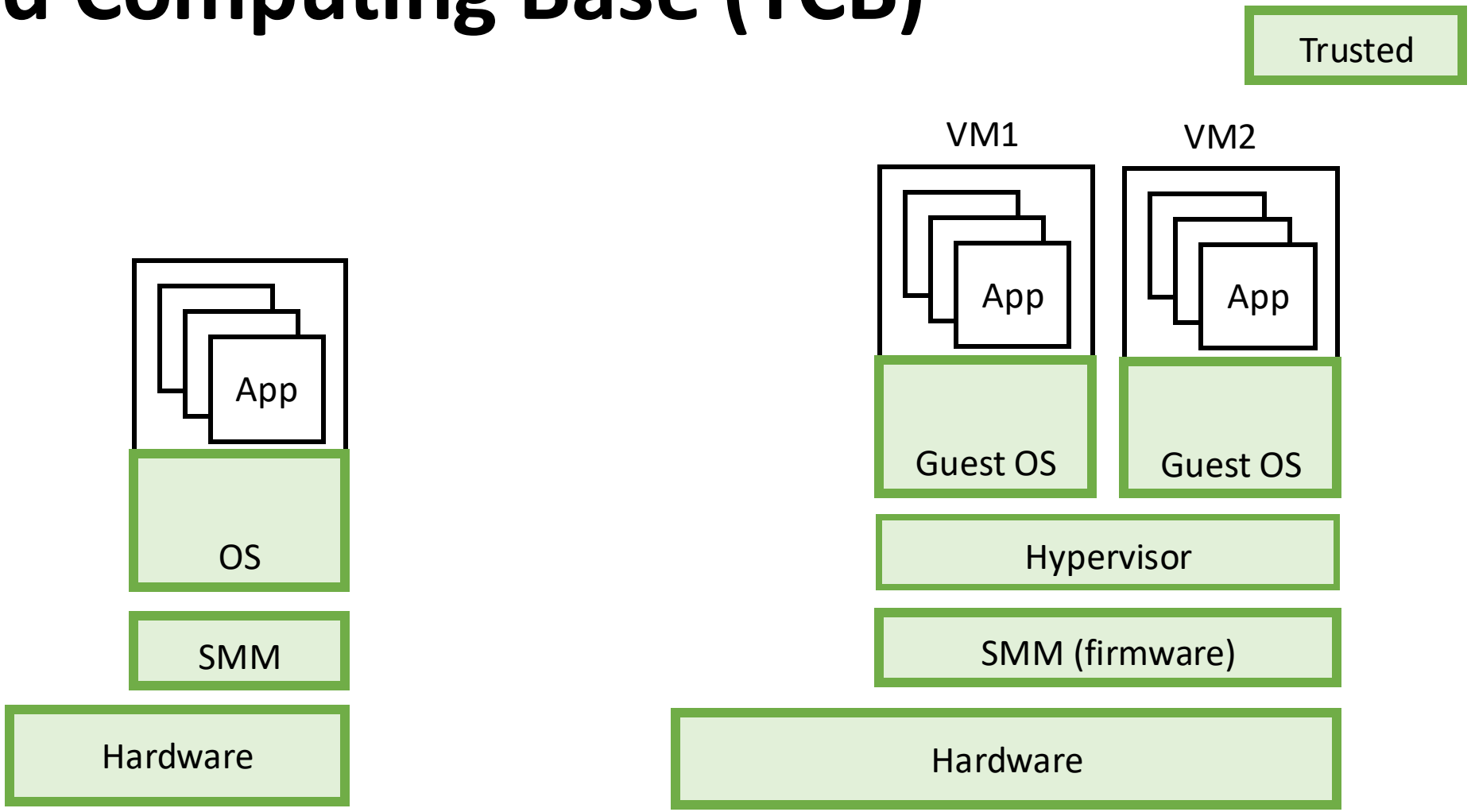
Hardware-supported Trusted Execution Environment (TEE)

Mengjia Yan

Spring 2025

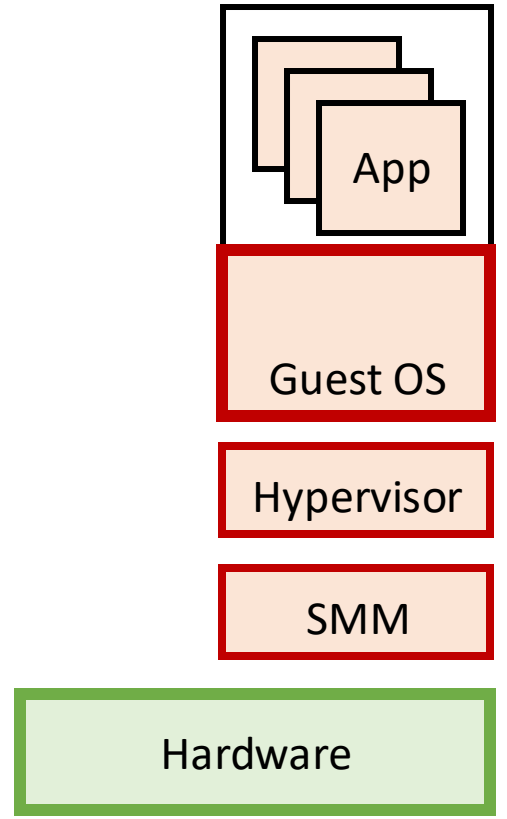
What do we trust when we execute a program?

Trusted Computing Base (TCB)



Shrink TCB. Why?

