More Computation on More Sensitive Data
Trusted Execution Environments (TEEs)

Isolation via software (e.g., virtual memory)

How it’s done traditionally

= What needs to be trusted

30 million lines of code

very wide interface

has no idea what the OS is doing

Microcode / Firmware

OS

App

App

App
Trusted Execution Environments (TEEs)

Isolation via software (e.g., virtual memory)

How it’s done traditionally

Isolation via hardware

A promising alternative (TEE)

Microcode / Firmware

Root of Trust

narrow interface

trusted functions

a few thousand lines of code
Trusted Execution Environments (TEEs)

- How it’s done traditionally
  - Isolation via software (e.g., virtual memory)
  - Microcode / Firmware

- A promising alternative
  - Isolation via hardware (TEE)
  - Microcode / Firmware

Reducing Trusted Computing Base (TCB)

- Root of Trust
  - Trusted functions
  - A few thousand lines of code

= What needs to be trusted

How it’s done traditionally
Challenges in Existing TEEs

- Security
- Performance
- Functionality

- Intel SGX
- ARM TrustZone
- AMD SEV
Challenges in Existing TEEs

Fixed Design Decisions

Security

Performance

Functionality

Intel SGX

ARM TrustZone

AMD SEV
Challenges in Existing TEEs

Closed-Source Hardware

Security

Fixed Design Decisions

Performance

Functionality

Intel SGX

ARM TrustZone

AMD SEV
“Agile Development of TEEs”
Keystone Enclave Project

- A framework for trusted execution environments
- Open source
- Started in 2018
- RISC-V
- Academically started in the ADEPT Lab @ UC Berkeley
Goals of the Project

- **Enable TEE on (almost) all RISC-V processors**
  - Follow RISC-V standard ISA
  - Standard TEE specification for various RISC-V sub-ISA

- **Make TEE easy to customize** depending on needs
  - Base implementation vs. platform-specific implementation
  - Reuse the implementation across multiple platforms

- **Reduce the cost** of building TEE
  - Reduce hardware integration cost
  - Reduce verification cost
  - Integrate with existing software tools
Keystone Architecture and Trust Model

User (U-mode)

Supervisor (S-mode)

Machine (M-mode)

Optional HW

Root of Trust

Trusted Hardware

Higher Privilege
Keystone Architecture and Trust Model

- User (U-mode)
  - App
  - App
  - Enclave App
  - Enclave App

- Supervisor (S-mode)
  - OS
  - Runtime
  - Runtime

- Machine (M-mode)
  - Security Monitor (SM)

- Trusted Hardware
  - Optional HW

Higher Privilege

Root of Trust

C0 C1 C2 C3

Trusted
Keystone Architecture and Trust Model

User (U-mode)

Supervisor (S-mode)

Machine (M-mode)

Security Monitor (SM)

Hardware-Enforced and Software-Defined Isolation
RISC-V TEE on Multiple Platforms
Agile Customization of TEE

- Various Threat Models
  - Software-based memory encryption via on-chip scratch pad [Andrade, CARRV’20]
  - Cache partitioning via way masking [Lee, EuroSys’20]
  - Oblivious paging [Roy, WOOT’20]

- Performance Optimization
  - Dynamic memory resizing [Lee, EuroSys’20]
  - Crypto accelerator [Hoang, IEEE Access’20]
  - Memory sharing [Yu, USENIX Security’22]

- TEE on Embedded Systems
  - TEE on FreeRTOS [Thomas, CARRV’21]
Power of Open Source

- Security by Transparency
  - Academic security analysis [Van Bulck, CCS’19]
  - Push-button formal verification [Nelson, SOSP’19]
  - Academic insights [Cerdeira, S&P’20] [Dessouky, USENIX Security’19]

- Reuse SW/HW Implementations
  - Reduce implementation/verification cost
Formal Verification of Enclave Platform

- A good layer of abstraction for formal modeling

- CPU state: pc, regs, mem
- Addr translation state: addr_map, cache, os_metadata
- Enclave state: current_eid, owner, enc_metadata
Conclusion

- Keystone is an open-source framework for RISC-V TEEs
- Keystone enables agile development of TEEs
  - Open implementation
  - Customizable architecture
  - Portable design with RISC-V standard
- Future work
  - Scalable memory isolation via RISC-V H-extension
  - Secure and efficient enclave cloning
  - Standardize measured boot and attestation
  - Dynamic library support with trusted loader
Thank You

dayeol <at> berkeley <dot> edu