September Week1

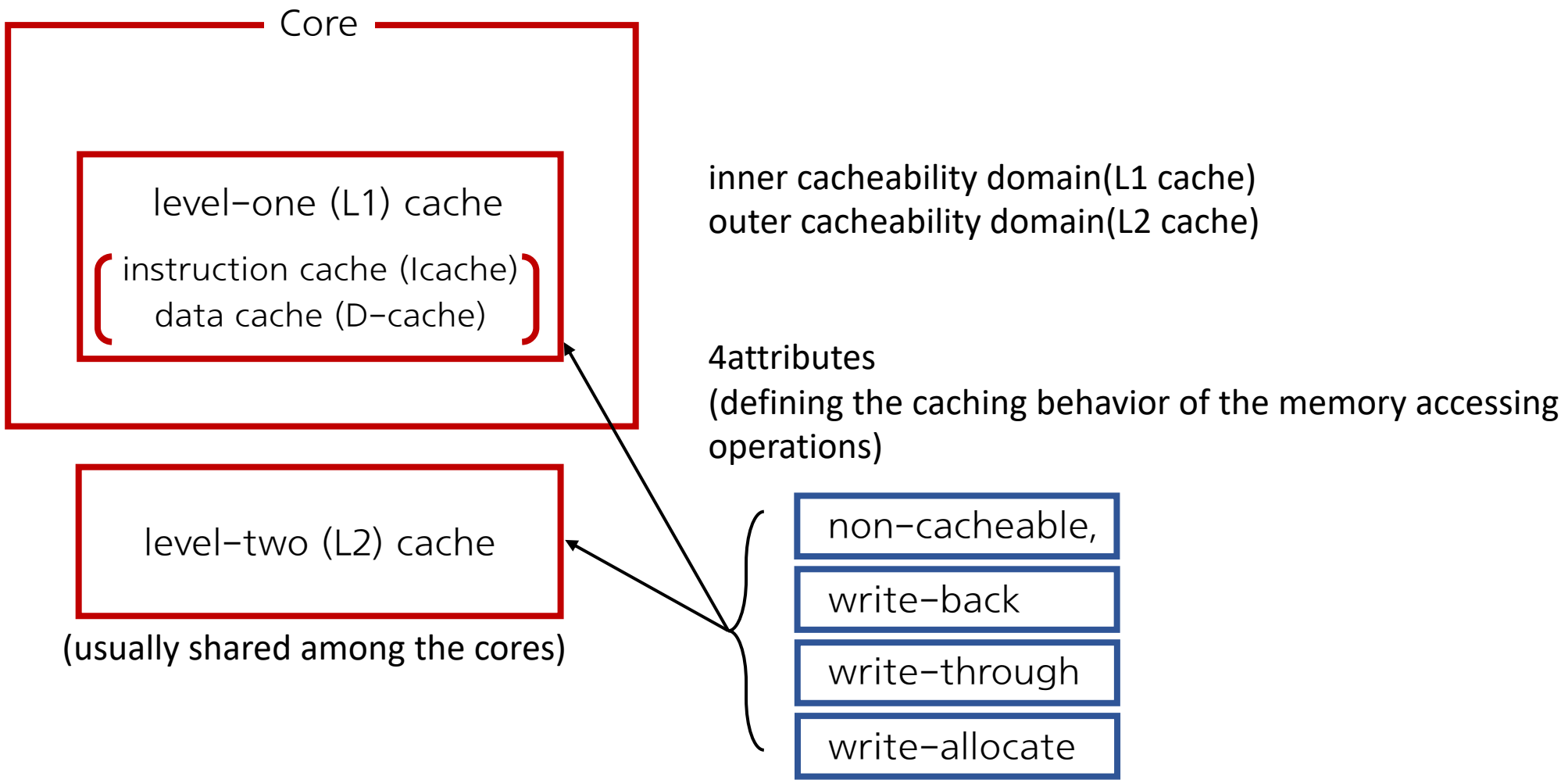
PROGRESS

2020019252 김나현

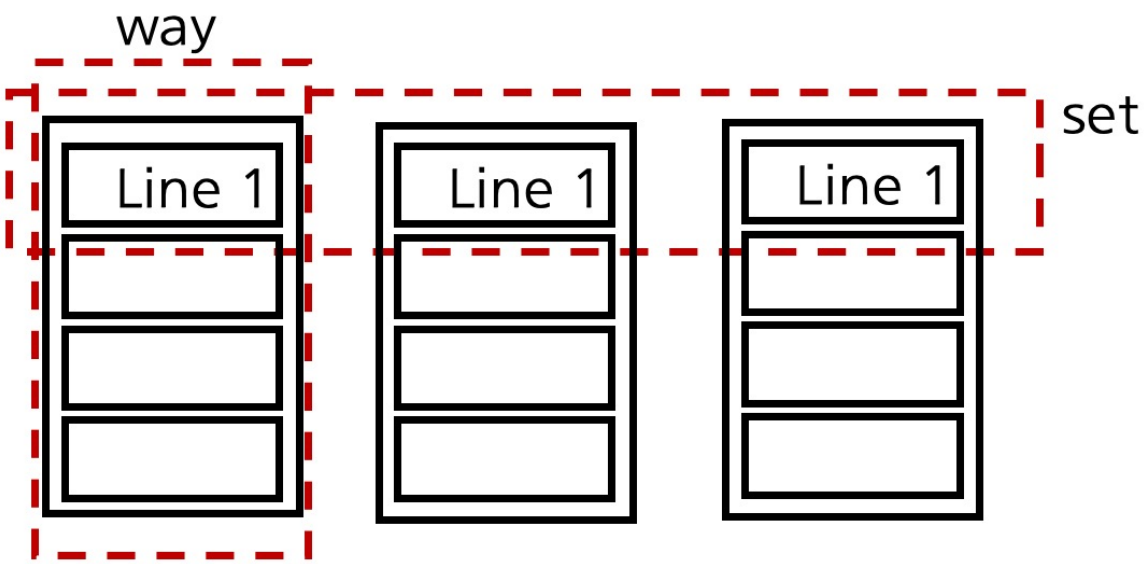
Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments

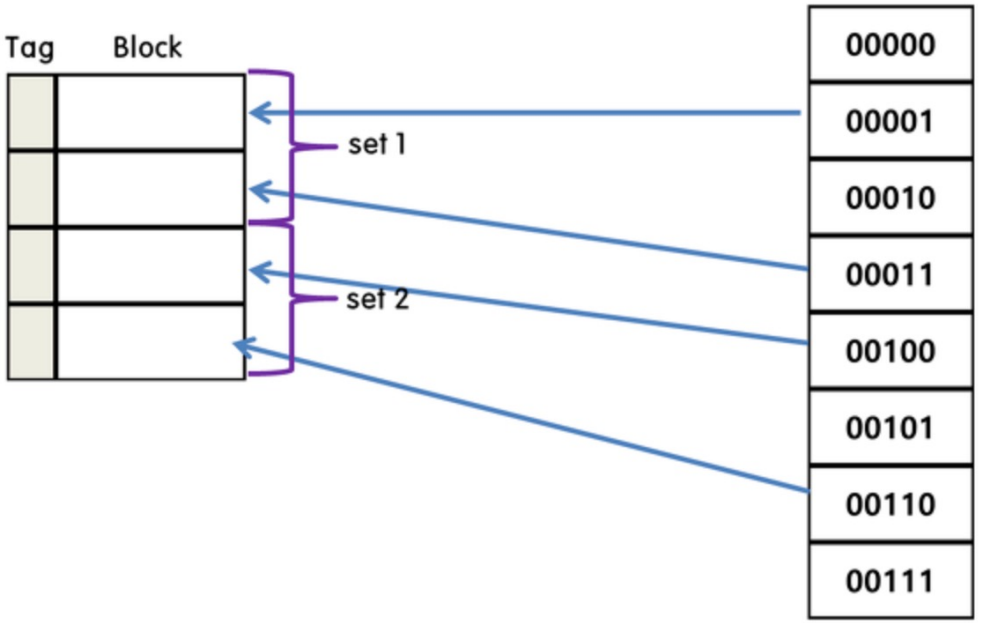
CCS '20: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications SecurityOctober 2020

