An Open-Source Trusted Execution Environment for Resource-Constrained RISC-V MCUs

Luís Cunha Daniel Oliveira João Sousa Tiago Gomes Sandro Pinto

Centro ALGORITMI/LASI - Universidade do Minho

Abstract

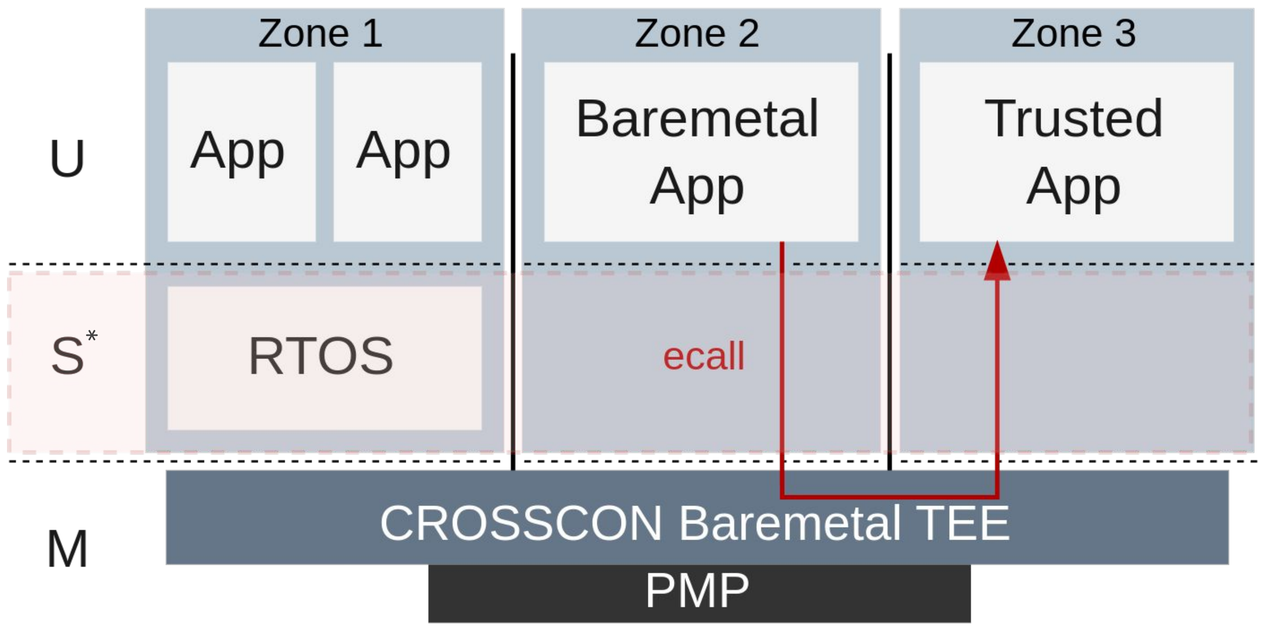
This work presents the design and implementation of CROSSCON Baremetal TEE, an open-source Trusted Execution Environment (TEE) targeting resource-constrained RISC-V-based MCUs with support for Machine, (Supervisor,) and User modes. The CROSSCON Baremetal TEE leverages RISC-V's privilege levels and memory isolation primitives to enable multi-world execution while maintaining strong security guarantees at the least privileged level. So far, we have implemented and validated the system on Machine and User mode-enabled cores, running a set of low-level benchmarks and test applications. Future plans include extending the support to the Supervisor mode and open-sourcing all artifacts to foster collaboration within the RISC-V community.

CROSSCON Baremetal TEE Overview

- RISC-V lacks standardized TEE architectures/specifications.
- Proprietary TEEs hinder interoperability and security.
- No clear security guarantees due to the absence of standards.

Proposal:

- ***CROSSCON Baremetal TEE***, a novel open-source TEE for resource-constrained RISC-V-based MCUs.
- Enables a multi-world architecture based on the zero-trust model, where only the TEE kernel is trusted.
- Leverages RISC-V ISA privilege levels and PMP for isolation, ensuring a reduced attack surface.
- GlobalPlatform compliant.



**Only M+U implementations currently supported*

Status & Evaluation

- We have implemented the proposed solution to run on Machine- and User-mode enabled cores.
  - The current implementation was evaluated on a Digilent ARTY7 35T FPGA, running the E300 SoC.

Microbenchmarks

- Analyzed the performance overhead associated with interactions between the executing Worlds.
  - **World switch time.** Clock cycles required to transition from the last instruction of the calling world to the first instruction of the target world.
  - **TA-based API.** Clock cycles between the CA invoking the API and the execution of the first instruction of the TA's corresponding handler function.
- **Primary bottleneck.** Overhead associated with saving and restoring the full execution context.

World Switch Time		TA-Based APIs		
u.d. → t.d.	t.d. → u.d.	Open Session	Invoke Command	Close Session
446 cycles	479 cycles	6880 cycles	6782 cycles	1082 cycles

- Initial performance evaluation considers both low-level microbenchmarks and real-world applications.
- As part of the open-source focused CROSSCON project, we plan to release all implementation artifacts

Real-world Applications

- Compare the execution time of the applications running in a bare-metal setup against a setup where security services are isolated within a TA.
- The **keylogger** is based on a reference implementation of a trusted keypad integrated in the Vulcan system.
  - The solution introduced an overhead of *2.73x*.
- The **Bitcoin Wallet** is based on an open-source implementation that supports six commands, ranging from master key generation to transaction signing.
  - The solution introduced an overhead of *1.003x*.

Keylogger		Bitcoin Wallet	
Single World	Two Worlds	Single World	Two Worlds
137K cycles	374K cycles **(2.73x)**	373M cycles	374M cycles **(1.003x)**