisor-system-architecture-a-Arm-
TrustZone.png

Problem

- Trusted Computing
  - Manufacturers
  - Software providers
  - Users
- Open platform
  - Openness
  - PinePhone
  - OP-TEE

https://cdn.arstechnica.net/wp-content/uploads/2022/01/1-3-980x551.jpg

Secure boot, trusted boot and remote attestation for arm trustzone-based iot nodes

- Secure boot
  - Root of Trust
- Trusted boot
- Remote attestation
  - Remote verifier
  - Prove state
- Limitations
  - Openness
  - Code integrity
  - Implementation

Zhen Ling et al. “Secure boot, trusted boot and remote attestation for ARM TrustZone-based IoT Nodes”. In: Journal of Systems Architecture 119 (July 2021), p. 102240. DOI: 10.1016/j.sysarc.2021.102240

Goals

- Root of Trust?
  - User control
- Preserve openness
  - User-controlled attestation
- Stakeholder expectations
  - Users, manufacturers & software providers
- Comparable to related solutions
  - 'Closed' systems

Attacker Model

- Goals
  - Personal data
  - Computing resources
- Physical access
  - Evil maid
- OS attacks
- Software attacks
  - Malware
- Software integrity

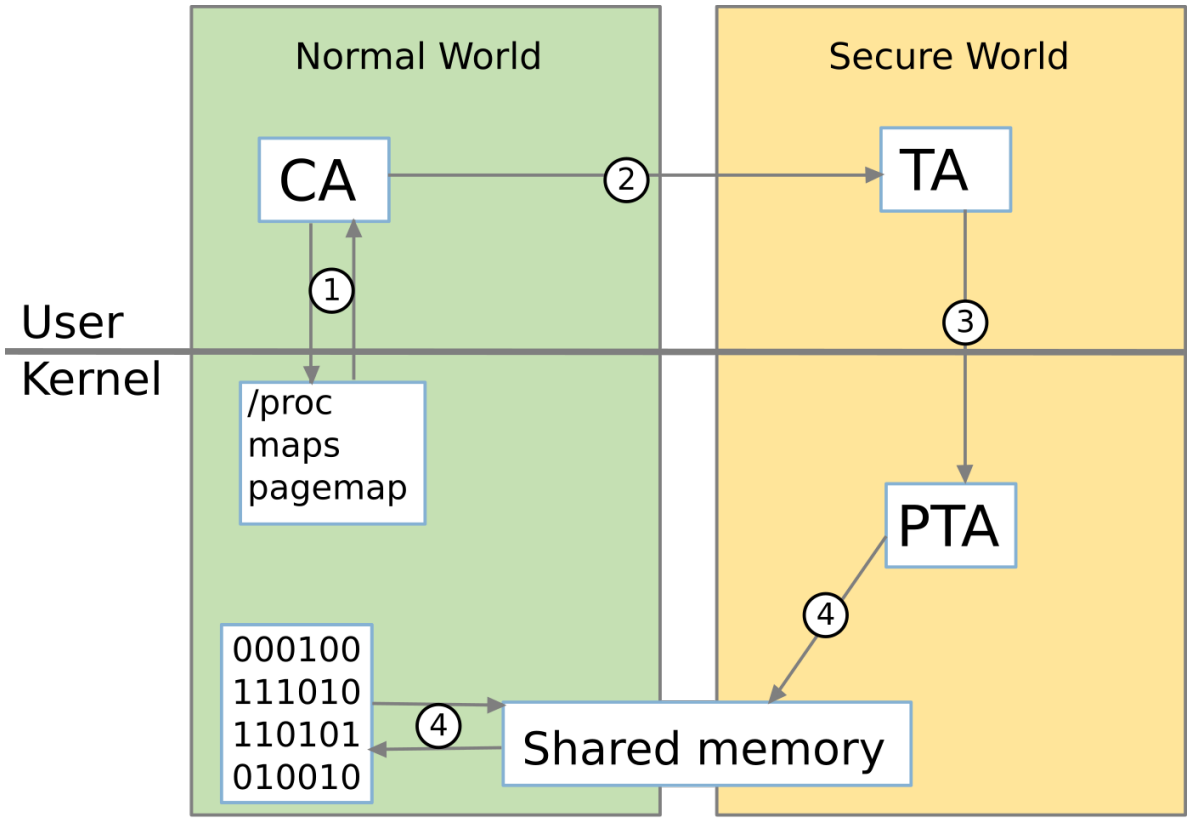
https://techxmedia.com/wp-content/uploads/2021/09/Smartphone-smartphone-security-hackers-TECHXMEDIA-1200x900.jpg

Outline

What is achieved

1. Identify processes
2. Address translation
3. Provide information
4. Shared memory
   - Calculate hash
   - Verify integrity

When to execute?