ARM Cache Architecture



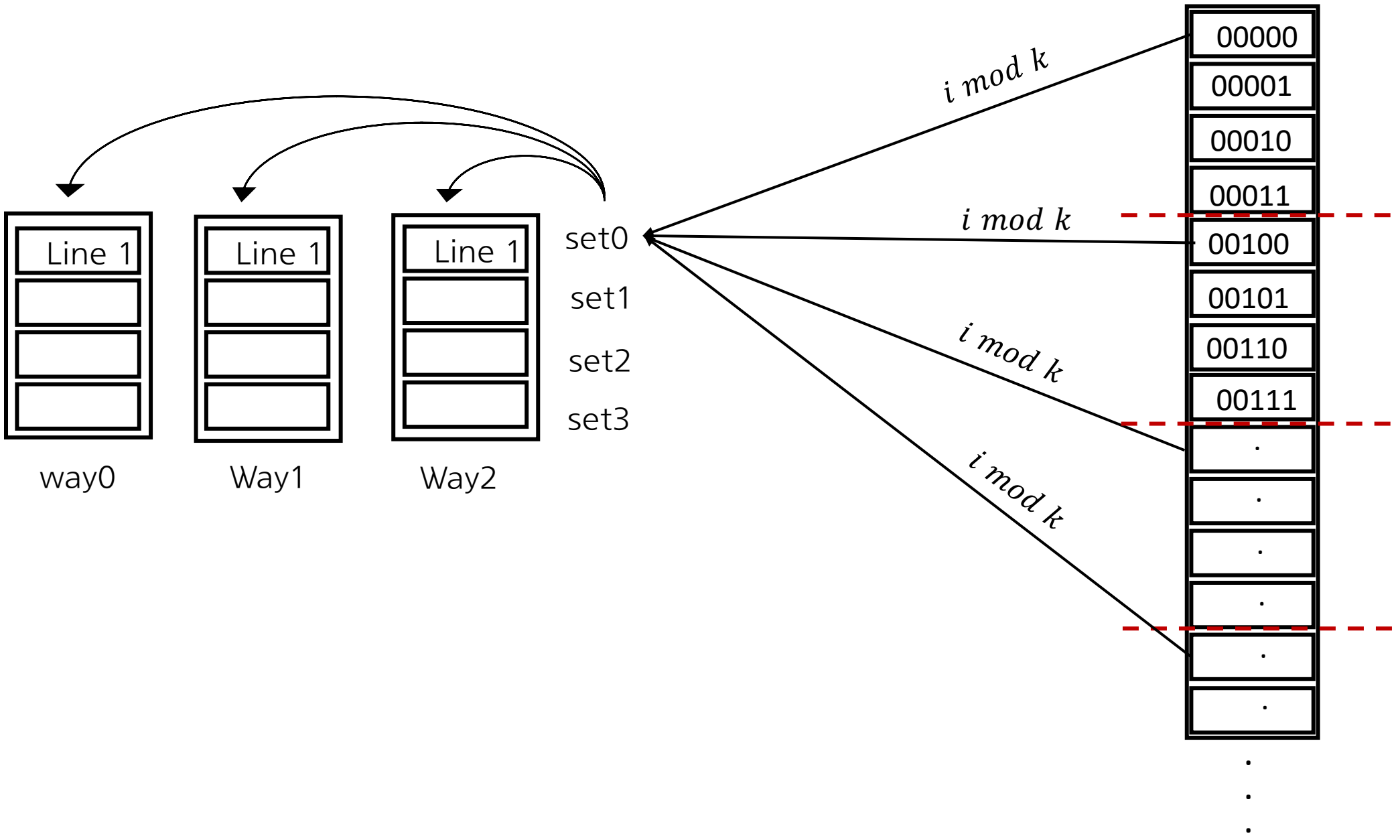
N-way Set Associative Cache





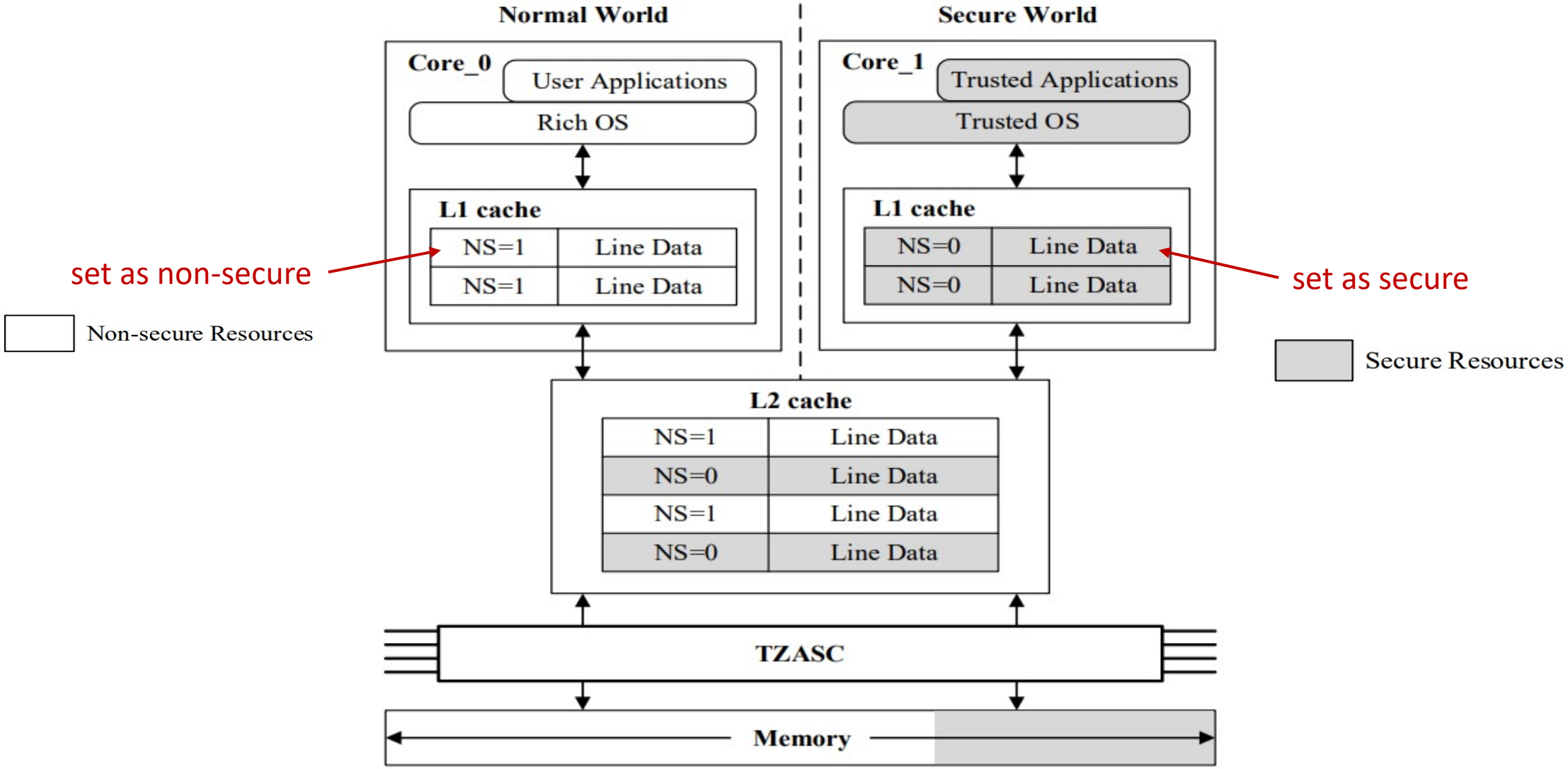
Set Associative mapping

N-way Set Associative Cache



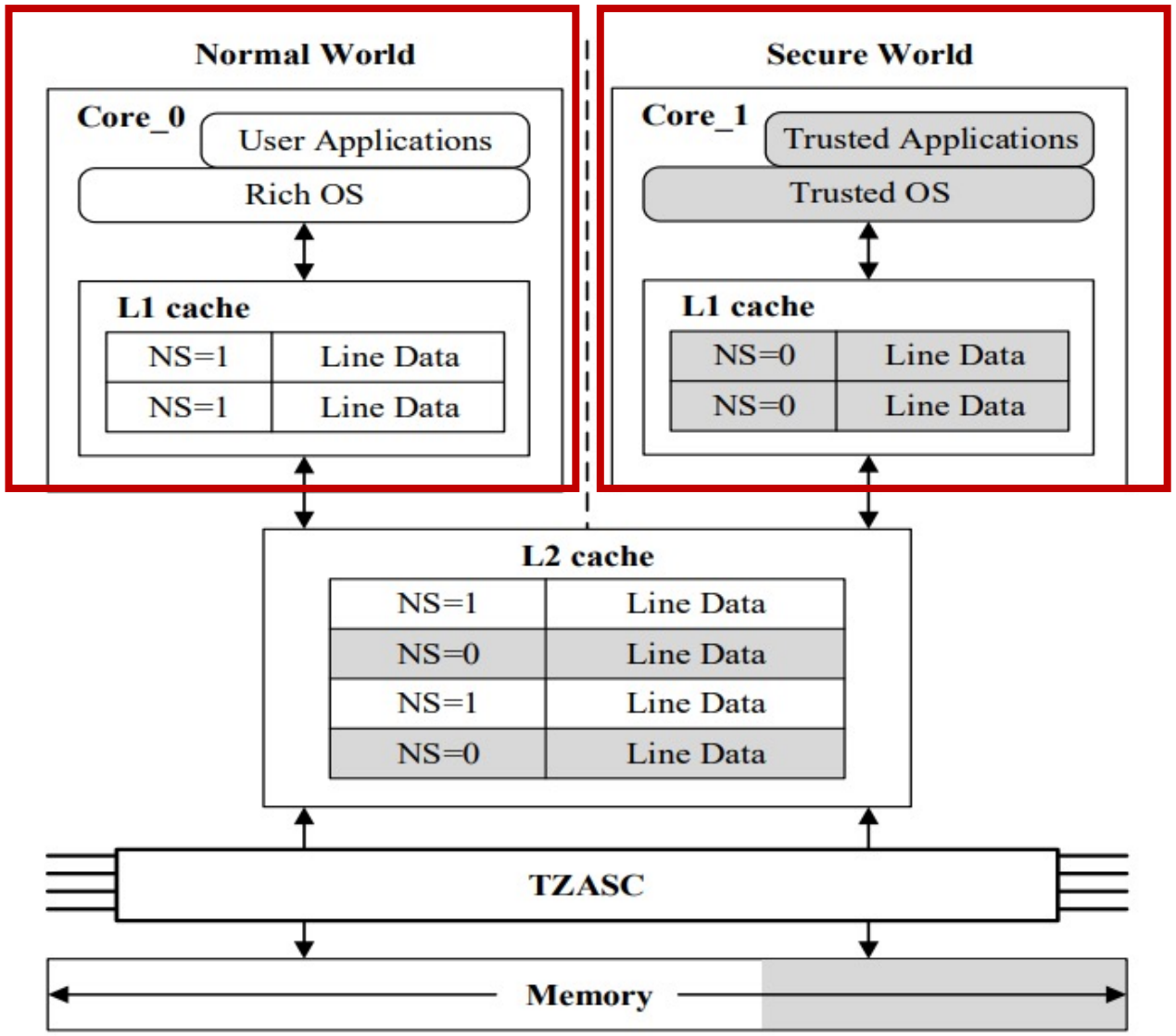
For ARM processors with TrustZone

Recording their security state



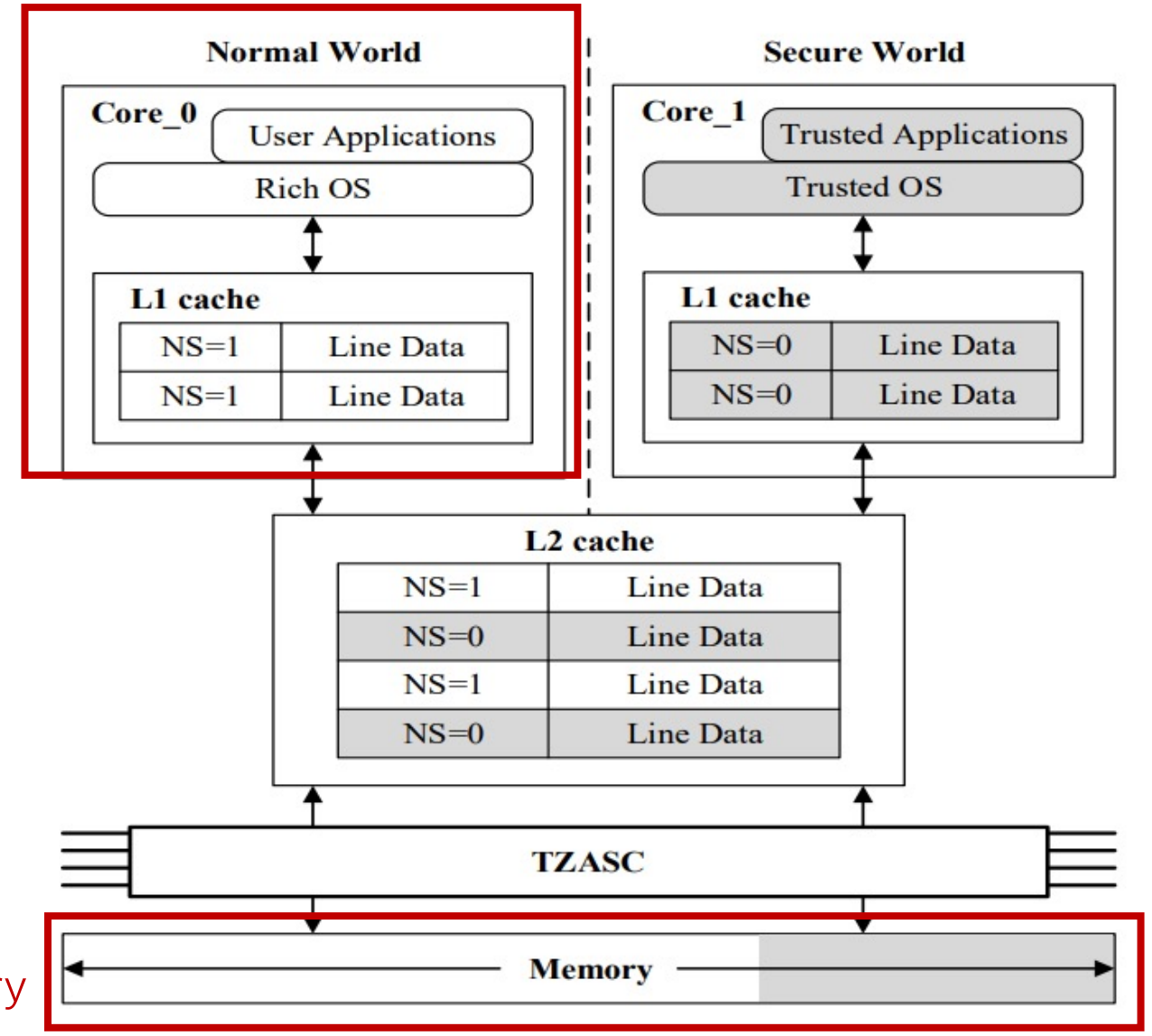
Arm trust zone

a hardware-based isolation environment for secure code execution



Arm trust zone

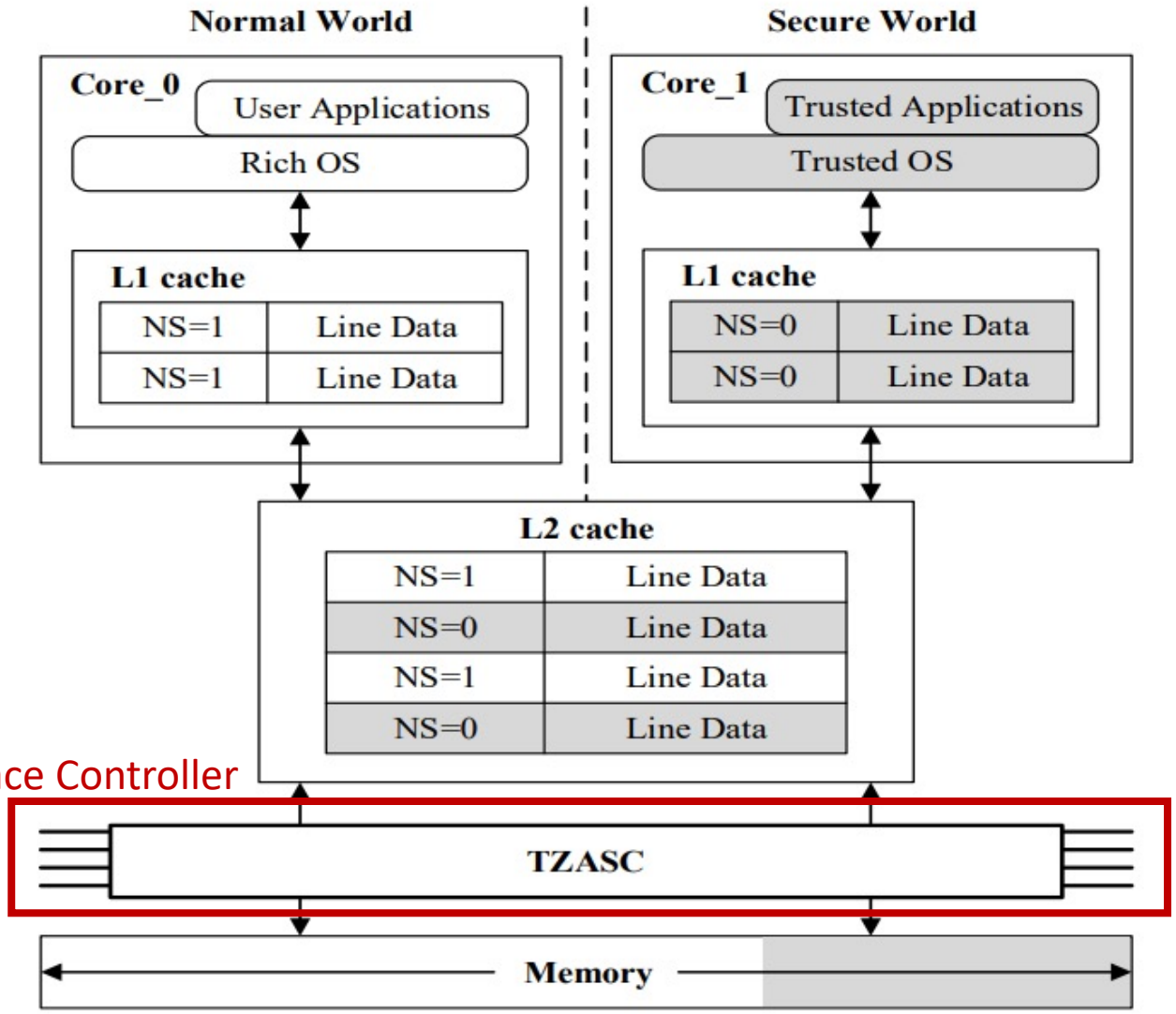
a hardware-based isolation environment
for secure code execution



separation of the physical DRAM

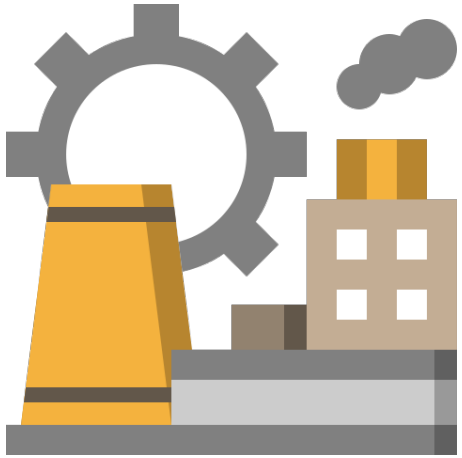
Into non-secure memory and secure memory

Arm trust zone



TrustZone Address Space Controller

conflicting requirements



from device manufacturers

