Extending Trusted Execution Environments in Architectural Simulators

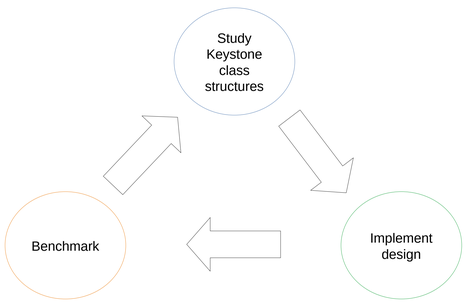
Will Buziak

Abstract

- Trusted Execution Environments (TEEs) provide hardware guarantees that seek to protect the security and isolation of application data.
- Many proprietary TEEs exist, each with its own implementation and respective way of providing security.
- Open-source TEEs like Keystone grant users the ability to contribute to their standard for security
- In order to extend TEEs, developers need to either implement their designs on FPGAs, or turn to architectural simulators.
- Lack of effective tools for development and thorough testing on real-world benchmarks make pre-fabrication development difficult.
-
- This work outlines the methods for implementing and evaluating contributions to Keystone within the gem5 architecture simulator.

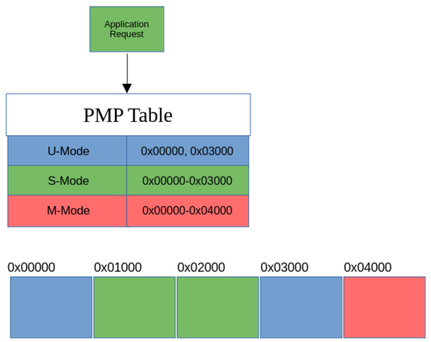
Methodology

- Keystone is an open-source, RISC-V TEE designed for custom configuration and modulation
- gem5 is an open-source architectural simulator that provides full-system emulation, including Keystone components

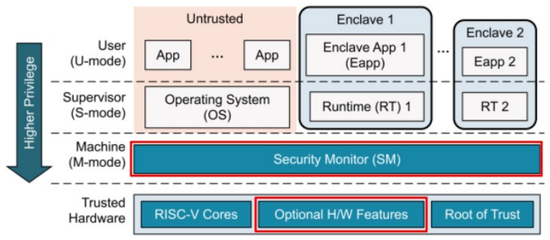


Keystone

- Enclaves like Keystone provide hardware guarantees that data is safe from a malicious:
  - Application thread
  - Operating System
  - Remote User
- Security is offered by isolating memory regions, rules are initiated by the bootloader (Security Monitor) and enforced during execution by a component in each core known as the Physical Memory Protection (PMP) Tables

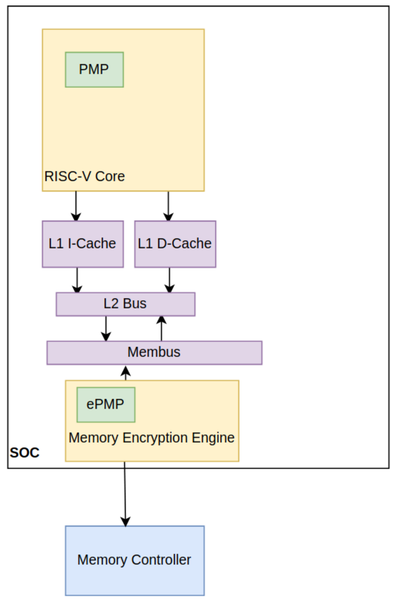


- Keystone grants different processes privileges by distinguishing its *mode:*
  - Machine (M) mode has the highest privileges and largely consists of the Security Monitor and the PMP table itself
  - Supervisor (S) & User (U) modes consist of OS-related and user threads



gem5

- gem5 presents architectural design as typical class structures with attributes based on the behavior of the real world component
- Developers wishing to extend Keystone components can create their own version of the desired component's class structure, often requiring close attention to:
  - Port connections
  - Packet handling



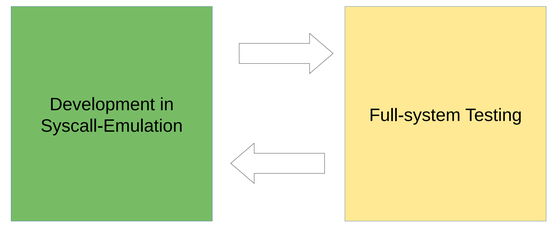
- For example, in order to properly connect the PMP to the proposed ePMP, it is either necessary to write a new:
  - Packet type, bypassing the cache hierarchy

Or

  - Port type directly connecting the core(s) to the Memory Encryption Engine

Evaluation

- gem5 supports syscall-emulation mode that uses the OS and system calls from the host machine, assisting quick development
- Thorough benchmarking can be achieved with full-system emulation in which an entire OS and emulated file system can be loaded into the simulation.
- Designs can be tested on real workloads and state-of-the-art or custom benchmarks designed to stress test the specific design components of interest.
- The ability to rapidly test and redesign is a major benefit of using a simulation environment over FPGA development



References

- D. Lee, Building Trusted Execution Environments, 2022.
- J. Lowe-Power, A. Mutaal Ahmad, A. Alian, R. Amslinger, and et. al., The gem5 Simulator: 20.0+, 2020.
- A. Akram, V. Akella, S. Peisert, and J. Lowe-Power, Enabling Design Space Exploration for RISC-V Secure Compute, In: *Fifth Workshop on Computer Architecture Research with RISC-V (CARRV 2021)*, 2021.
- Z. Moolman, T.S. Lehman, Extending RISC-V Keystone to Include Efficient Secure Memory, in: *Eighth Workshop on Computer Architecture Research with RISC-V (CARRV 2024),* 2024.

MINES Computer Science