- 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



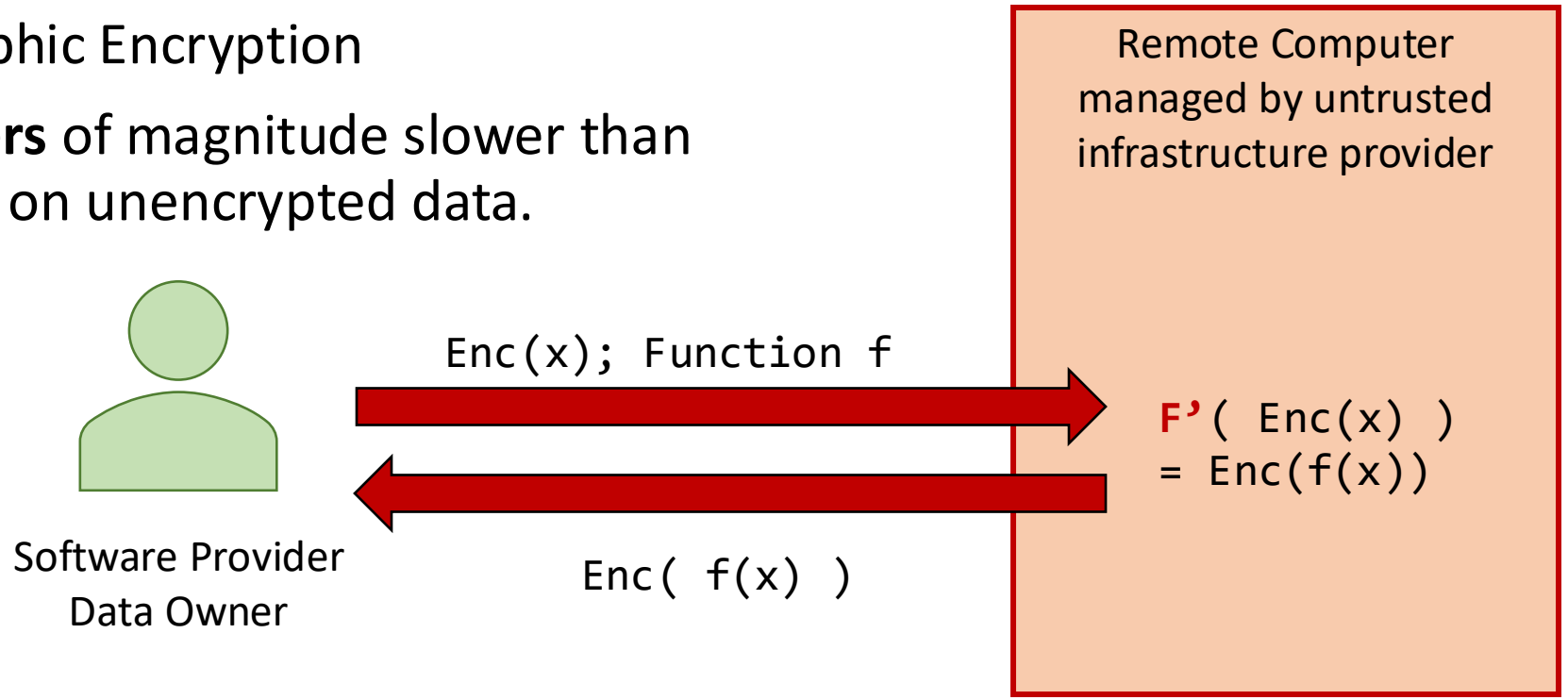
Secure Remote Computing

- Example: DNA Analysis

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

Potential Solutions

- Homomorphic Encryption
- **4 to 5 orders** of magnitude slower than computing on unencrypted data.

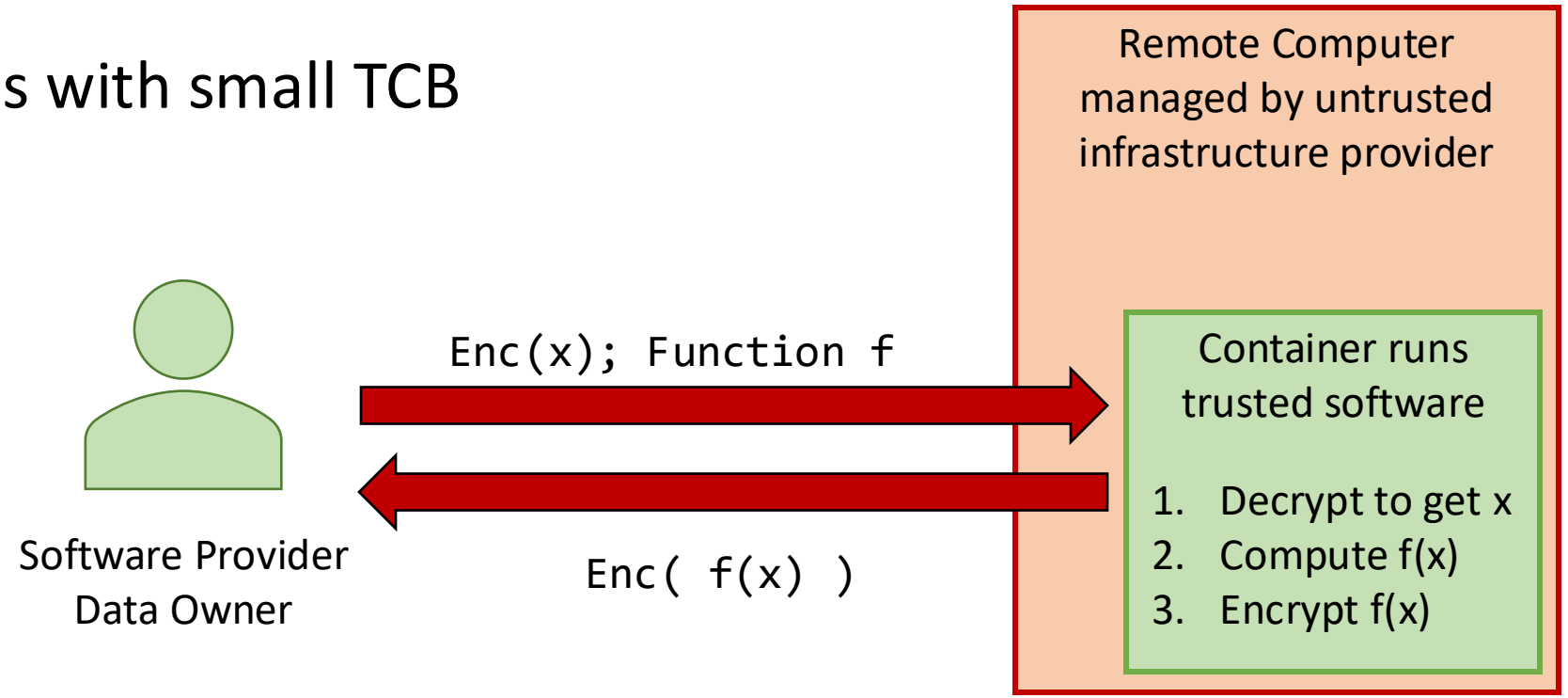


- Performance? Accelerators?

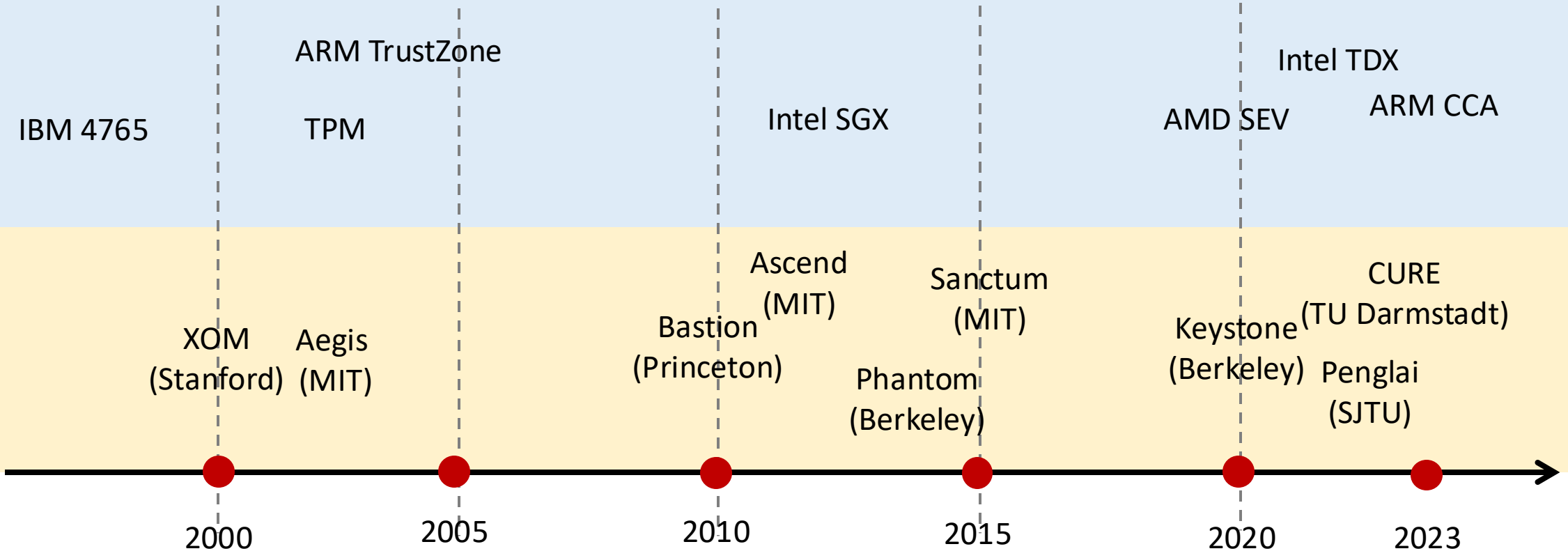
e.g., F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption; Axel Feldmann, Nikola Samardzic et al. MICRO'21