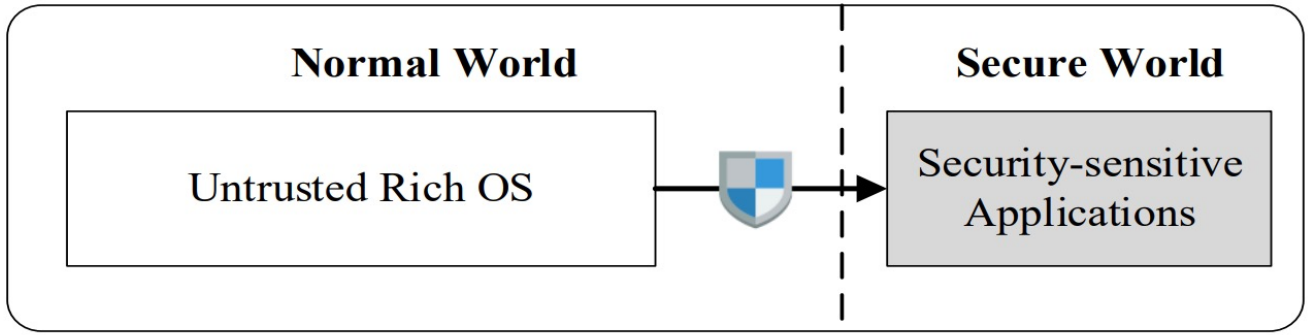
reluctant to open the secure world
for freely installing third-party applications

applications expect to be imported
into the secure world for an enhanced security protection

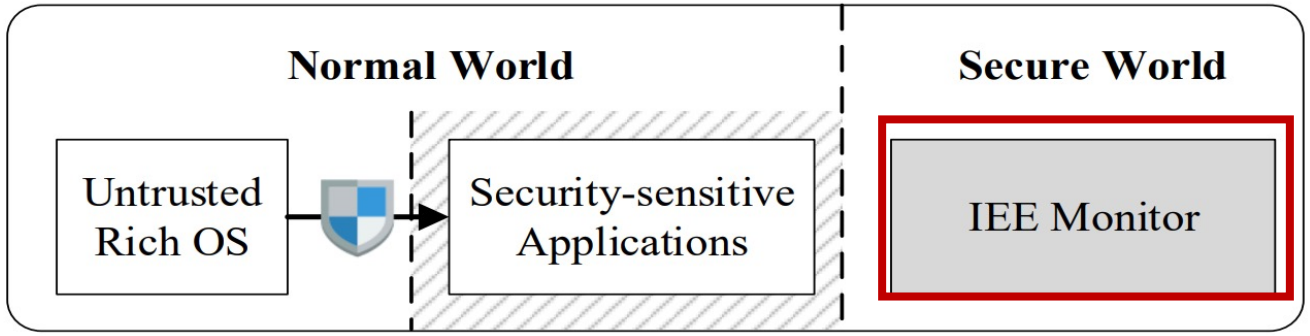
third-party security-sensitive applications

IEE Systems

(isolated execution environment)



(a) Architecture of TEE System

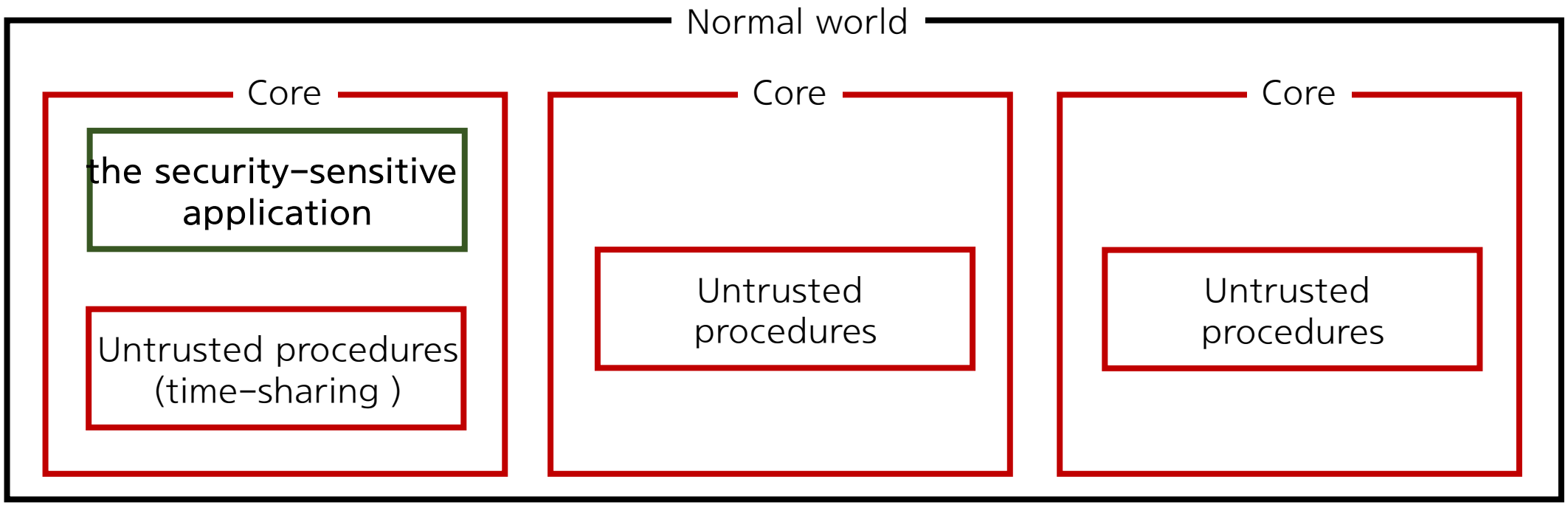


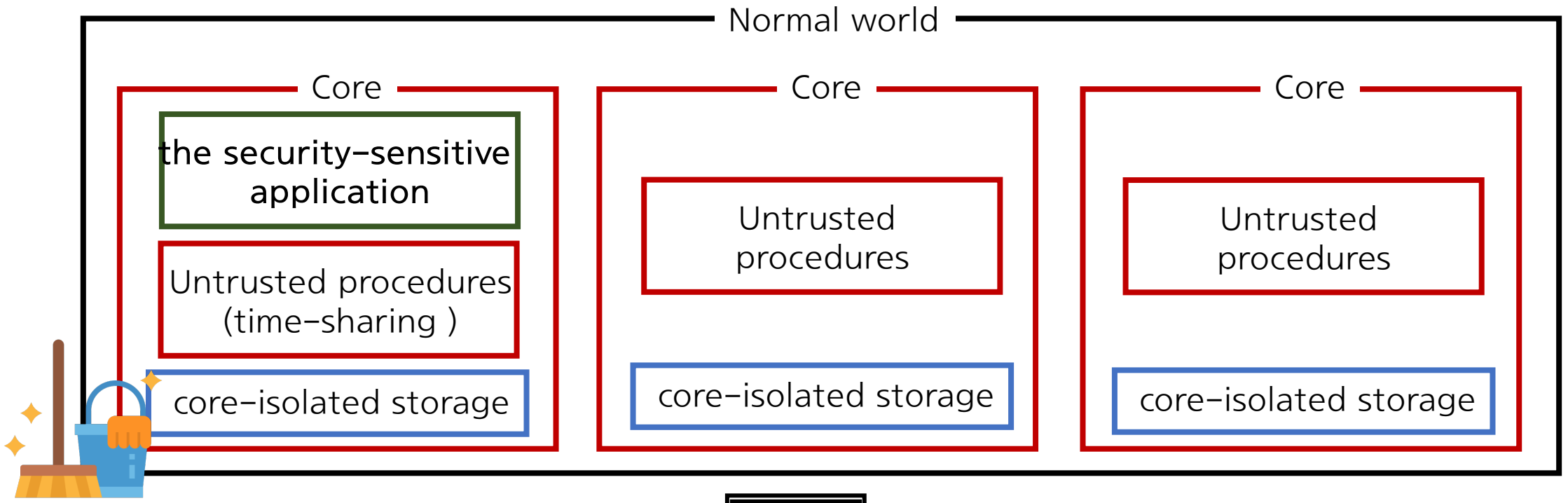
(b) Architecture of IEE System

Protected by IEE Monitor Protected by TrustZone

IEE Data Protection

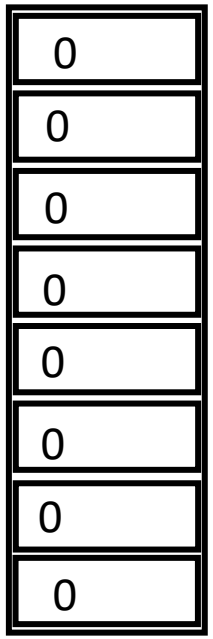
Model 1: Untrusted procedures are allowed to run concurrently with a security-sensitive application on two (or more) different cores in the normal world.





