Trusted Execution Environment and Confidential Computing

Mengjia Yan
Spring 2022

Based on slides from Intel SGX Tutorial
Trusted Computing Base (TCB)

- Trusted
- Hardware
- SMM (firmware)
- Hypervisor
- Guest OS
- App
- Ring -2
- Ring -1
- Ring 0
- Ring 3
- Hardware
- OS
- App

6.888 L8 - Trusted Execution Environment
Why Shrink TCB?

• Software bugs
  • SMM-based rootkits
  • Xen 150K LOC, 40+ vulnerabilities per year
  • Monolithic kernel, e.g., Linux, 17M LOC, 100+ vulnerabilities per year

• Remote Computing
  • Remote computer and software stack owned by an untrusted party
  • Why outsource computation?
  • What security problems do we have?
Secure Remote Computing

• Example: DNA Analysis

How to keep my data private without trusting the host OS/hypervisor/SMM?
Solutions

• Homomorphic Encryption

• 4 to 5 orders of magnitude slower than computing on unencrypted data.

Remote Computer managed by untrusted infrastructure provider

Software Provider
Data Owner

Enc(x); Function f

F'( Enc(x) ) = Enc(f(x))

• Performance? Accelerators?

F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption; Axel Feldmann, Nikola Samardzic et al. MICRO’21

6.888 L8 - Trusted Execution Environment
Solutions

• Move TCB to Hardware ...

• Performance?
• Need to trust hardware. How to achieve it?

Remote Computer managed by untrusted infrastructure provider

Container runs trusted software
1. Decrypt to get x
2. Compute f(x)
3. Encrypt f(x)

Software Provider
Data Owner

Enc(x); Function f

Enc( f(x) )
Move TCB to Hardware ...

Arm TrustZone

Intel SGX

AMD SEV

6.888 L8 - Trusted Execution Environment
Privileged Software Attacks

• Manipulate everything

• Directly see and modify application code and data
  → Need to encrypt secret data
  → Need to verify integrity (software attestation)

• Mess up with
  • Address translation
  • Process initialization and context switch
  • Interrupts, I/Os
  • etc.

6.888 L8 - Trusted Execution Environment
Enclave High-level View

• Goal: A protected environment that contains the code and data of a security-sensitive computation.
SGX HW TCB

Memory Management Unit (MMU)

Processor Chip (socket)
- core
- L1/L2

LLC

System Bus (logically)

Integrated Memory Controller

Memory (DRAM)

Non-volatile storage device

other I/O Device

6.888 L8 - Trusted Execution Environment
Intel SGX Security Mechanisms

Attestation

Isolation

DRAM Protection

6.888 L8 - Trusted Execution Environment
Attestation

• Platform Attestation

• Enclave Measurement

https://www.conclave.net/blog/decrypting-enclaves-encryption-key-hierarchy/
Enclave Initialization

- BIOS setup PRM region
Enclave Initialization

• Enclave creation (ECREATE)
Enclave Initialization

- Add page (EADD)
- Measure (EEXTEND)
Enclave Measurement

- Hardware generates a cryptographic log of the build process
  - Code, data, stack, and heap contents
  - Location of each page within the enclave
  - Security attributes (e.g., page permissions) and enclave capabilities

- Enclave identity (MRENCLAVE) is a 256-bit digest of the log that represents the enclave
Enclave Initialization

- Add page (EADD)
- Measure (EEXTEND)
- Init (EINIT)
  - Finalize measurement
- Active (EENTER)
  - Switch to enclave mode

Virtual Address Space

Physical Address Space

Plaintext Code/Data

PRM

Code/Data

ELRANGE

Enclave 1 metadata

Update mapping information in EPCM

6.888 L8 - Trusted Execution Environment
Enclave Attestation and Sealing

• HW based attestation provides evidence that “this is the right application executing on an authentic platform” (approach similar to secure boot attestation)
Intel SGX Security Mechanisms

Attestation

Isolation

DRAM Protection

Remote Platform

Enclave

App

Guest OS

Hypervisor

SMM

Hardware

Processor Reserved Memory (PRM)

Integrated Memory Controller

Processor Chip (socket)

core L1/L2

core L1/L2

... LLC

Memory (DRAM)

6.888 L8 - Trusted Execution Environment
SGX Access Control

• Assume software attestation is done
• Can have multiple enclaves

Performance issues.
PRM size is 128MB in SGX V1.0
All enclaves loaded at the same time cannot exceed said ~90MB
Recap: Virtual and Physical Address

Virtual Address Space (Programmer's View)

Physical Address Space (limited by DRAM size)

Page Table per process
Recap: Virtual and Physical Address

Virtual Address Space (Programmer's View)

Physical Address Space (limited by DRAM size)

Page Table \textit{per process}

System software handles “page fault”

6.888 L8 - Trusted Execution Environment
Recap: Virtual and Physical Address

Virtual Address Space (Programmer's View)

Physical Address Space (limited by DRAM size)

Process 1

Process 2

Page Table per process

4KB

4KB

4KB

6.888 L8 - Trusted Execution Environment
Address Translation

- SW or HW or both?

1. Virtual Address
2. TLB Lookup
   - hit
   - miss
3. Page Table Walk
   - the page is in memory
   - the page is not in memory
4. Protection Check
   - permitted
   - denied
5. Update TLB
6. Page Fault (OS loads page)
7. SEGFAULT

Physical Address (to cache)

Where?