Potential Solutions

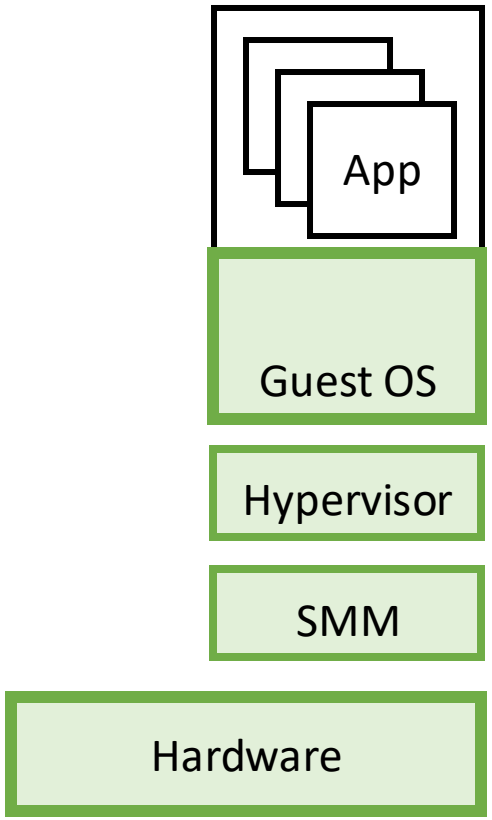
- Build TEEs with small TCB



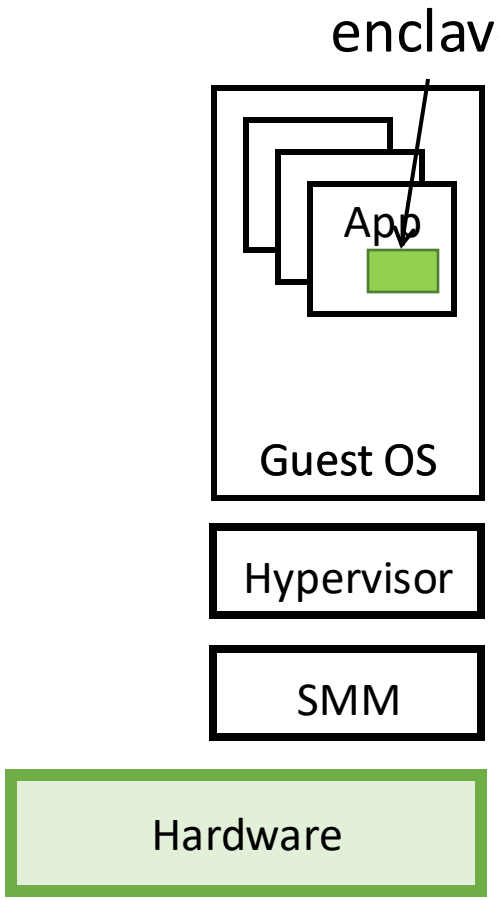
TEE Examples



Protection Granularity & TCB Size



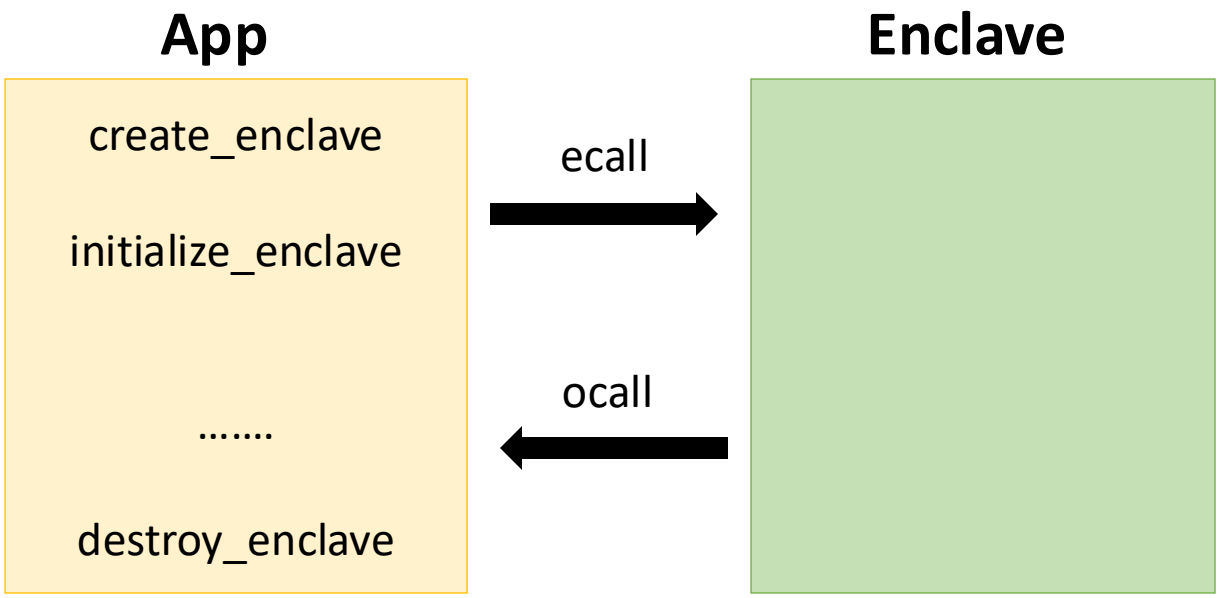
Traditional System



Intel SGX

SGX Enclave Programming Model

- Examples from: https://github.com/intel/linux-sgx



What Does Privileged SW Do?

```
./helloworld
```

- Operations at launch time:
  - Create a process (PID, status, etc.)
  - Create a virtual address space: allocate memory for stack, heap, code region, set up the page tables 
  - Setup file descriptor for input and output 
  - Load the binary into the code region, and linked library if needed 
  - Transfer the control to user space 