cleaning the core-isolated storage during the "switch out" process

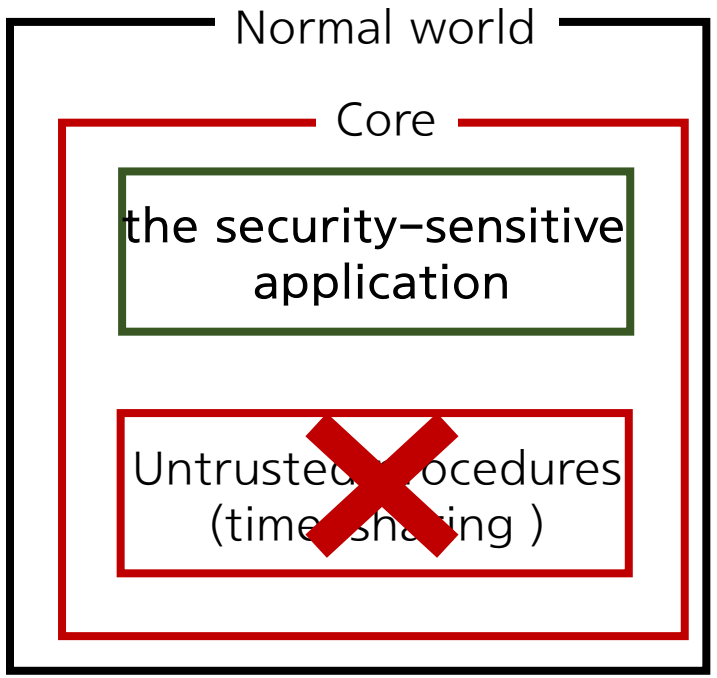
(overwriting the IEE memory with random data or all zero data)



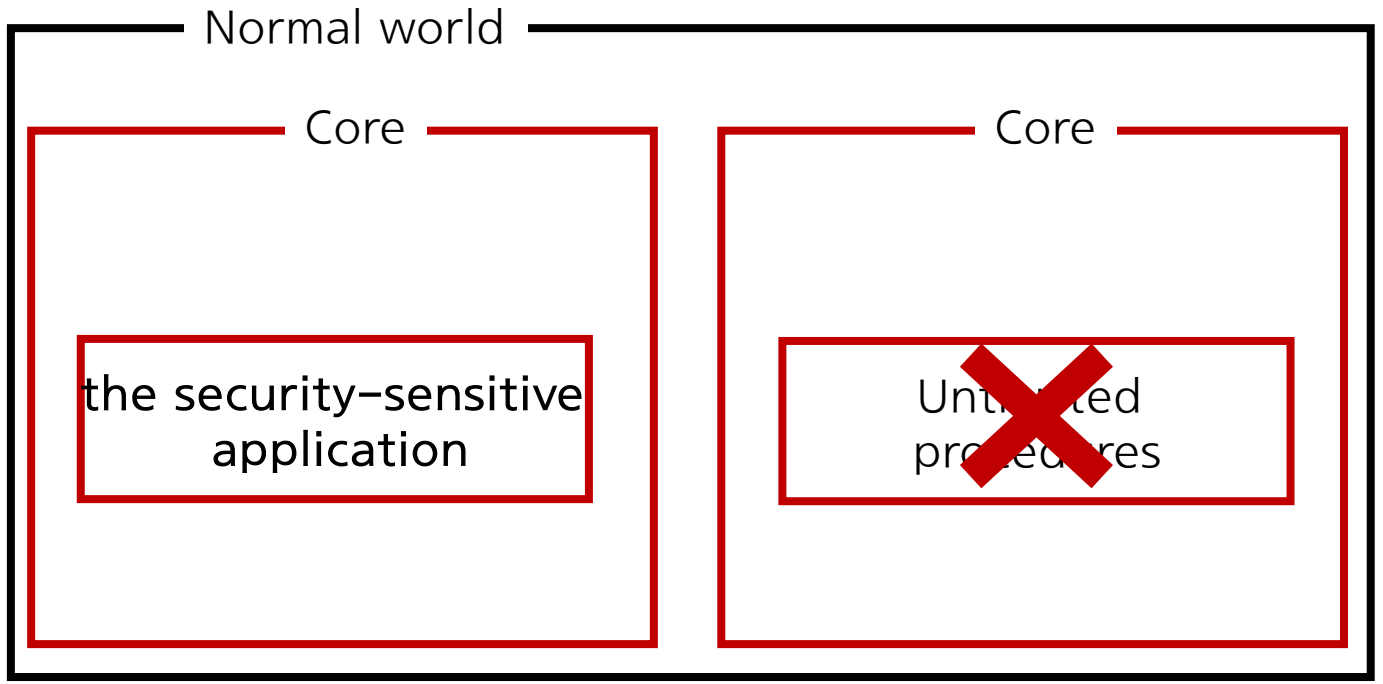
restoring the core-isolated storage or allocating blank core-isolated storage during the "switch in" process

Model 2: Untrusted procedures are NOT allowed to run concurrently with security-sensitive applications in the normal world.

On single-core platforms,



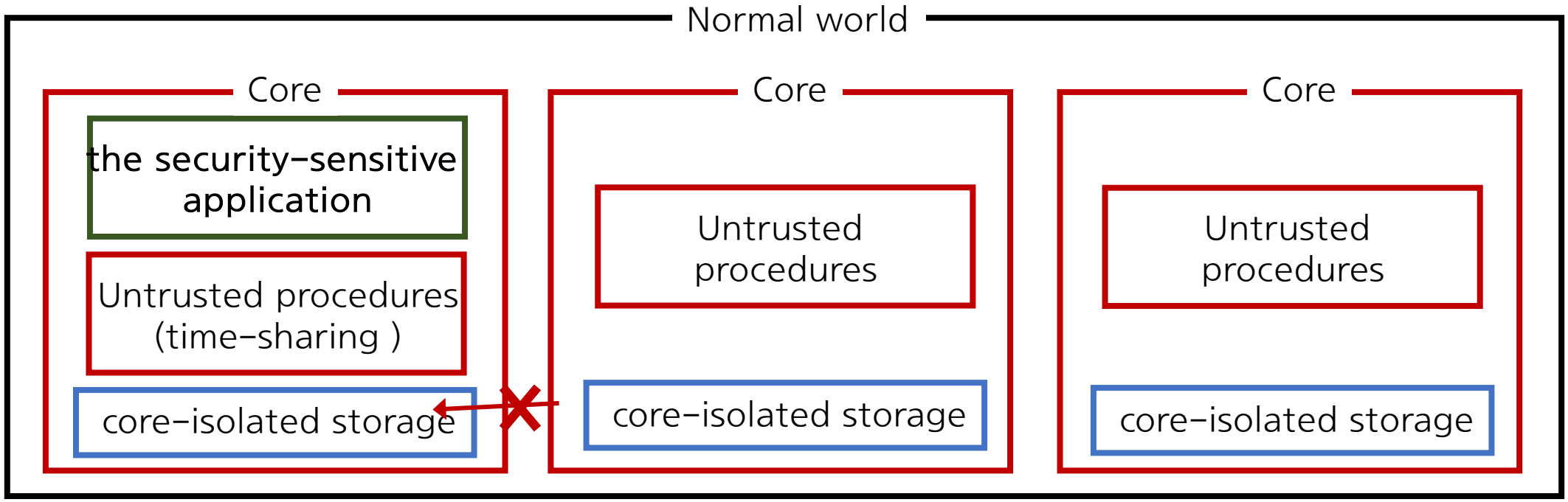
On multi-core platforms,



no need to allocate core-isolated storage.