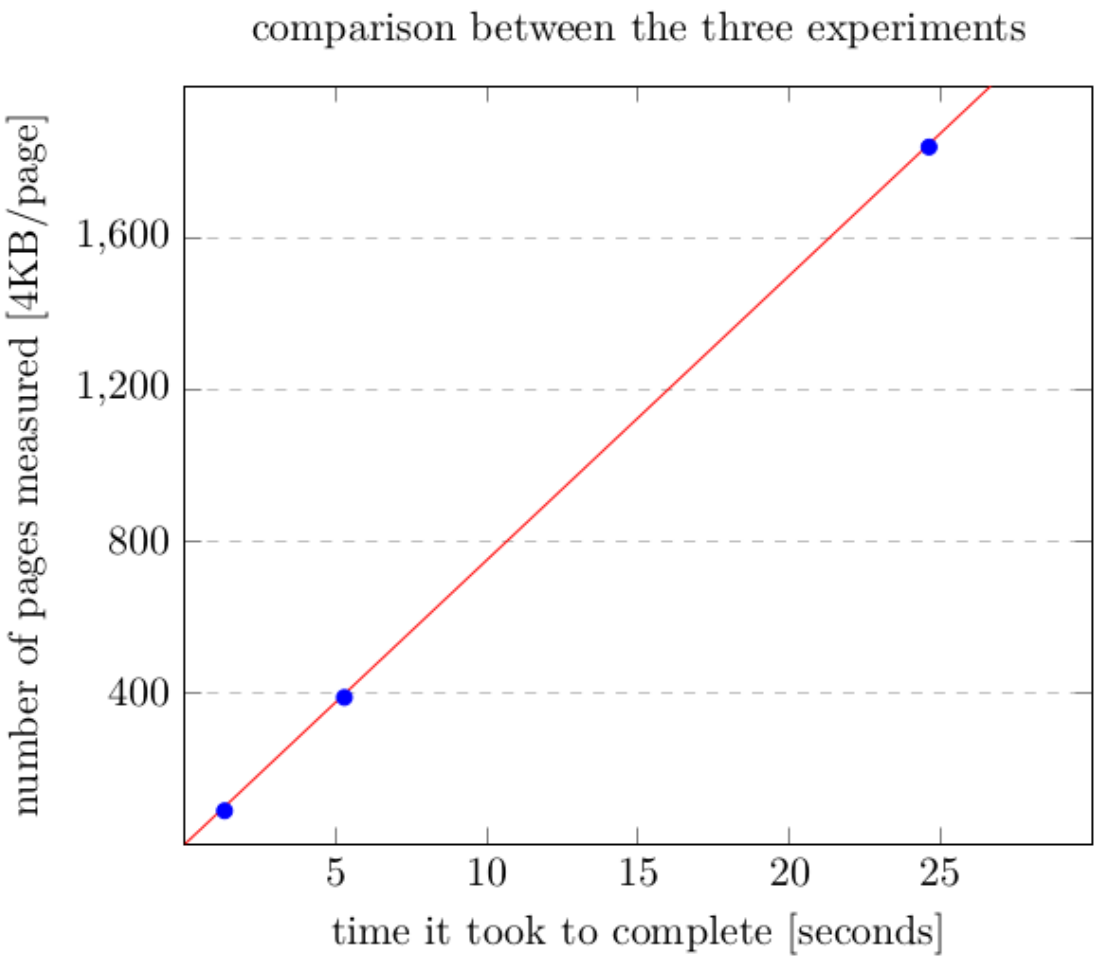


Outline

Performance Comparison

- Experiments (1850, 390 and 88 pages)
- Extrapolation (107 MB $\approx$ 360 seconds)
- Performance gap
  - Non-contiguous memory
  - QEMU emulator