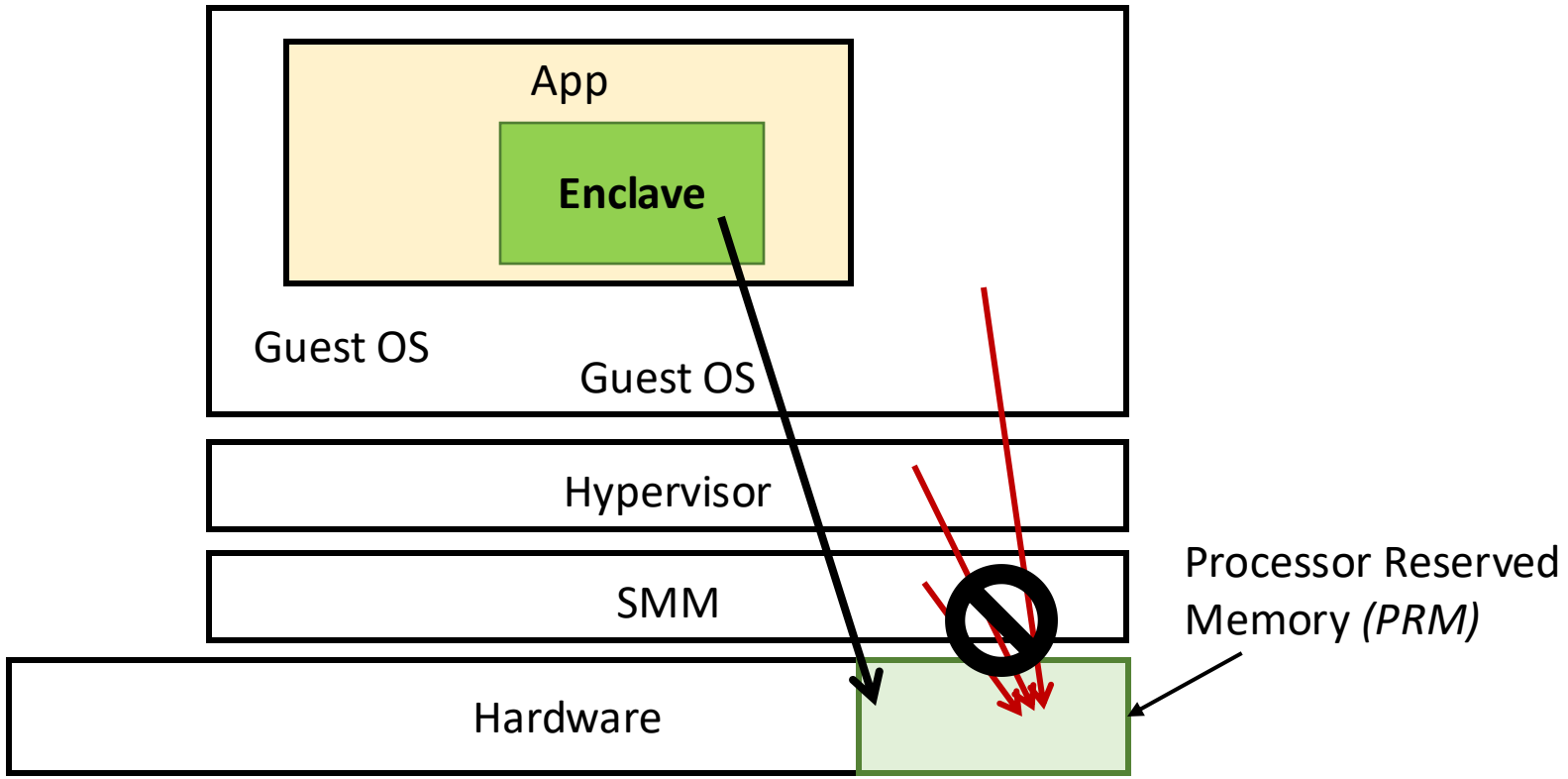
What can a privilege software attacker do?

- A non-comprehensive list
  - Modify the code to be executed
  - Monitor the whole execution process and data in register and in memory
  - Modify data in register and memory
  - Intercept IO, eavesdrop and tamper with the communication
  - ……

Case Study: Memory Management in Intel SGX

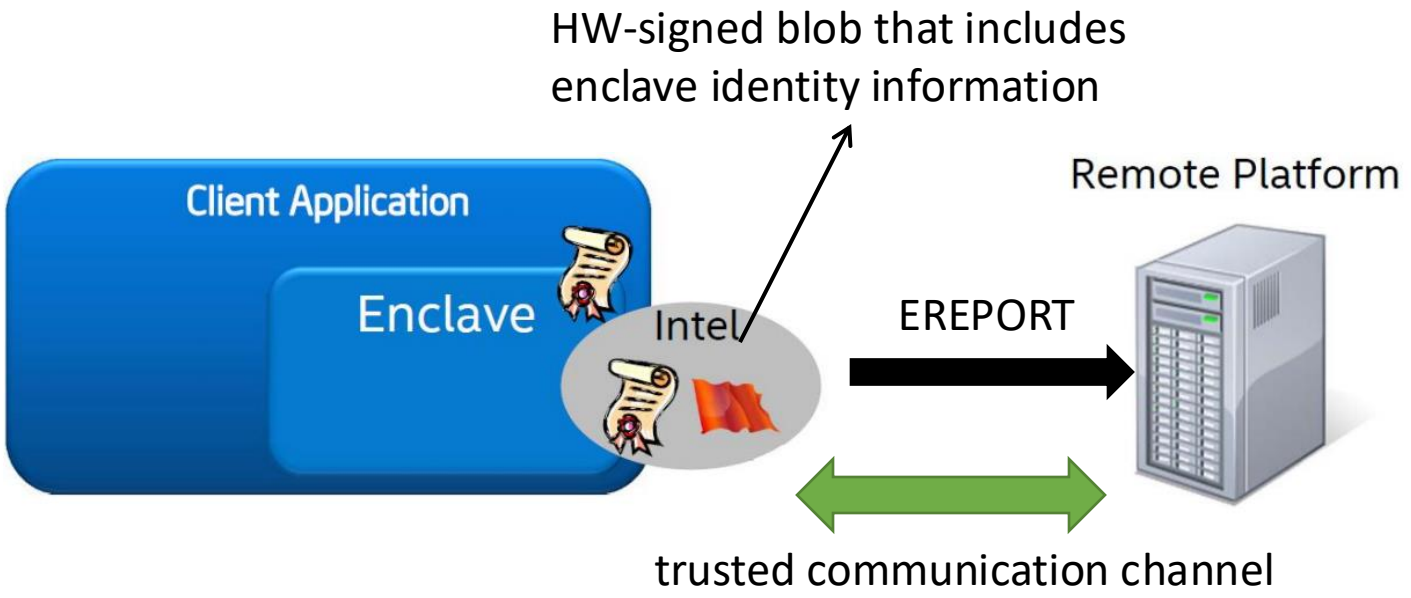
Intel SGX Overview

- Enclave code/data map to PRM; Different enclaves access their own memory region