security measures

during the IEE's context switching processes



inaccessible → accessible

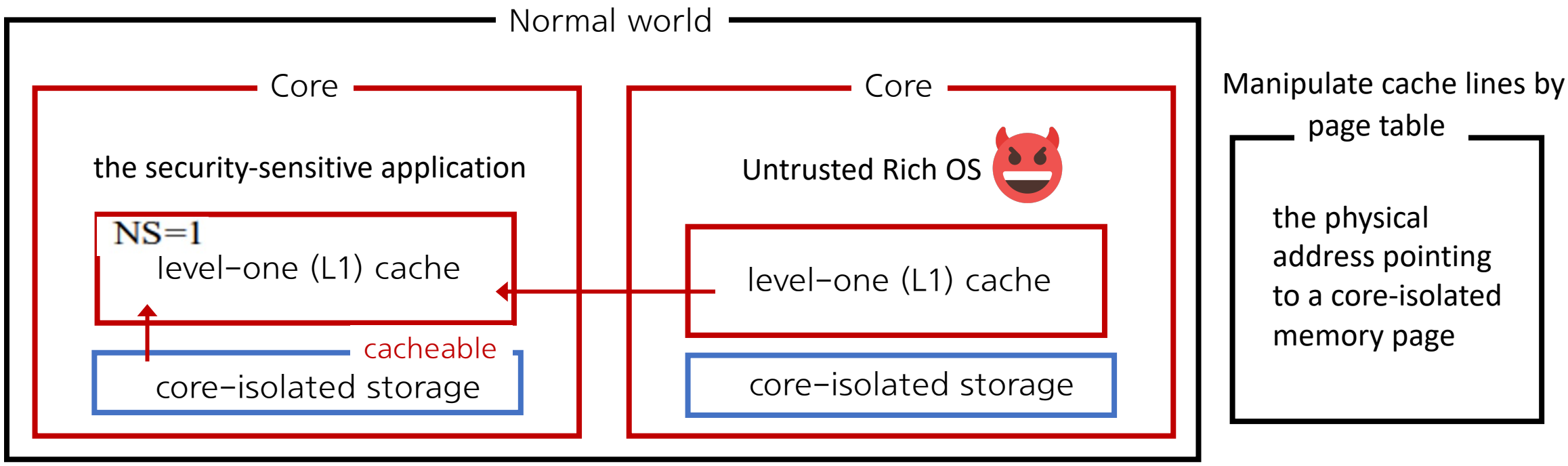
during the "switch out" process

during the "switch in" process

CITM Attack Types

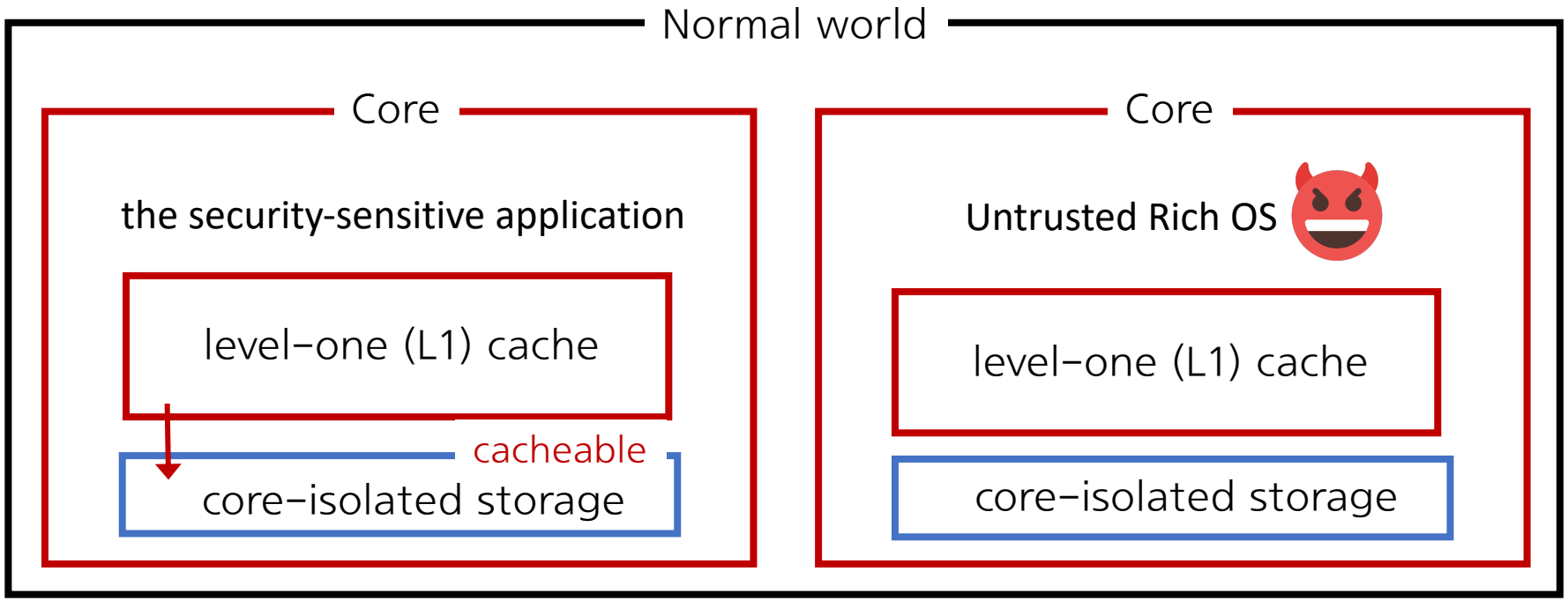
Type I. Manipulating core-isolated memory during concurrent execution.

When the core-isolated memory is set as cacheable,

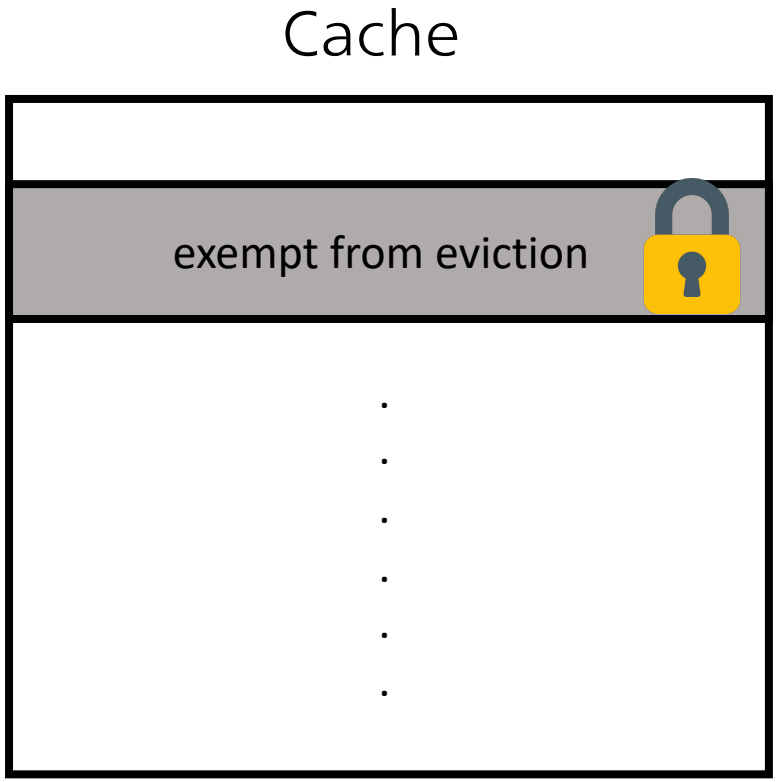


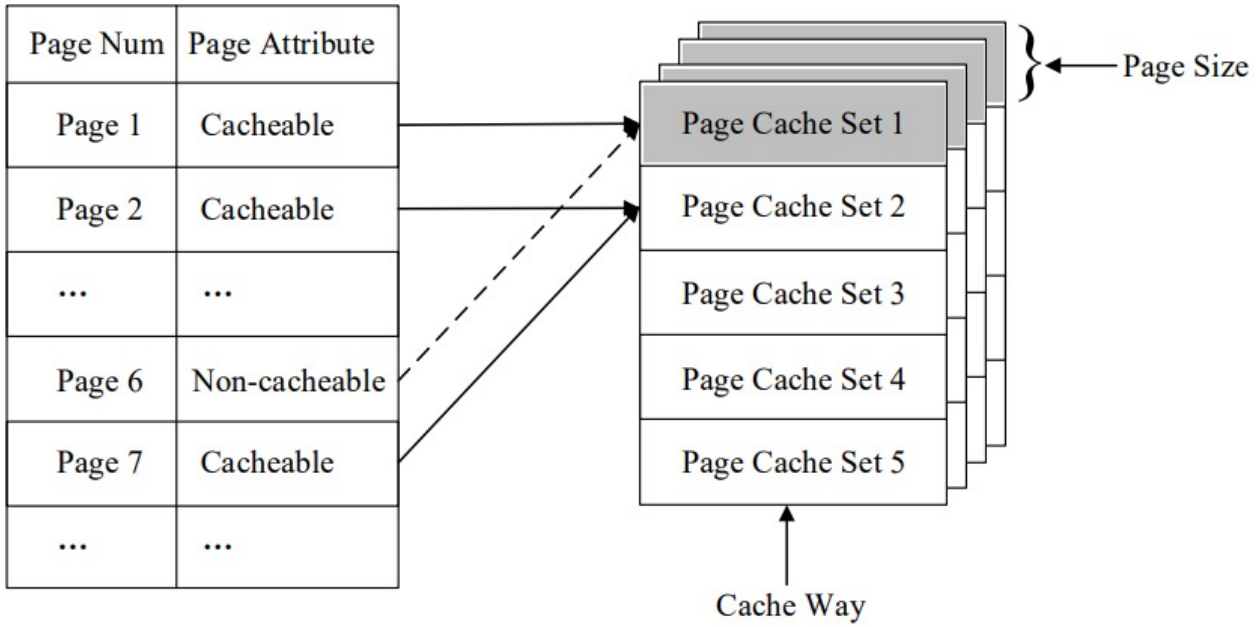
Type II. Bypassing security measures during IEE "switch out" process

constrain the memory

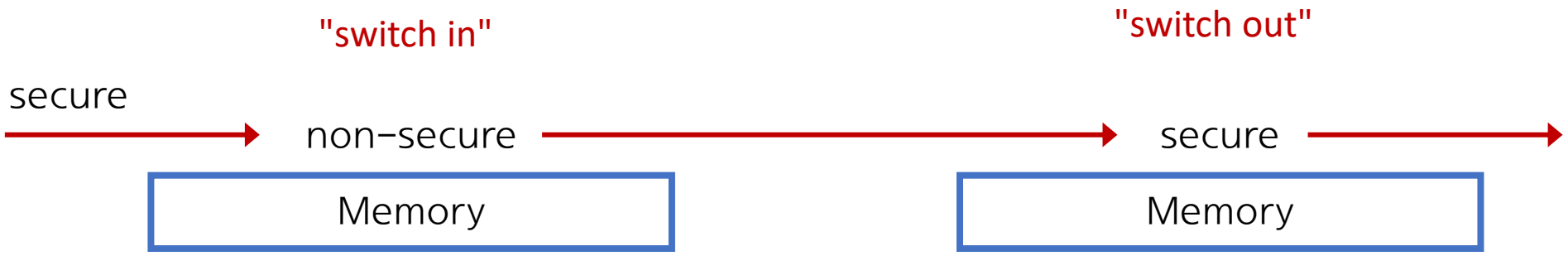


Cache Lockdown Technique





Type III. Misusing incomplete security measures during IEE's context switching.



the memory configuration is achieved through TZASC,
but the corresponding cache lines might be still non-secure
in the normal world

- Model 1 may suffer from all three identified attacks
- Model 2 vulnerable to Type II and III attacks

Type I and III attacks –> only when memory is employed
Type II attack –> only when the security measures for "switch out" process are performed

