Friedrich-Alexander-Universität
Erlangen-Nürnberg

Secure Services for Standard RISC-V Architectures

Davide Bove

August 25, 2022

IT Security Infrastructures Labs
Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

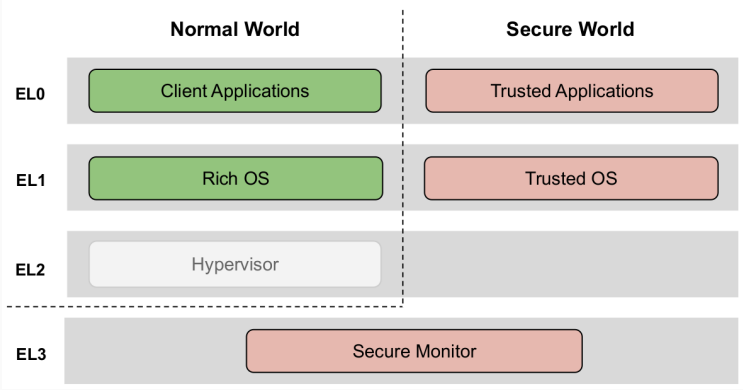
Table of contents

Contributions

- Analysis of current RISC-V capabilities for TEE features
- Proof of concept implementation of
  - secure file storage
  - cryptographic key storage
- Evaluation of secure I/O on standard RISC-V devices

Background

Trusted Execution Environments



ARM TrustZone Architecture. Source: [1]

RISC-V Instruction Set Architecture (ISA)

- **R**educed **I**nstruction **S**et **C**omputer (RISC) principles
  - RISC: ARM, PowerPC
  - CISC: x86, AMD64
- Open-Source ISA
  - General operation of a CPU
  - Defines instructions, states, memory access etc.
  - (Optional) extensions
  - “Rulebook” for hardware vendor and programmer

Privilege modes

- **U**ser mode
- **S**upervisor mode
- **M**achine mode
- (**D**ebug mode)

Levels	Supported Modes	Intended Usage
1	M	Simple embedded systems
2	M,U	Secure embedded systems
3	M, S, U	Unix-like OSes

Source: RISC-V ISA Volume 2, Privileged Spec v. 20190608

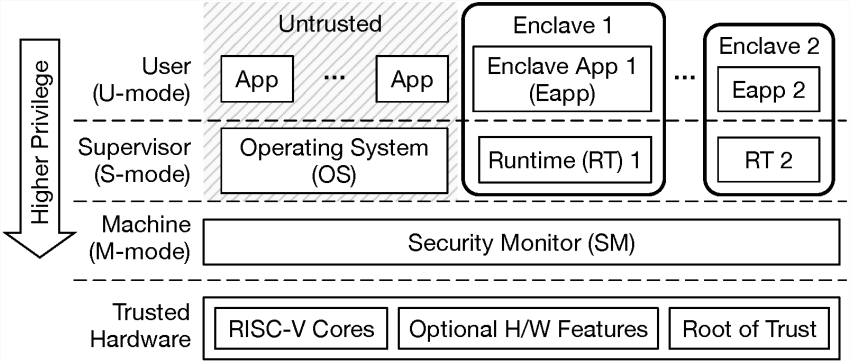
Physical Memory Protection

PMP

- M-mode controls memory access from U-mode and S-mode.
- Restrict access, set read / write/ execute flags for defined memory regions.

$\Rightarrow$ memory isolation of enclaves

Keystone Enclave



Keystone Enclave architecture [2]

Secure Storage

Requirements

1. Backed by non-secure resources
2. Bound to a particular device
3. Should be able to hide sensitive key material from the enclave itself.
4. Each enclave has access to its own storage space

Implementation

Problem

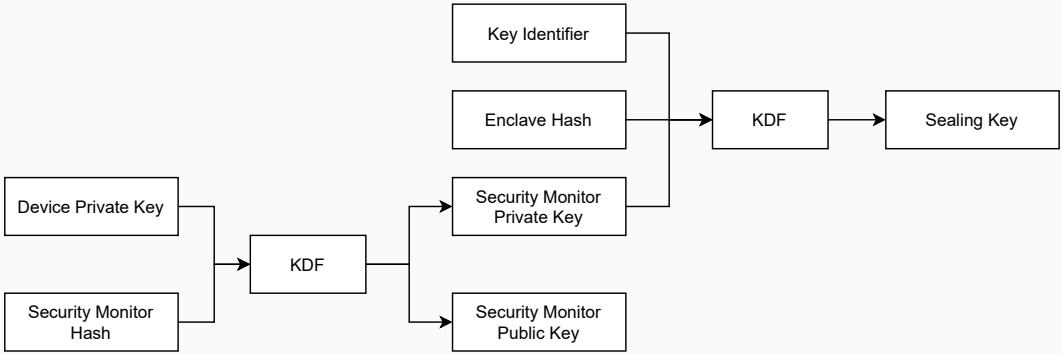
- No trusted storage
- Need for cryptography