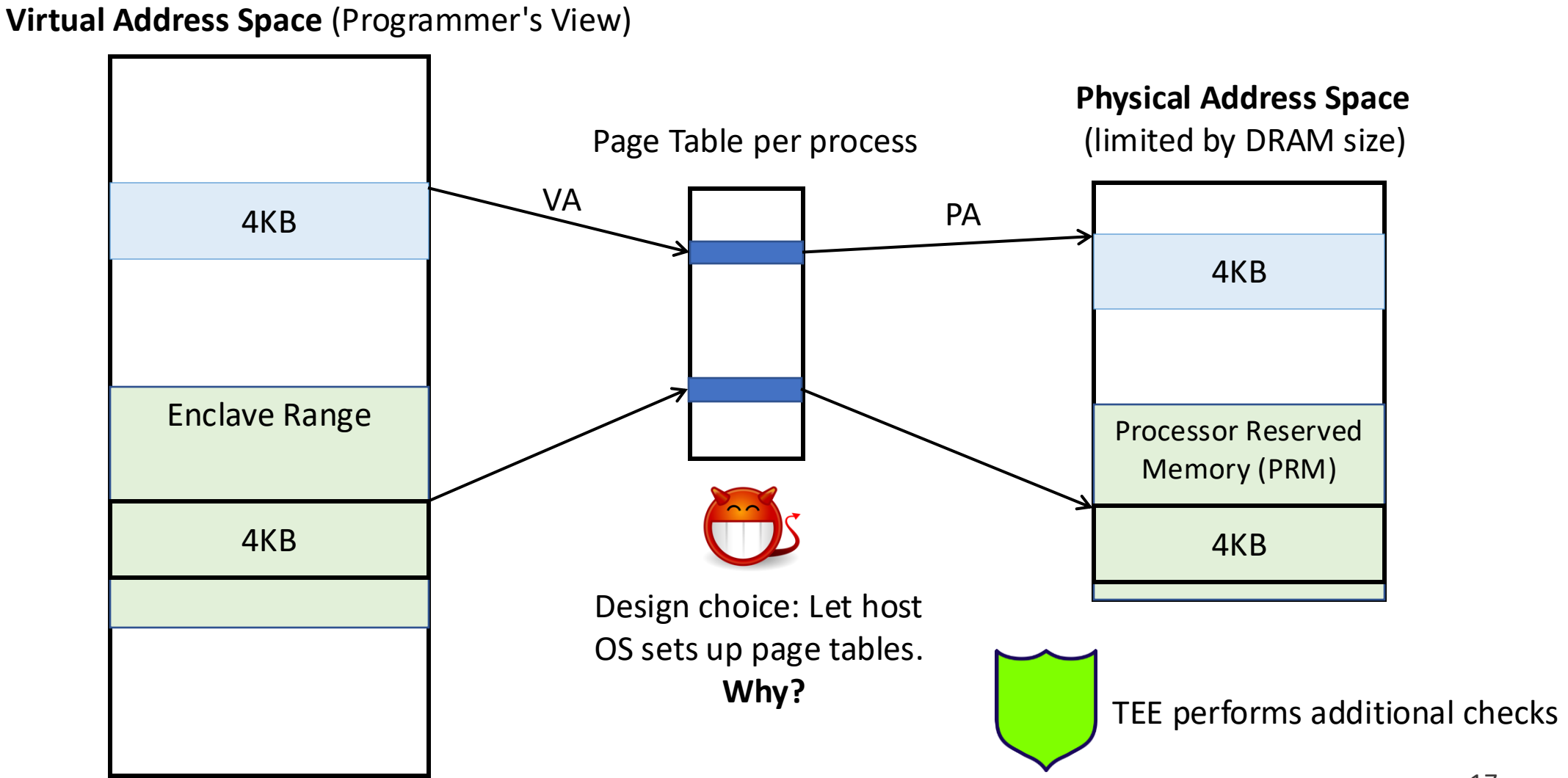


Enclave Attestation and Sealing

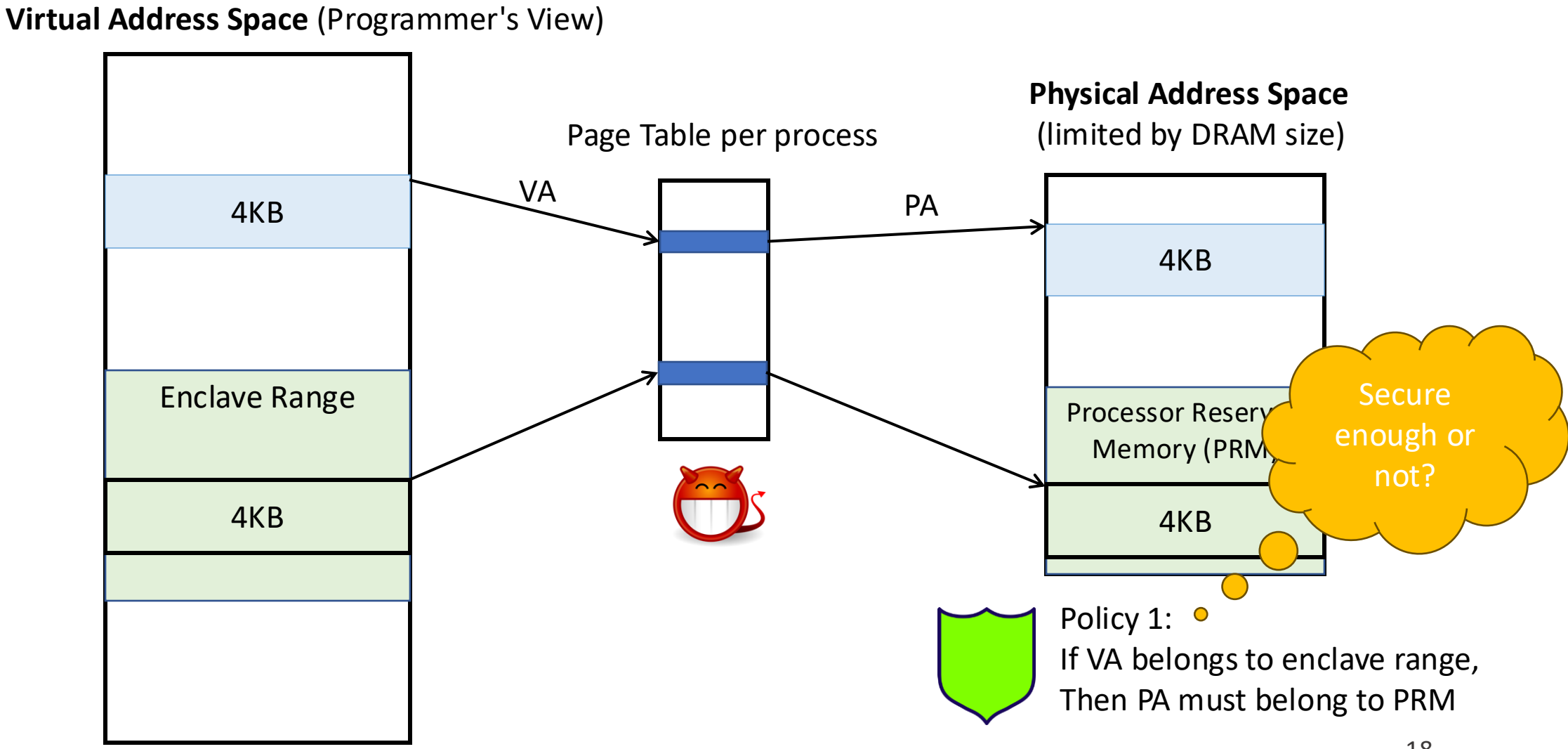
- HW based attestation provides evidence that “this is the right application executing on an authentic platform” (approach similar to secure boot attestation)