CASE STUDY OF CITM ATTACKS

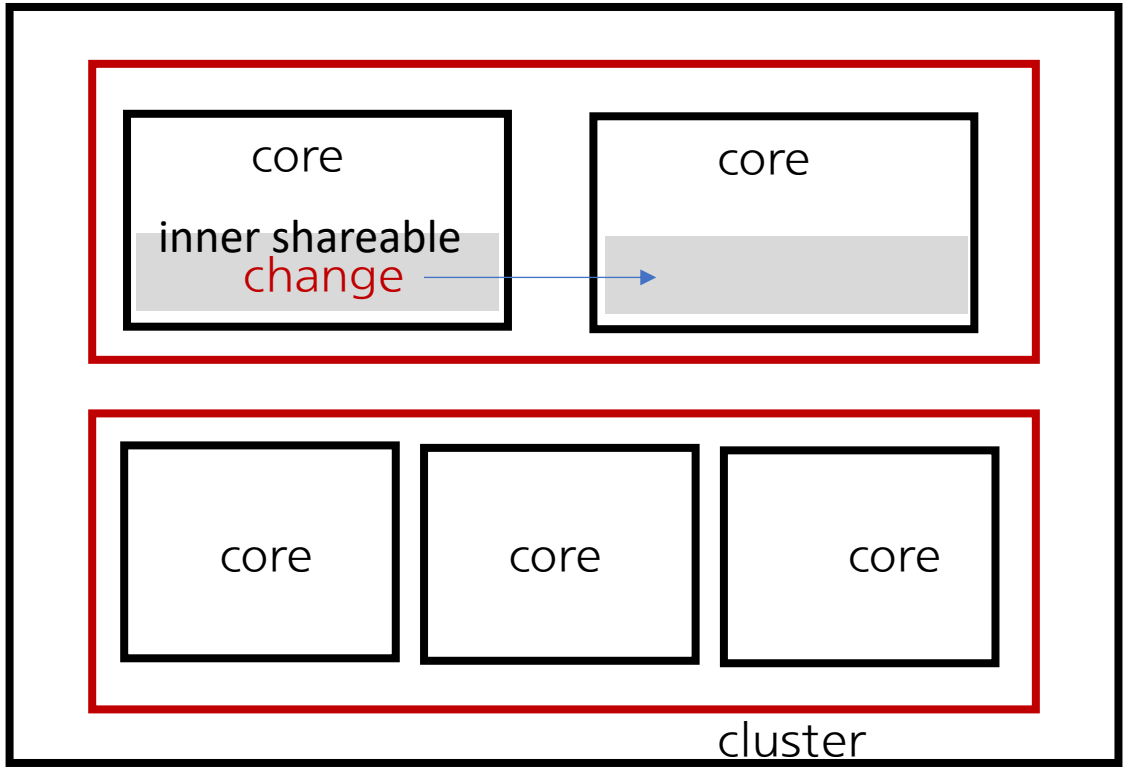
On SANCTUARY, Ginseng, and TrustICE

SANCTUARY

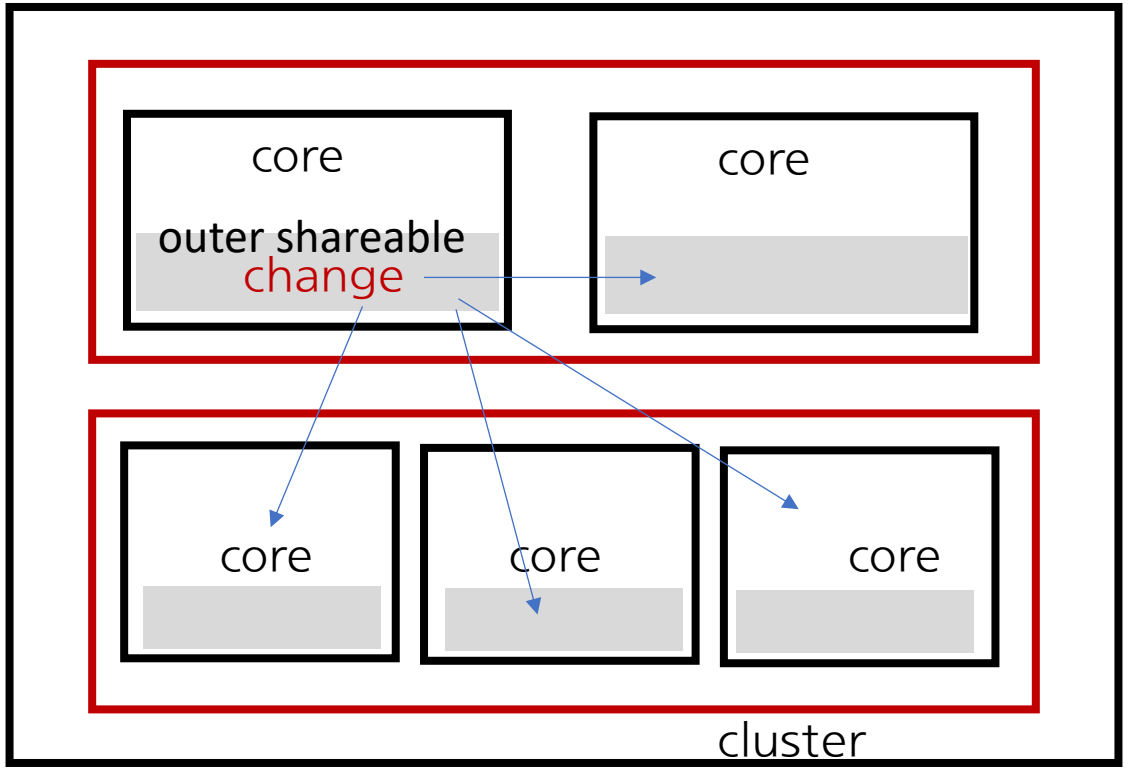
The shareability attribute

- shareability

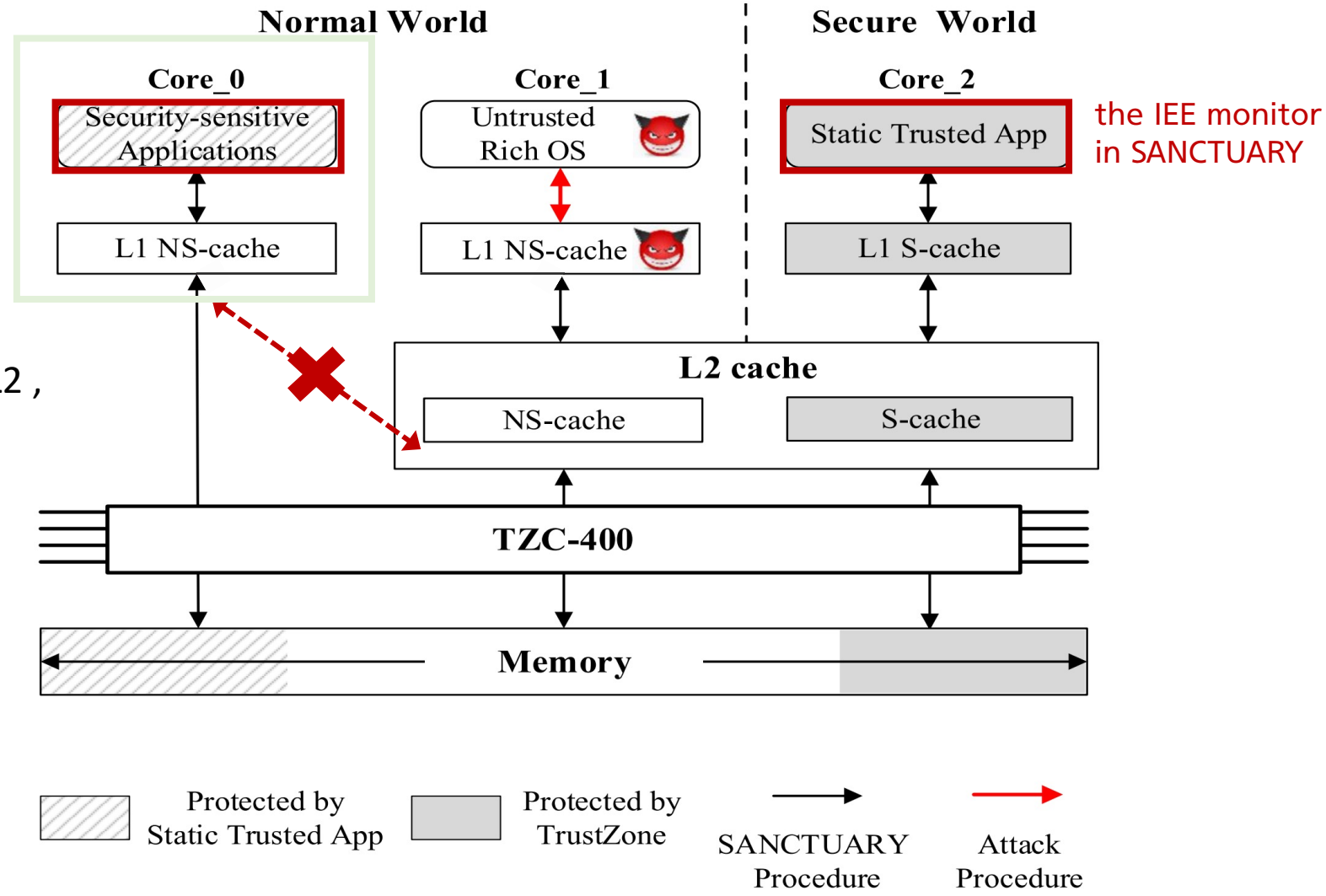
  the inner shareability domain
  the outer shareability domain