$\Rightarrow$ derive a cryptographic key dynamically

Sealing key

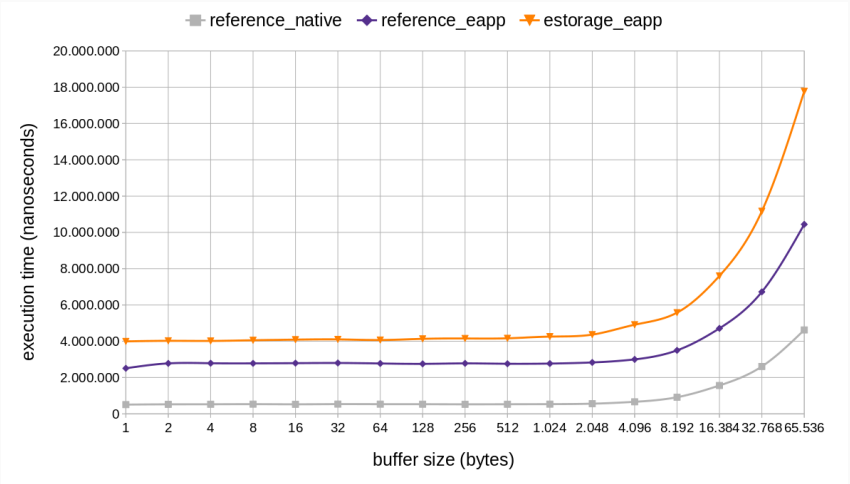
- Device-specific
- Bound to Secure Monitor hash
- Bound to enclave binary hash

Key Derivation



Key derivation for Sealing Key. Image: Jonathan Schmidt

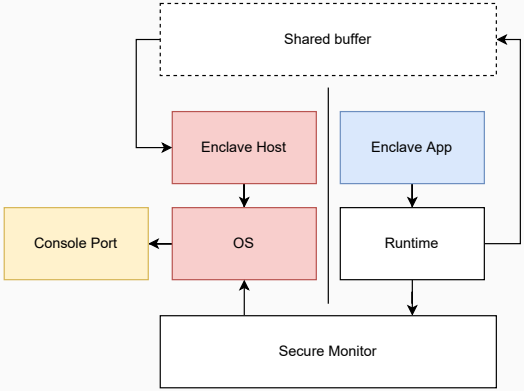
Performance of Secure Storage



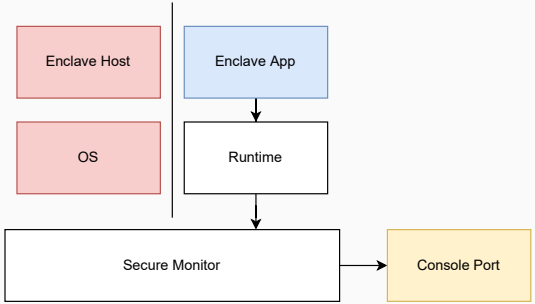
Performance measurements of native reference file (read/write) accesses compared to Secure Storage

Secure I/O

Secure Output Idea

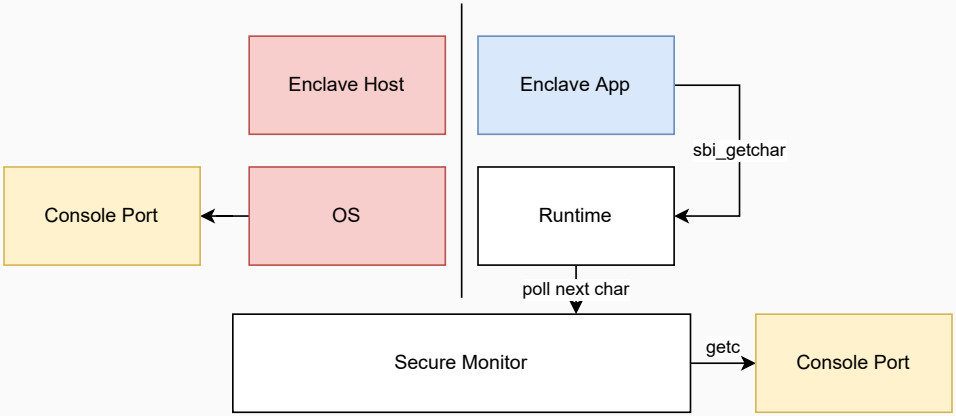


Keystone untrusted I/O calls



Our Secure I/O solution

Secure Input



Secure input through the Secure Monitor

Conclusion

Summary

Use standard RISC-V features to achieve basic security services

- secure file storage
- secure (cryptographic) keystore

TEE was not considered in original RISC-V design.

But: standard can be extended!

Thank you

Contact me: davide.bove@fau.de

Download this presentation: `https://d4vi.de/riscv-ss-presentation`

References i


M. Busch.
On the Security of ARM TrustZone-Based Trusted Execution Environments.
Doctoral thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2021.


D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song.
Keystone: An open framework for architecting trusted execution environments.
In *Proceedings of the Fifteenth European Conference on Computer Systems*, EuroSys '20, New York, NY, USA, 2020. Association for Computing Machinery.