Update TLB

Protection Fault
Malicious Address Translation

Virtual Address Space (Programmer's View)

Physical Address Space (limited by DRAM size)

Enclave Linear Range (ELRANGE)

Page Table per process

Processor Reserved Memory (PRM)

4KB

4KB

4KB

4KB

4KB

6.888 L8 - Trusted Execution Environment
Malicious Address Translation

Virtual Address Space (Programmer's View)

Enclave Linear Range (ELRANGE)

4KB

Physical Address Space (limited by DRAM size)

4KB

Processor Reserved Memory (PRM)

4KB

4KB

Page Table per process

VA

PA

6.888 L8 - Trusted Execution Environment
Malicious Address Translation

Virtual Address Space (Programmer's View)

Physical Address Space (limited by DRAM size)

Enclave Linear Range (ELRANGE)

Page Table per process

Processor Reserved Memory (PRM)

4KB (belong to a different enclave)

4KB

4KB

4KB

4KB

How to deal with all these attacks?
Enclave Page Mapping Information

Virtual Address Space (Programmer's View)

Enclave Linear Range (ELRANGE)

Page Table per process

VA

PA

Enclave Page Cache Mapping (EPCM)

Reversed Page Table

{PA, VA, Enclave ID}

Physical Address Space (limited by DRAM size)

4KB

Processor Reserved Memory (PRM)

4KB

if (PA belongs to PRM) {
    compare VA in EPCM
    if (NOT match) {
        #Signal Fault
    }
}

6.888 L8 - Trusted Execution Environment
SGX Address Translation Attack Protection

- **AMD**
  - AMD-SEV uses a different mechanism
  - AMD-SEV-SNP adds a similar feature to have reversed page tables
Attacks on Intel SGX

• Page access side channels
  • Xu et al. “Controlled-channel attacks: Deterministic side channels for untrusted operating systems,” S&P’15

• L1FT/Foreshadow

• Single-Step/Zero-Step

Figure 3: The physical enclave secret is mapped to an inaccessible virtual address for transient dereference.
Intel SGX Security Mechanisms

Attestation

Isolation

DRAM Protection

Client Application

Enclave

Remote Platform

App

Enclave

Guest OS

Guest OS

Hypervisor

SMM

Hardware

Processor Reserved Memory (PRM)

Processor Chip (socket)

core L1/L2

core L1/L2

…

Integrated Memory Controller

LLC

Memory (DRAM)

6.888 L8 - Trusted Execution Environment
Protect Memory

- Processor Chip (socket)
  - core L1/L2
  - core L1/L2
  - LLC
  - ...
  - ... System Bus (logically)
- Integrated Memory Controller
- Memory (DRAM)
- Non-volatile storage device
- other I/O Devices

6.888 L8 - Trusted Execution Environment
Memory Encryption Engine (MEE)

- Confidentiality:
  - DATA written to the DRAM cannot be distinguished from random data.

- Integrity + freshness:
  - DATA read back from DRAM to LLC is the same DATA that was most recently written from LLC to DRAM.

*What attacks can be mitigated?*

*Rowhammer? Bus tapping? Side channels on address access?*
Confidentiality

• AES 128-CTR mode

Counter (CTR) mode encryption
Message Authentication Code (MAC)

- Hash(plaintext)

- Keyed Hash
  - $\text{MAC} = \text{Hash}(\text{ciphertext || key})$

- Freshness
  - $\text{MAC} = \text{Hash}(\text{ciphertext || key || nonce})$

If the same MAC is found: then the message is authentic and integrity checked. Else: something is not right.
Integrity Storage Problem

• For each cache line: \{ciphertext + CTR + MAC\}
  • MAC 56 bits
  • CTR 56 bits
• Can we store all the three components off-chip?
• Problem: if store CTR on-chip $\rightarrow$ high on-chip storage requirement
Operations on Merkle Tree

- Only need to store the root node on chip
- How to verify block B1?
- Write to block B3?

\[
\begin{align*}
\text{root} &= \text{Hash}(f_{2i} \ || \ f_{2i+1}) \\
\text{f}_i &= \text{Hash}(g_{2i} \ || \ g_{2i+1}) \\
\text{g}_i &= \text{Hash}(h_{2i} \ || \ h_{2i+1}) \\
\text{h}_i &= \text{Hash}(B_i)
\end{align*}
\]
Counter Integrity Tree
Intel SGX v.s. AMD SEV

SGX
- Application
  - Enclave
  - Enclave
- OS
- Intel CPU

SEV
- VM
- VM
- VMM
- AMD CPU
AMD SEV

- CPU
- AMD Secure Processor
  - Manages AES Keys
  - Handle SEV API
- Memory Controller
  - Memory Encryption Engine (MEE)
  - AES encryption/decryption
Arm TrustZone

from Hua et al. vTZ: Virtualizing ARM TrustZone. Usenix’17
• ARM Confidential Computing Architecture
Summary

• What is trusted execution environment/confidential computation?

• Main security mechanisms

• Multiple commercialized design
  • Intel SGX, AMD SEV, ARM CCA
  • Keystone, Sanctum, Penglai, etc