The shareability attribute



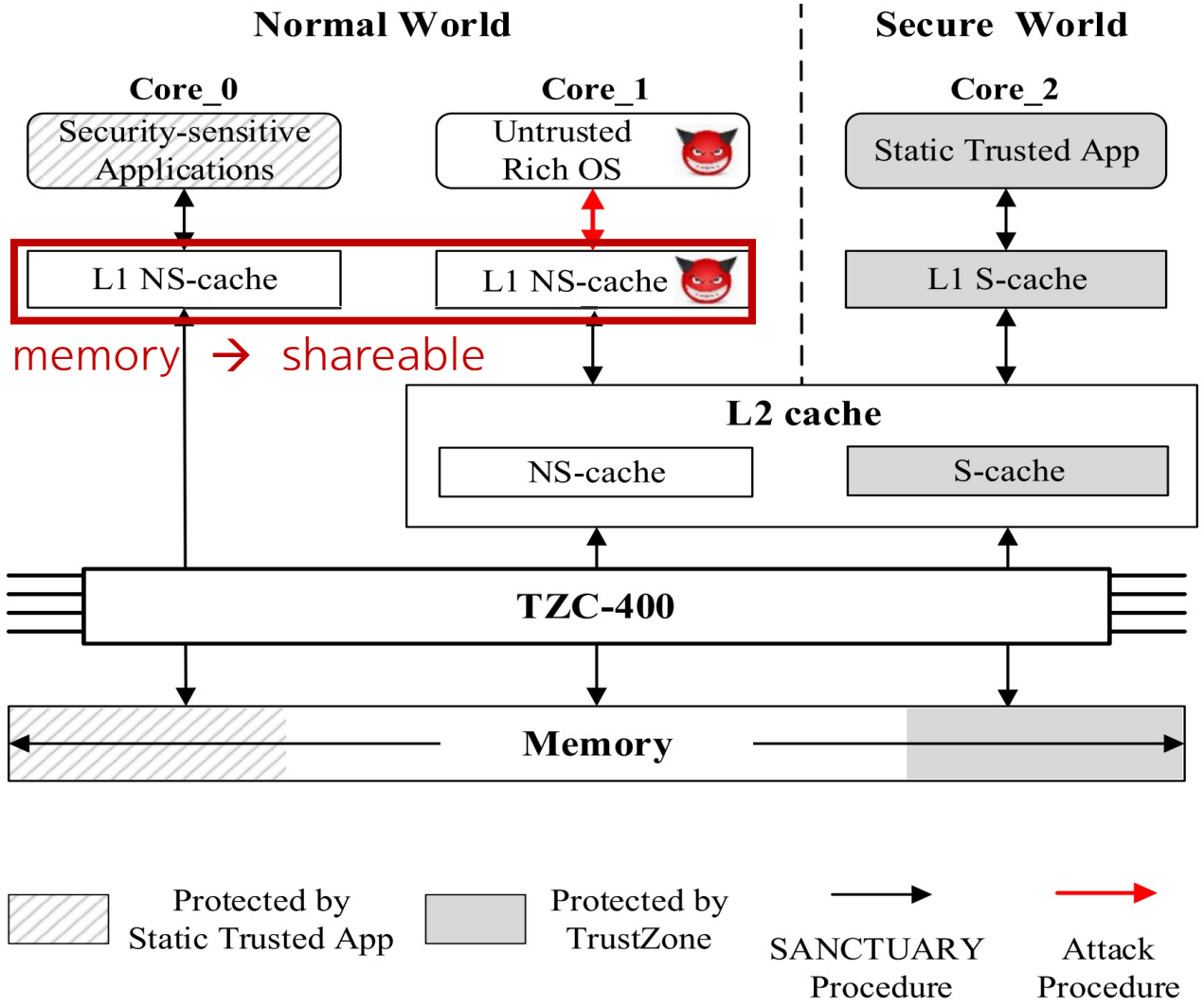
Attacking Procedure



disable the usage of L2 ,
memory is protected

Attacking Procedure

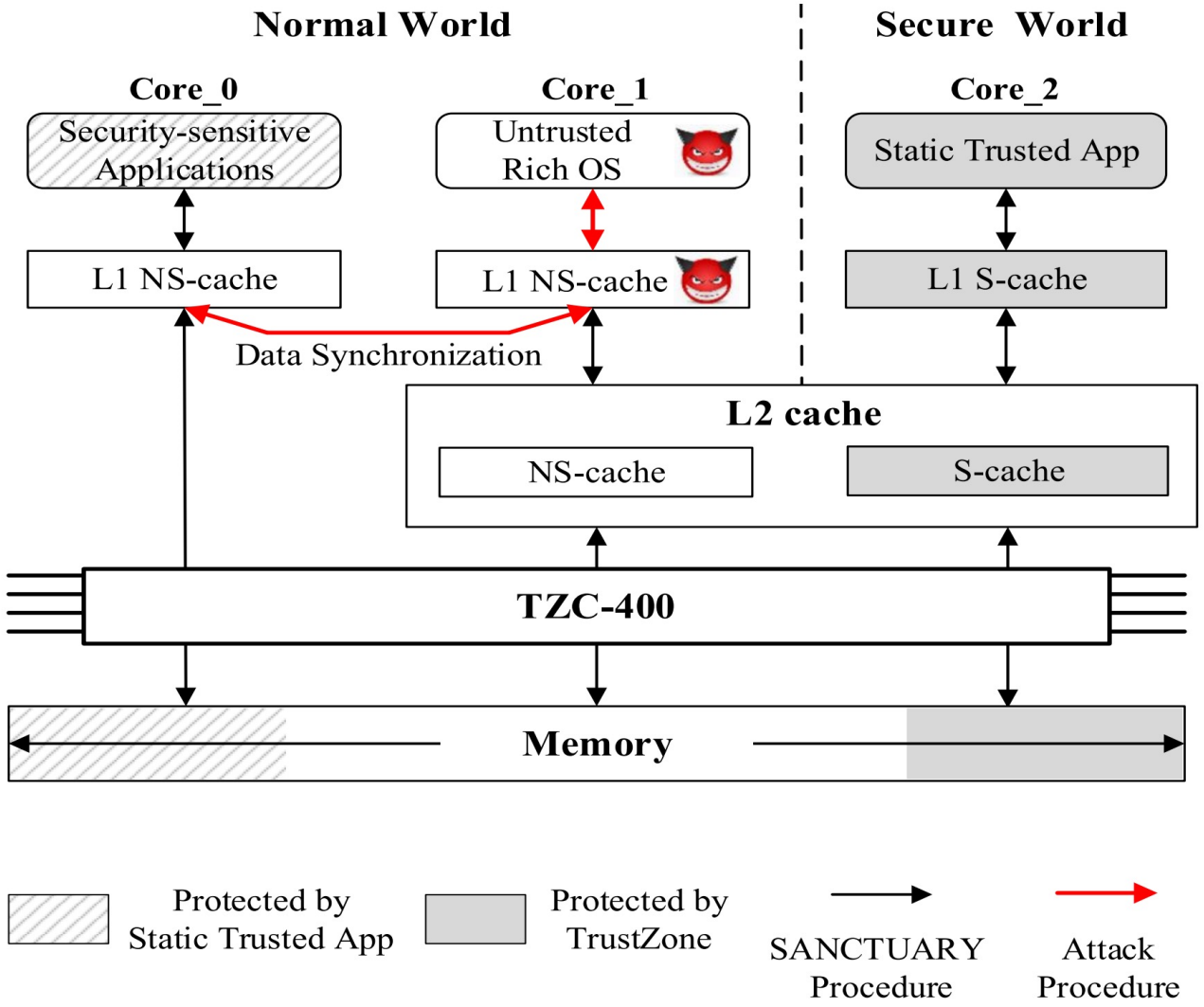
In Linux kernel,
all the cacheable memory → shareable



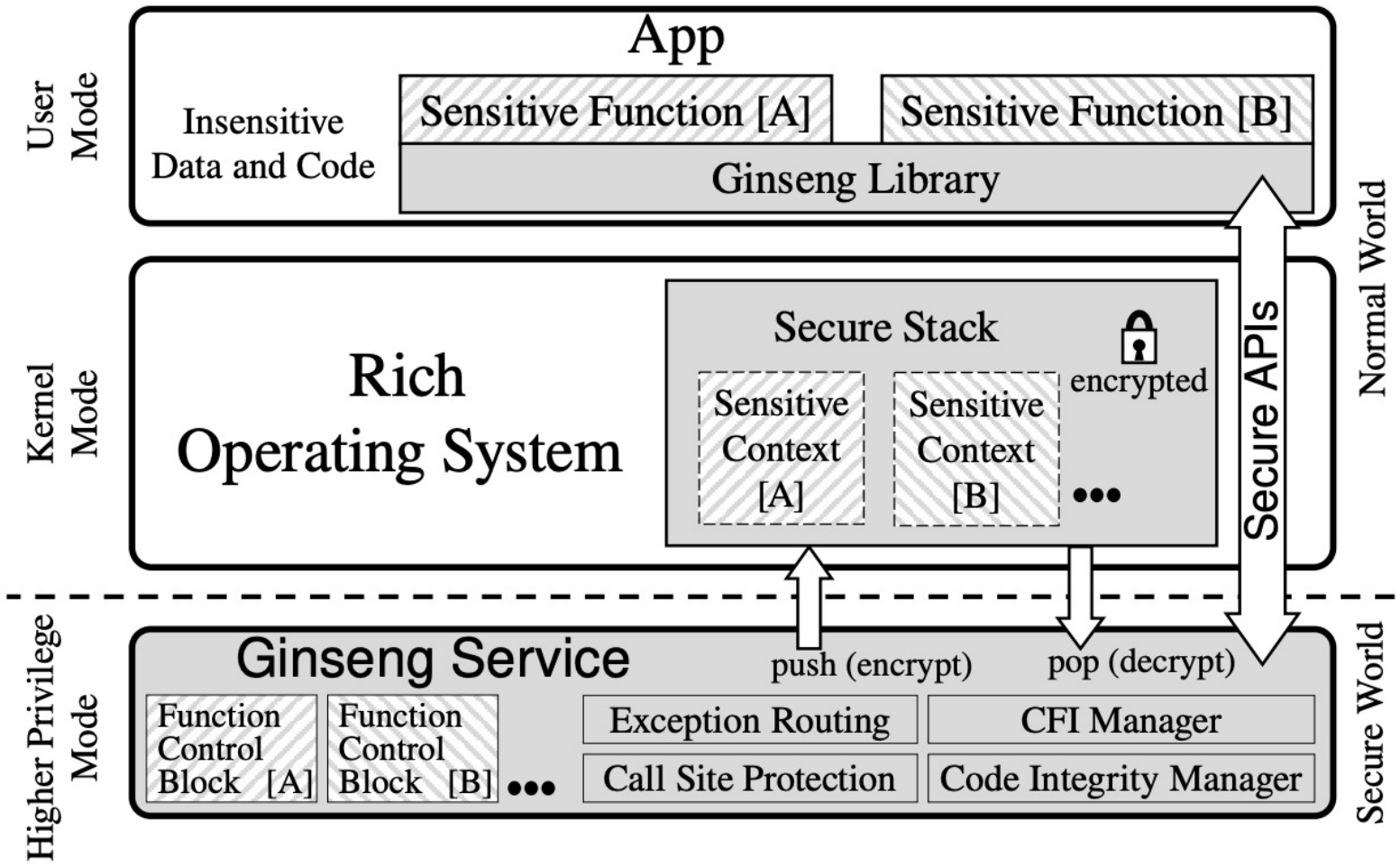
Attacking Procedure

page table

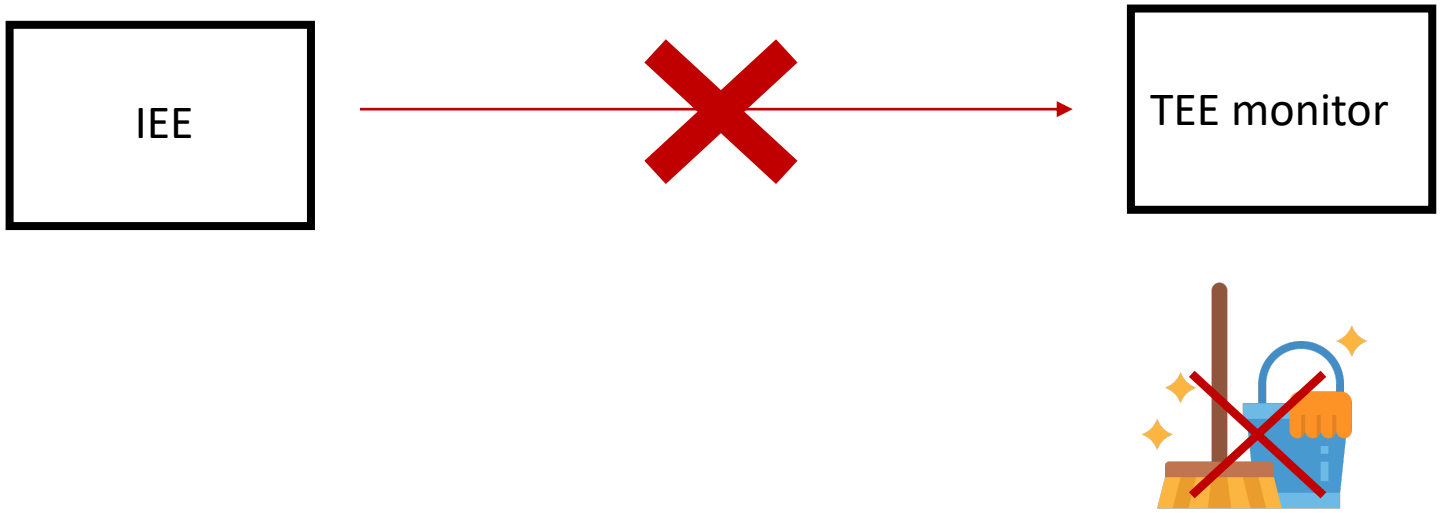
the physical address pointing to a memory page of core_0