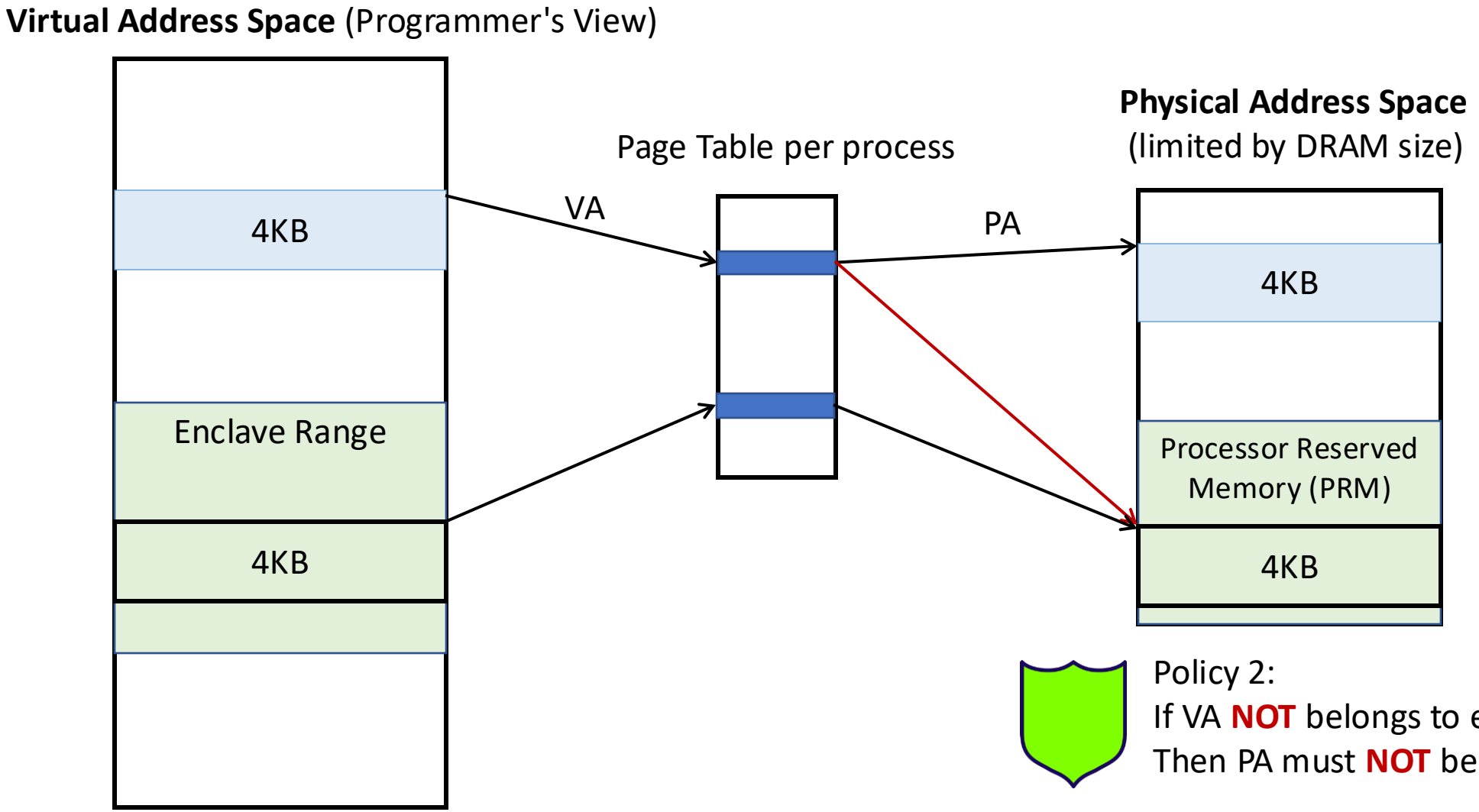
Intel SGX Address Translation Overview



Intel SGX Address Translation Overview

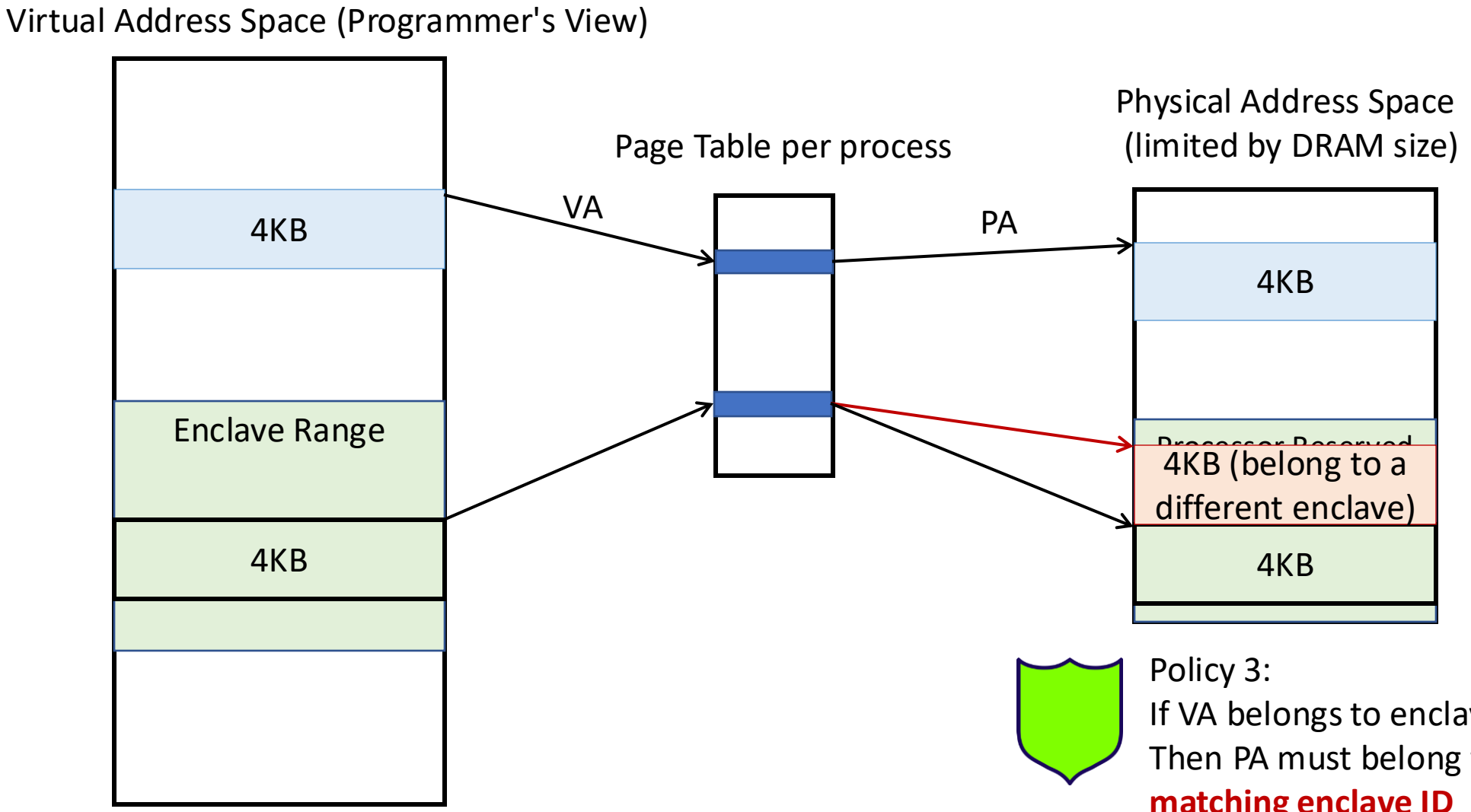


Malicious Translation #1



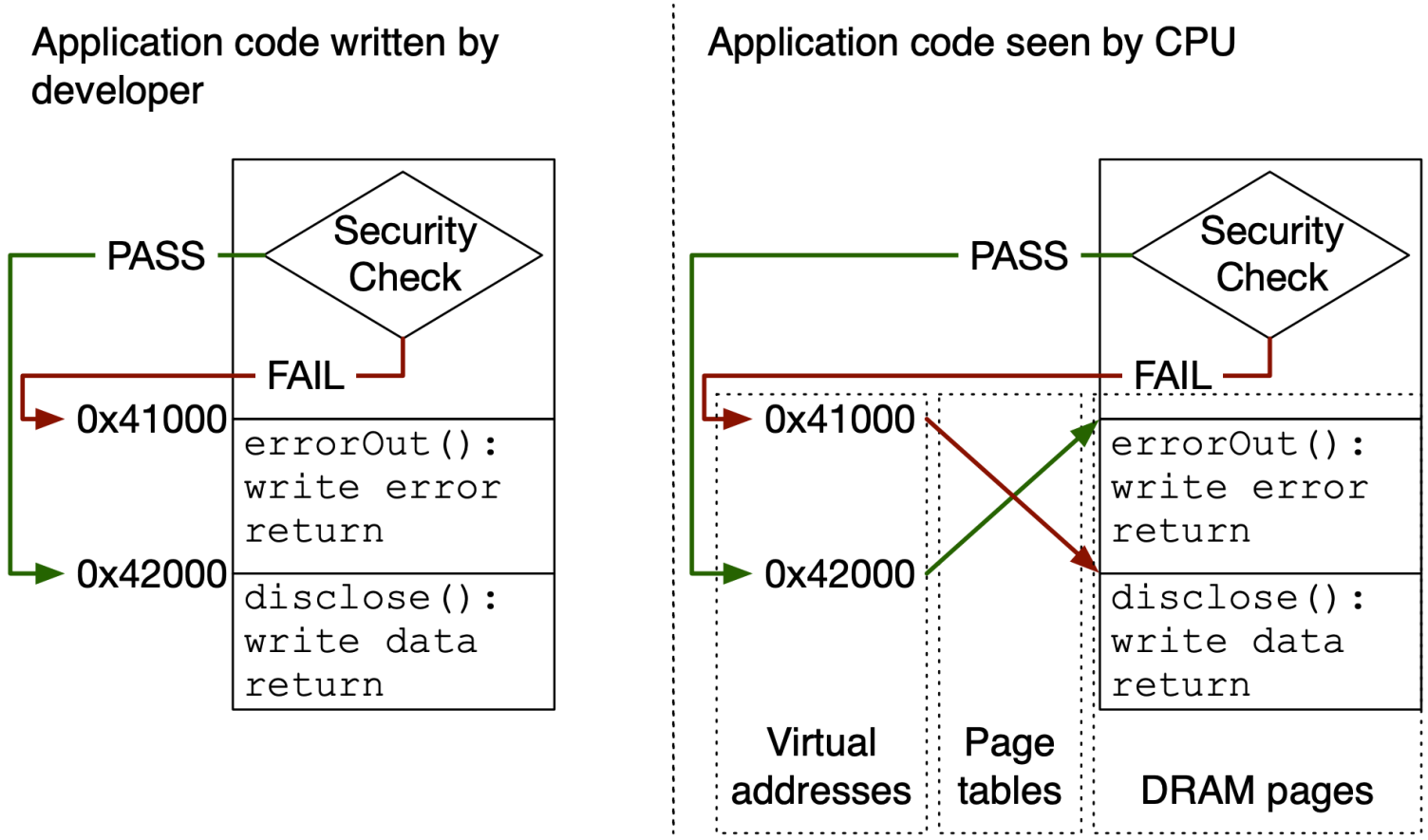
Policy 2:
If VA **NOT** belongs to enclave range,
Then PA must **NOT** belong to PRM

Malicious Translation #2



Policy 3:
If VA belongs to enclave range,
Then PA must belong to PRM
matching enclave ID

Malicious Translation #3



Solution: Inverted Page Table

- For each page in the PRM, remember the mapping from

  <PPN> → <VPN, Enclave ID>

  Keep the reversed page table in PRM, so privilege software cannot modify

- When to perform the check? (Review address translation process)
  - After each address translation