Better performance
$\rightarrow$ Better security



Security Evaluation

- Guarantees
  - Code integrity Normal World
- Assumptions
  - Trusted Execution Environment
  - Secure storage
  - Execution control flow
- Added value
  - User control
  - Normal World security

https://lh3.googleusercontent.com/qBAtU92beaLB

Outline

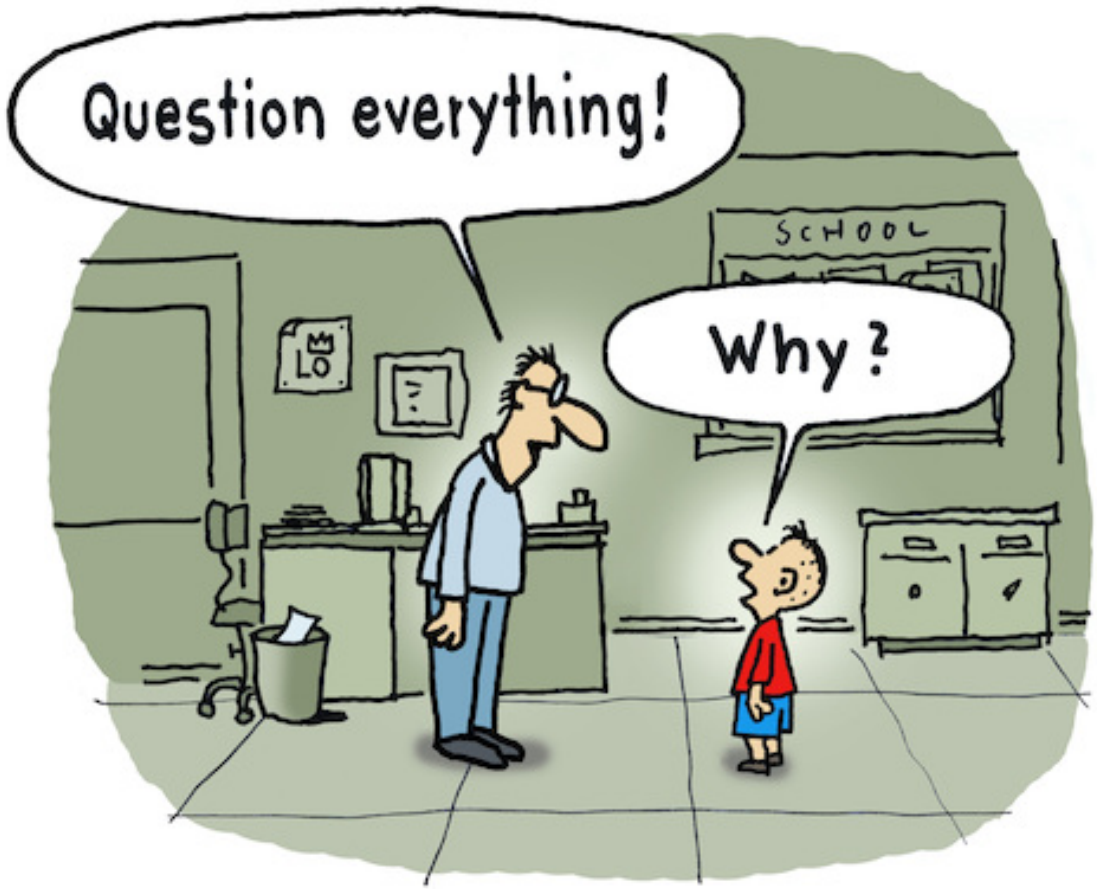
Findings

- User owned Root of Trust
  - User friendly?
  - Private key storage?
- Secure World checking integrity Normal World
  - Worth the overhead?
- Proof of Concept
  - Feasibility
  - Performance
- Similar security guarantees closed system
  - Assumptions and guarantees
  - User security
  - Manufacturer & software provider security?

Future work

- Rich OS dependency
  - Identifying processes
  - Insecure (compromised)
- Detailed attestation
  - Data structures
  - System calls
- Loaded memory pages
  - Sufficient comparison
  - Initialization phase
- Trusted I/O
  - Inform user
  - Allow to act

Questions?



https://www.toonpool.com/cartoons/Question_3768