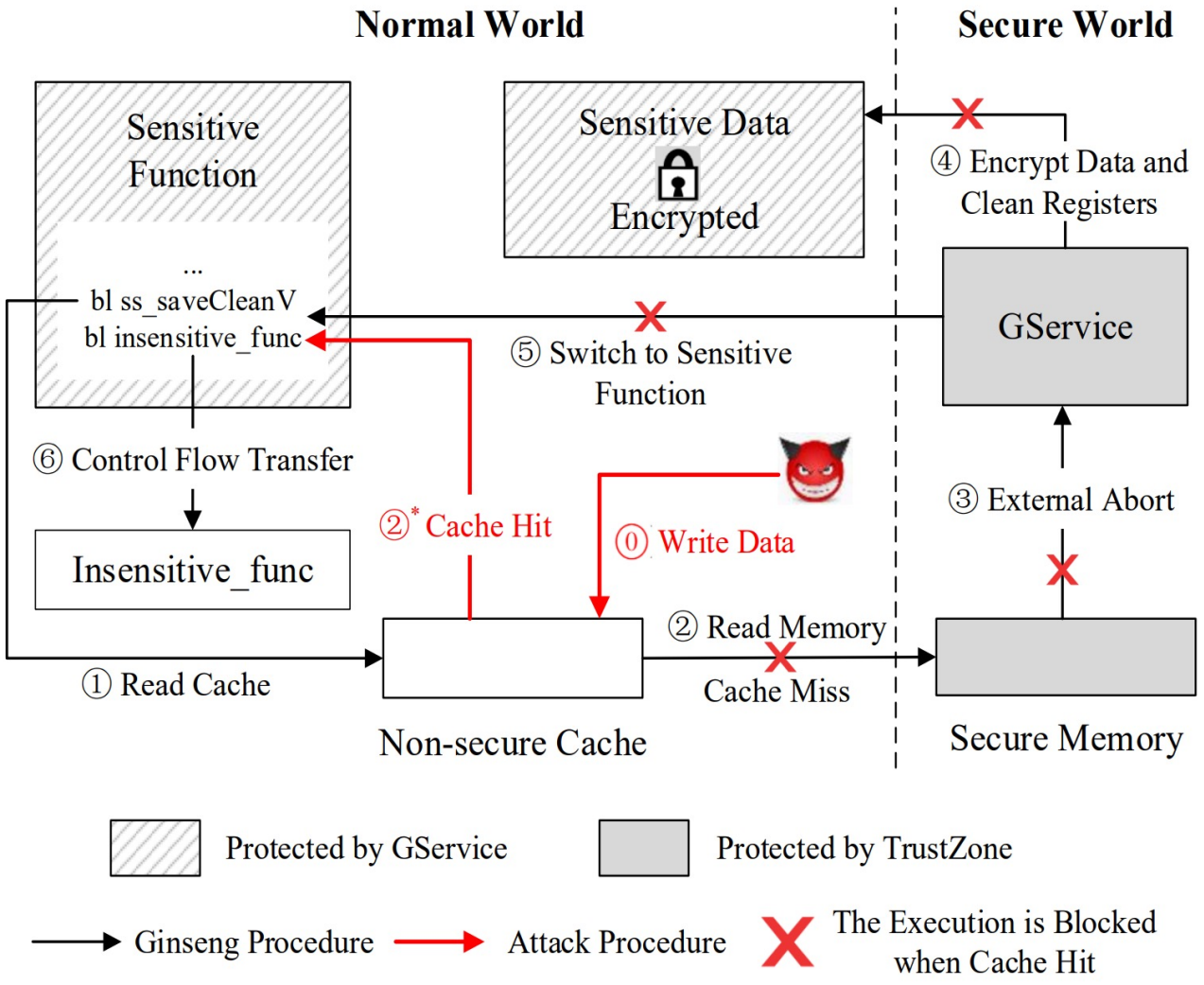
Ginseng: Mapping to Non-Secure Cache



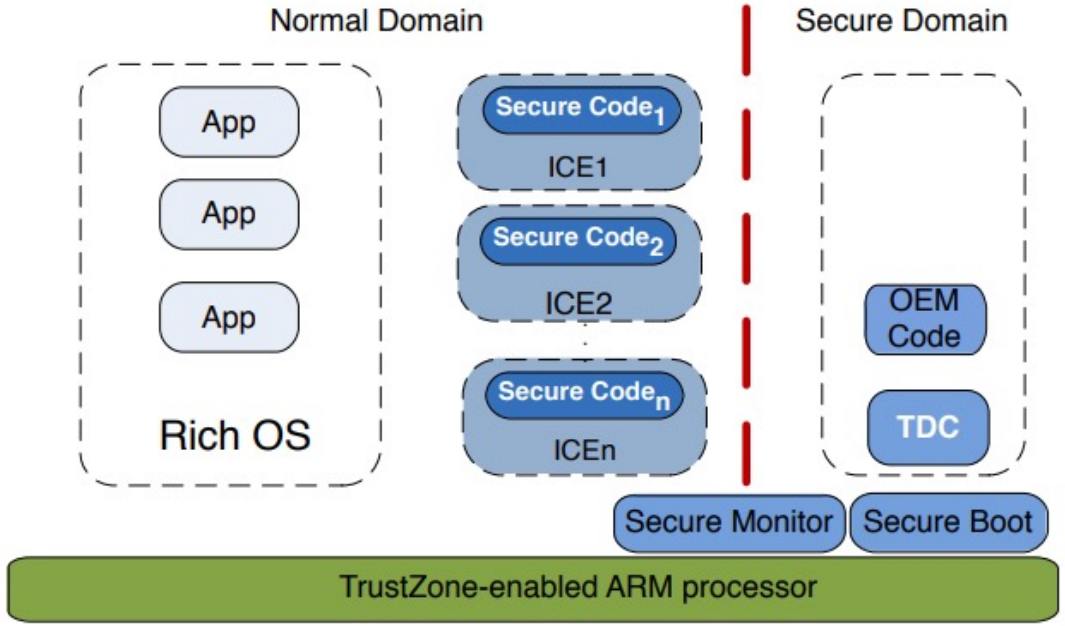
bypass the data cleaning

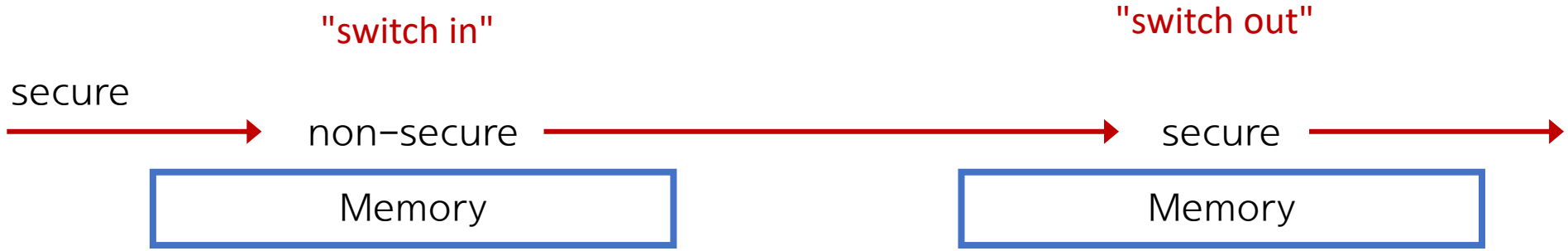


Attacking Procedure

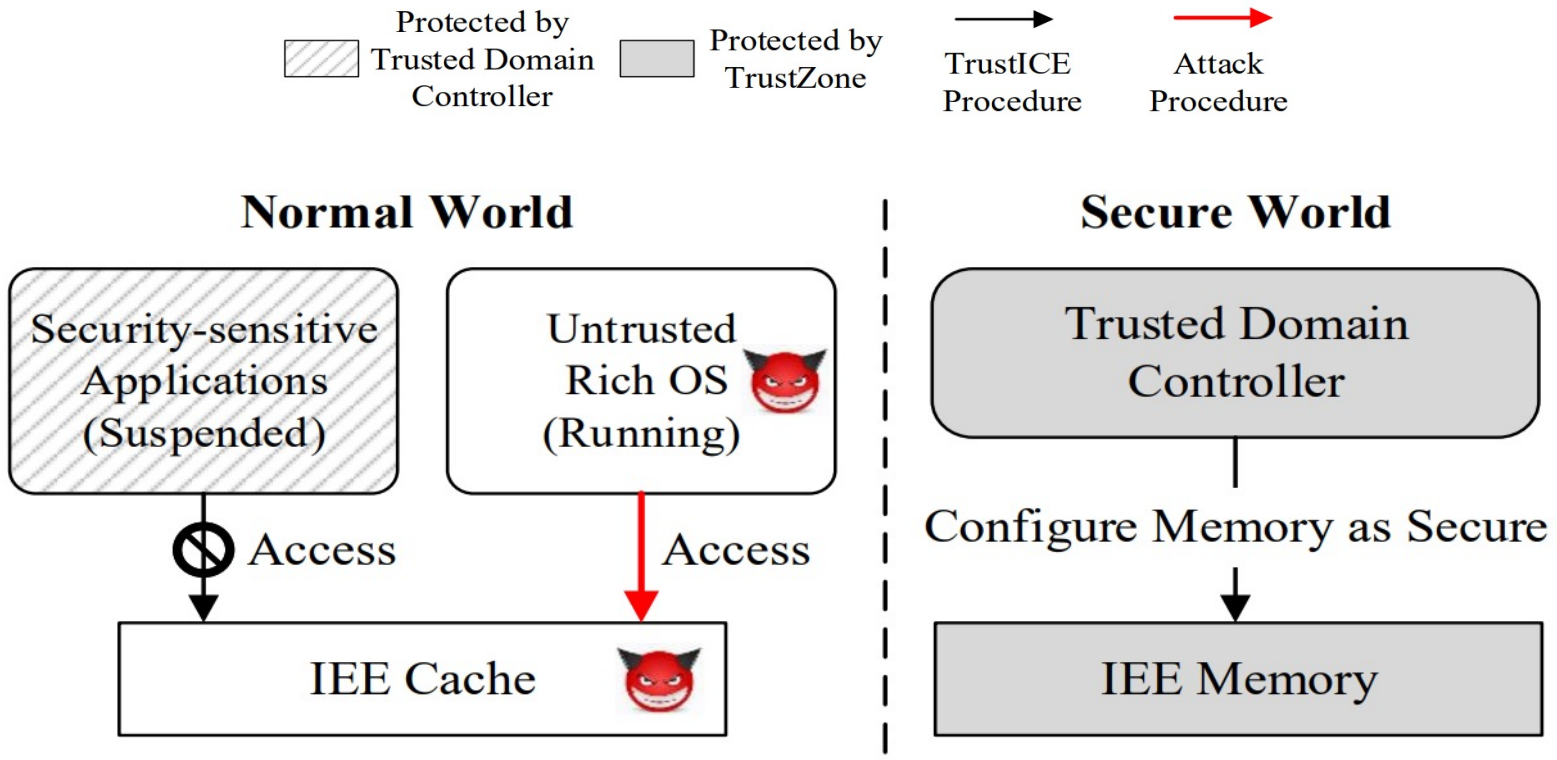


TrustICE: Incomplete Cache Cleaning





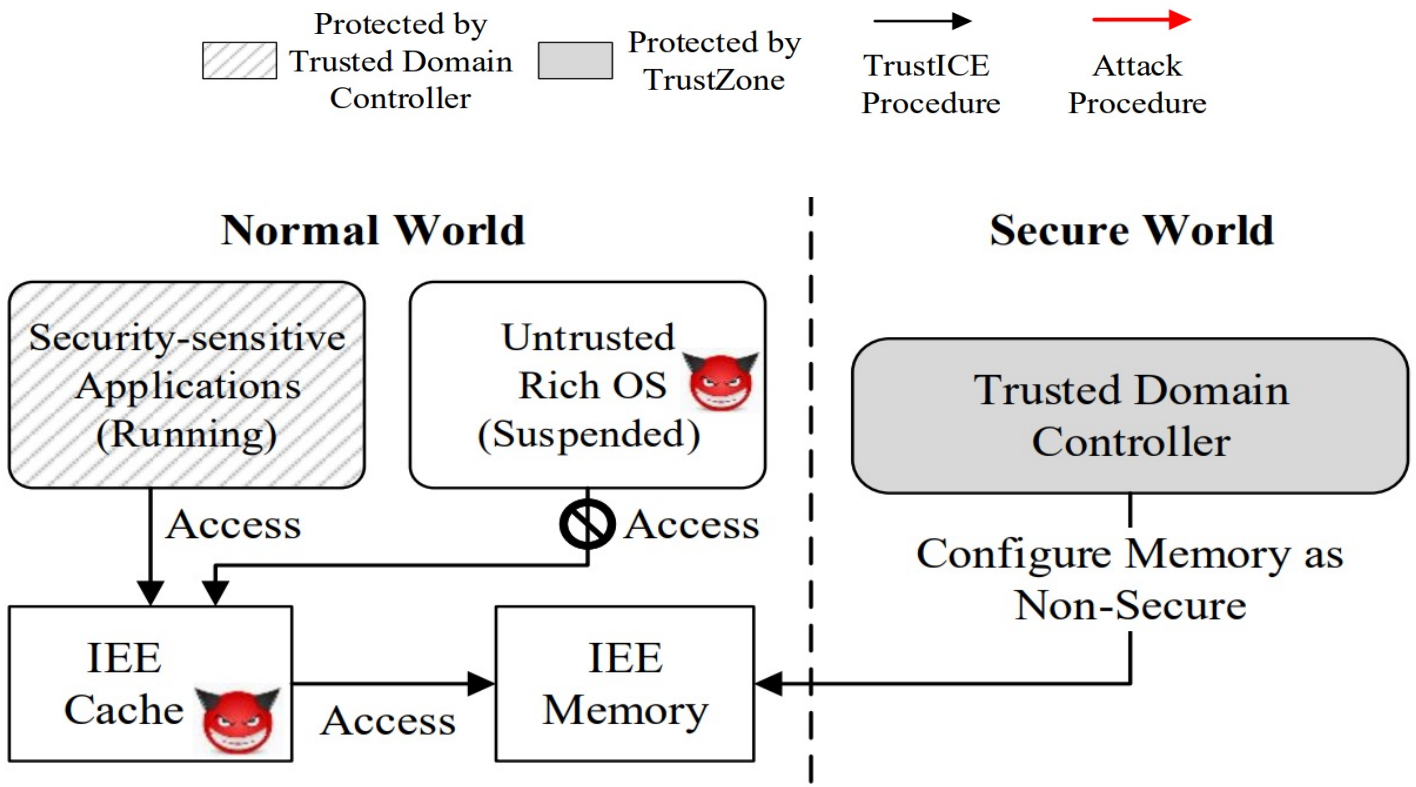
Attacking Procedure



page table

the physical address pointing to a IEE memory page

Attacking Procedure



COUNTERMEASURE

Defense Approaches

Because of incoherence between cache and main memory*!*

1. the memory isolation does not automatically guarantee the cache isolation

(Type I attack)

-> configuring the cache attributes as outer non-cacheable, non-shareable

2. the reading and writing operations are not synchronized between memory and cache

(Type II attack)

-> synchronizing the reading and writing operations between memory and cache

3. the security attribute of a cache line is determined by the status of the core who accesses it, not by TZASC

(Type III attack)

-> cleaning the cache lines during both "switch in" and "switch out" processes, so that attackers could not read residual sensitive data or retain malicious data in the cache.