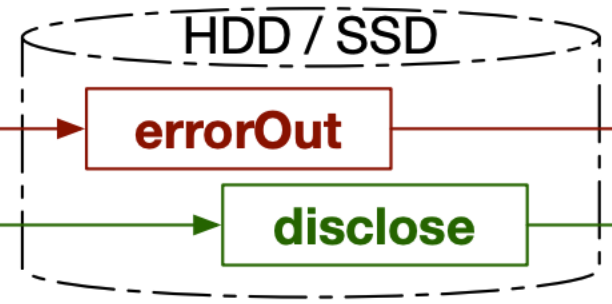
Malicious Translation #4

Page tables and DRAM before swapping

Virtual	Physical	Contents
0x41000	0x19000	errorOut
0x42000	0x1A000	disclose

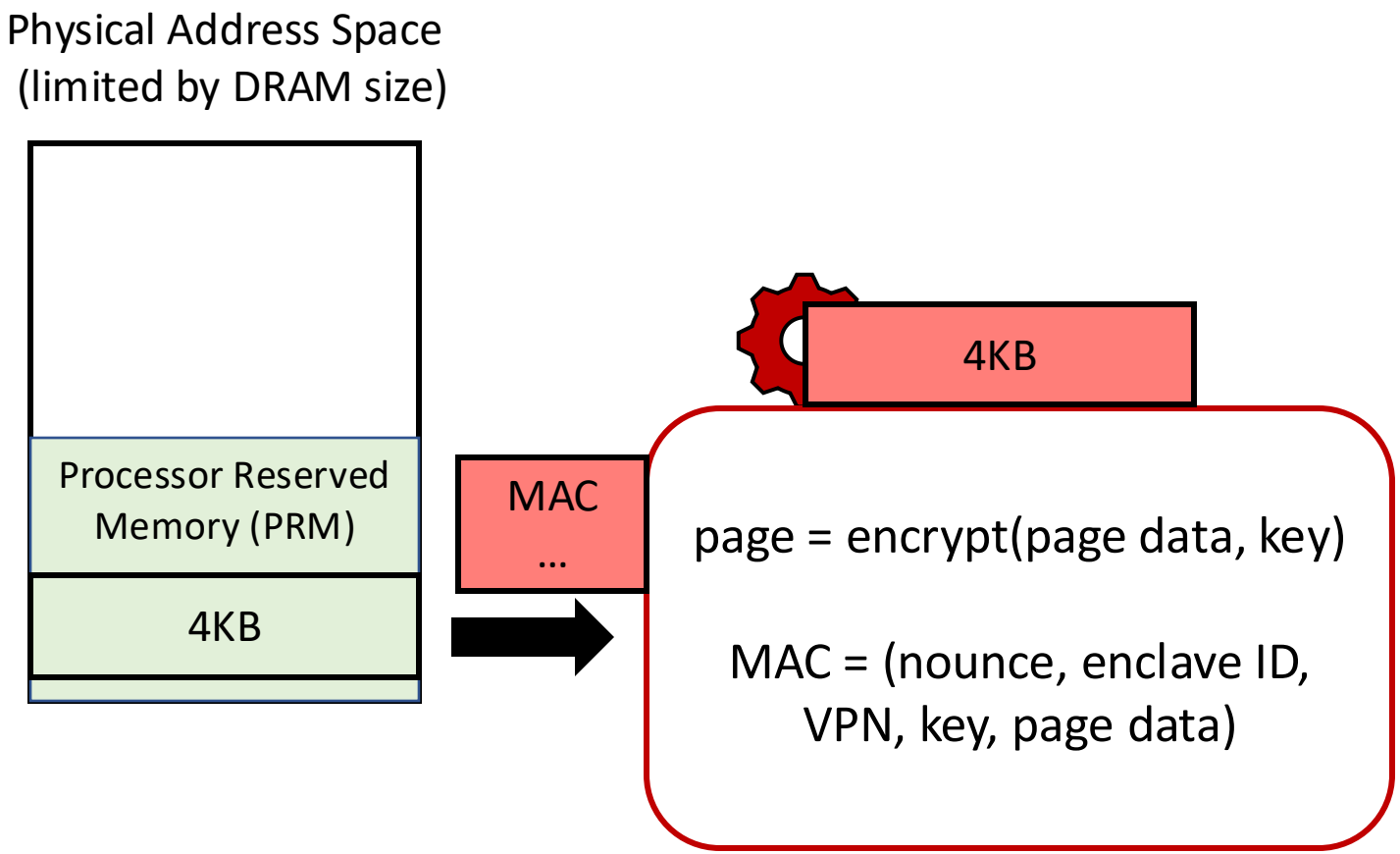
Page tables and DRAM after swapping

Virtual	Physical	Contents
0x41000	0x19000	disclose
0x42000	0x1A000	errorOut



Need to bind the virtual address mapping with the page content.

Solution: Page Encryption and Authentication



Summary: SGX Memory Management

- #1: Maintain a inverted page table and check after every address translation

  Physical page in PRM -> (enclave ID, virtual page number)

- #2: Encrypt/decrypt upon page swap to non-PRM region

  (nounce, enclave ID, virtual page number, key, page content) → MAC

- #3: Keep TLB state up-to-date

  Upon page swap, block the page in the inverted page table and unblock only until all the corresponding TLB entries are flushed

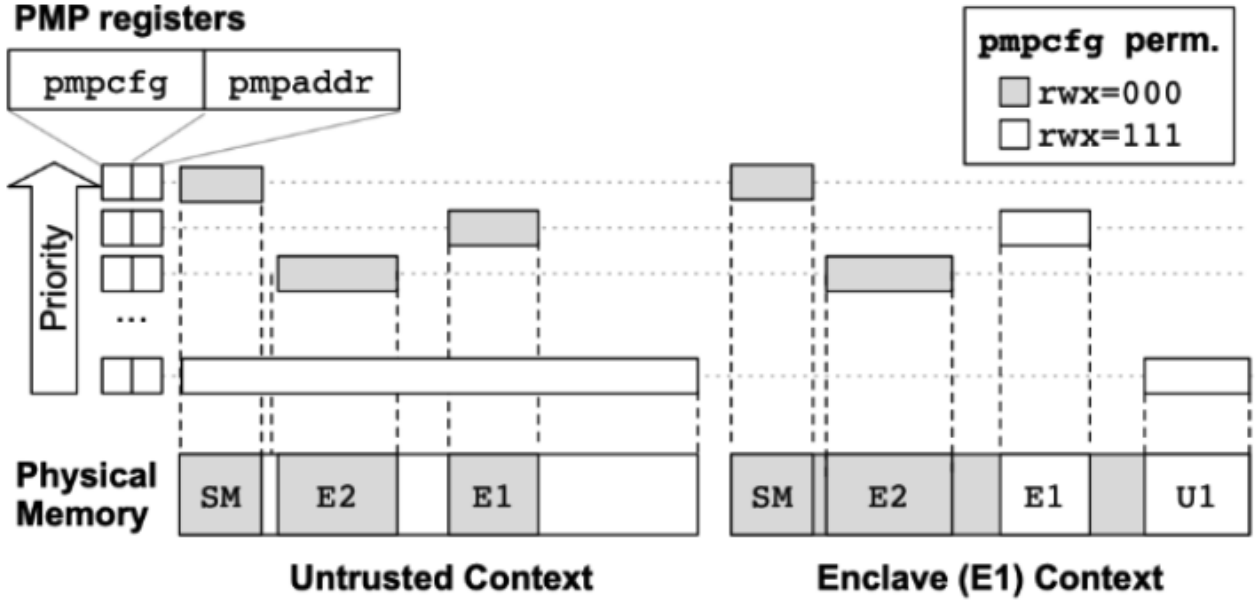
  (We did not discuss this item in detail. Feel free to refer to the reference materials)

Alternative Solutions

- Naïve idea:
  - Let the trusted component handle page management
  - Problem: Large TCB
- Keystone's approach: leverage RISC-V's PMP registers
  - Enforce coarse-grained isolation and let the application to manage their page mappings
- AMD SEV:
  - Rely on encryption. But symmetric key vulnerability
  - Recent version introduces reverse page table

Keystone's Solution

- PMP check after every page translation to avoid cross-domain access
- Enclave application uses a runtime to support self-managed mapping.



Future Trends and Research Challenges

- Extend TEE to heterogenous systems
  - GPUs
  - Accelerators

- Abundant side channels

Summary

- What can privilege software attackers do?

- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc

- Read more:
  - Intel SGX Explained; by Costan et al
  - SoK: Understanding Designs Choices and Pitfalls of Trusted Execution Environments; by Li et al