CURE: A Security Architecture with Customizable and Resilient Enclaves

Raad Bahmani, Ferdinand Brasser, Ghada Dessouky, Patrick Jauernig, Matthias Klimmek, Ahmad-Reza Sadeghi, Emmanuel Stapf

Technical University of Darmstadt
Enclave Security Architectures

- Enclaves prominent approach for protecting sensitive services
- OS assumed to be potentially compromised
- Isolated execution environment, backed by hardware-assisted security mechanisms
Enclave Security Architectures

• Enclaves prominent approach for protecting sensitive services

• OS assumed to be potentially compromised

• Isolated execution environment, backed by hardware-assisted security mechanisms

• Trusted SW configures security mechanisms

• Trusted SW assigns system resources to enclaves (memory, cores, caches)
Challenges of Enclave Computing

- **Security:**
  - Side-channels attacks not considered in industry solutions
  - Cache side channels and controlled side channels (page table, interrupt handlers)

- **Functionality:**
  - Missing functionality regarding secure I/O, secure Direct Memory Access (DMA)
  - Secure binding of enclaves to peripherals

- **Configurability:**
  - Enclaves cannot be adapted to security and functionality requirements
  - Existing proposals follow *one-size-fits-all* approach
Enclave Types in Existing Enclave Security Architectures
User-space Enclaves

- Processor
  - User Level
    - App
    - App
  - Privileged Level
    - Operating System (OS)
      - L1 Cache
      - MMU
      - TLB
  - System Bus
  - L2 Cache
  - DRAM
  - OS
User-space Enclaves

- User Level
- Privileged Level
- Highest Privilege Level

Processor

- App
- App
- Enclave_A
- Enclave_B

Operating System (OS)

Trusted Software (TS)

L1 Cache

MMU

TLB

System Bus

L2 Cache

DRAM

- TS
- OS
- A
- B

Software TCB
User-space Enclaves

Pros:
- Reuse of OS functionalities
- Low system resource consumption
- Easy to develop

Cons:
- No privileged code in enclave (I/O)
- Increased performance overhead for context switches
- Protection from controlled side-channel attacks challenging

- Provided by Sanctum [1], SGX [2] and various extensions [3-8]
Kernel-space Enclave

**Pros:**
- Privilege code in enclave
- No overhead on context switches
- Easier to prevent controlled side-channel attacks

**Cons:**
- Increased resource consumption
- Increased overhead for enclave setup
- Need to develop runtime

- Provided by TrustZone [9], Sanctuary [10] or Keystone [11]
CURE: A Security Architecture with Customizable and Resilient Enclaves
Goals of the CURE Security Architecture

• Tackle the aforementioned challenges

• **Security:**
  • Protect against controlled side-channel attacks (page table, interrupt handlers)
  • Protect against cache side-channels attacks

• **Functionality:**
  ▪ Provide a secure binding from peripherals to enclaves

• **Configurability:**
  • Provides different types of enclave, selected depending on sensitive service & usage scenario
CURE: Customizable and Resilient Enclaves

RISC-V Processor

User Level:
- App

Supervisor Level:
- Operating System (OS)

L1 Cache  MMU  TLB

System Bus

DRAM
  L2 Cache  OS
CURE: Customizable and Resilient Enclaves

- Multiple types of enclaves
- SM responsible for all security-critical tasks and services (e.g., remote attestation)
CURE: Customizable and Resilient Enclaves

- Multiple types of enclaves
- SM responsible for all security-critical tasks and services (e.g., remote attestation)
- Way-based cache partitioning on shared L2 cache
- Novel access control mechanism on system bus, minimal changes at processor
- Allows for secure binding between enclaves and peripherals
CURE: Customizable and Resilient Enclaves

RISC-V Processor

- Enclave B
- Enclave C

Operating System (OS)

- App
- Enclave A

Firmware

- Security Monitor (SM)

System Bus

- L1 Cache
- MMU
- TLB

DRAM

- L2 Cache

Filter Engine

Peripherals

- Multiple types of enclaves
- SM responsible for all security-critical tasks and services (e.g., remote attestation)
- Way-based cache partitioning on shared L2 cache
- Novel access control mechanism on system bus, minimal changes at processor
- Allows for secure binding between enclaves and peripherals
Adding *enclave ID* to TileLink protocol (A & C channels) propagated through system.
Details on System Bus Access Control

- Adding *enclave ID* to TileLink protocol (A & C channels) propagated through system
- Added logic and registers at arbiters and decoders for access control on memory transactions
Details on System Bus Access Control

- Adding *enclave ID* to TileLink protocol (A & C channels) propagated through system
- Added logic and registers at arbiters and decoders for access control on memory transactions
- Arbitration logic unmodified
- System bus connected to peripheral bus and interrupt bus
Details on Software Components

- Enclave setup triggered by OS
- OS performs security uncritical steps (e.g., load enclave binary)
- SM performs all security critical steps
  - Binary verification
  - Interrupt configuration
  - Setting up shared memory for communication
  - Page table modification (user-space enclave)
Details on Software Components

- **User Level**
  - App
  - App
  - Enclave

- **Supervisor Level**
  - Operating System (OS)
    - Enclave
      - Enclave setup triggered by OS
      - OS performs security uncritical steps (e.g., load enclave binary)
      - SM performs all security critical steps
        - Binary verification
        - Interrupt configuration
        - Setting up shared memory for communication
        - Page table modification (user-space enclave)

- **Machine Level**
  - Firmware
    - Security Monitor (SM)
  - Linux LTS kernel 4.19
    - 375 LOC modified
    - + custom kernel module (200 LOC)
  - 544 LOC + Crypto-Library (LibTomCrypt, 2586 LoC)
Conclusion

• CURE successfully tackles identified challenges

• **Security:**
  • Keep all side-channel sensitive data structures inside enclave (page tables, interrupt handlers)
  • Dynamic way-based cache partitioning

• **Functionality:**
  • New access control mechanism on system bus enables enclave-to-peripheral binding

• **Configurability:**
  • Provides multiple types of enclaves

• CURE offers many possibilities for further development (e.g., VM enclaves, new side-channel resilient cache architectures)
Questions ?

emmanuel.stapf@trust.tu-darmstadt.de


