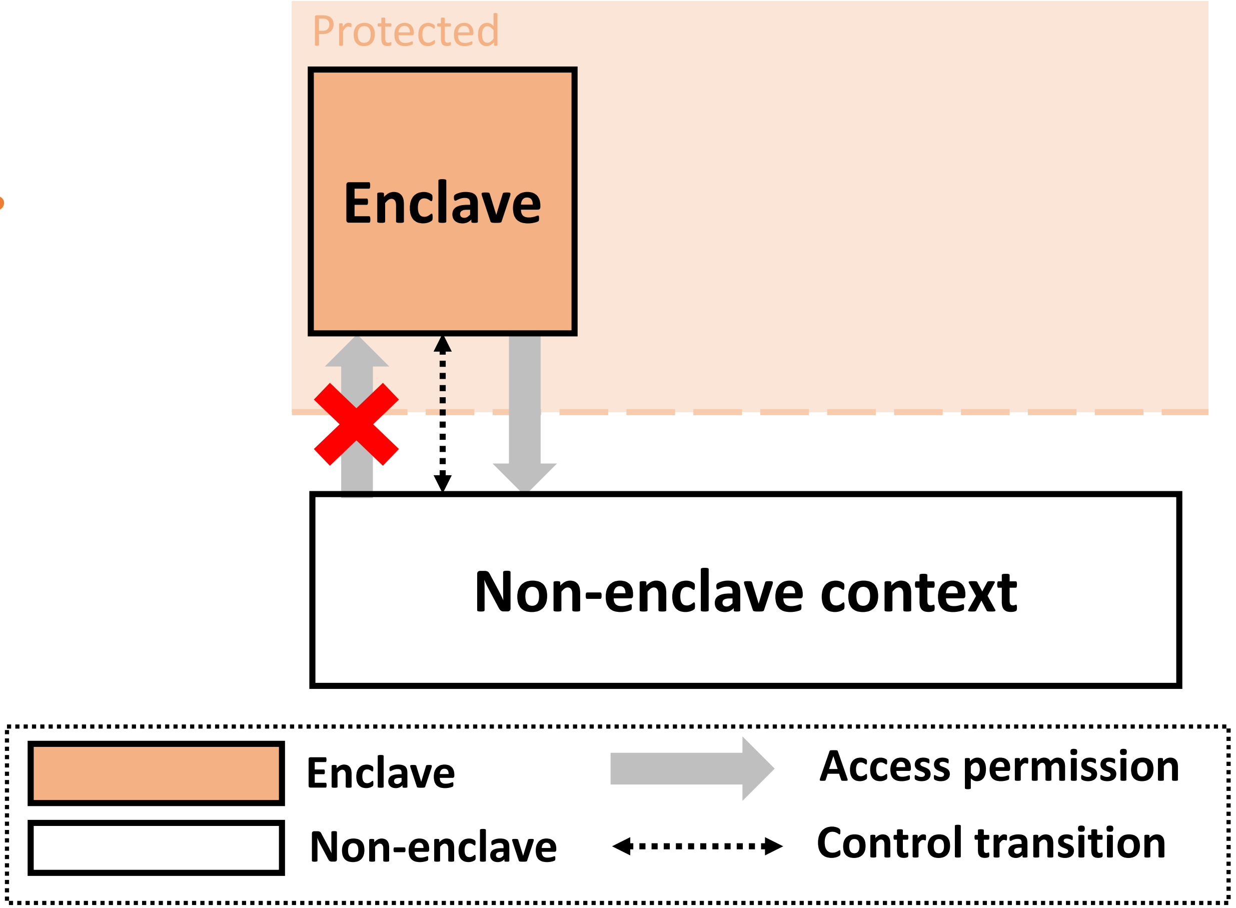
Nested Enclave: Supporting Fine-grained Hierarchical Isolation with SGX

Joongun Park, **Naegyong Kang, Taehoon Kim,**
Youngjin Kwon, Jaehyuk Huh

School of Computing,
KAIST

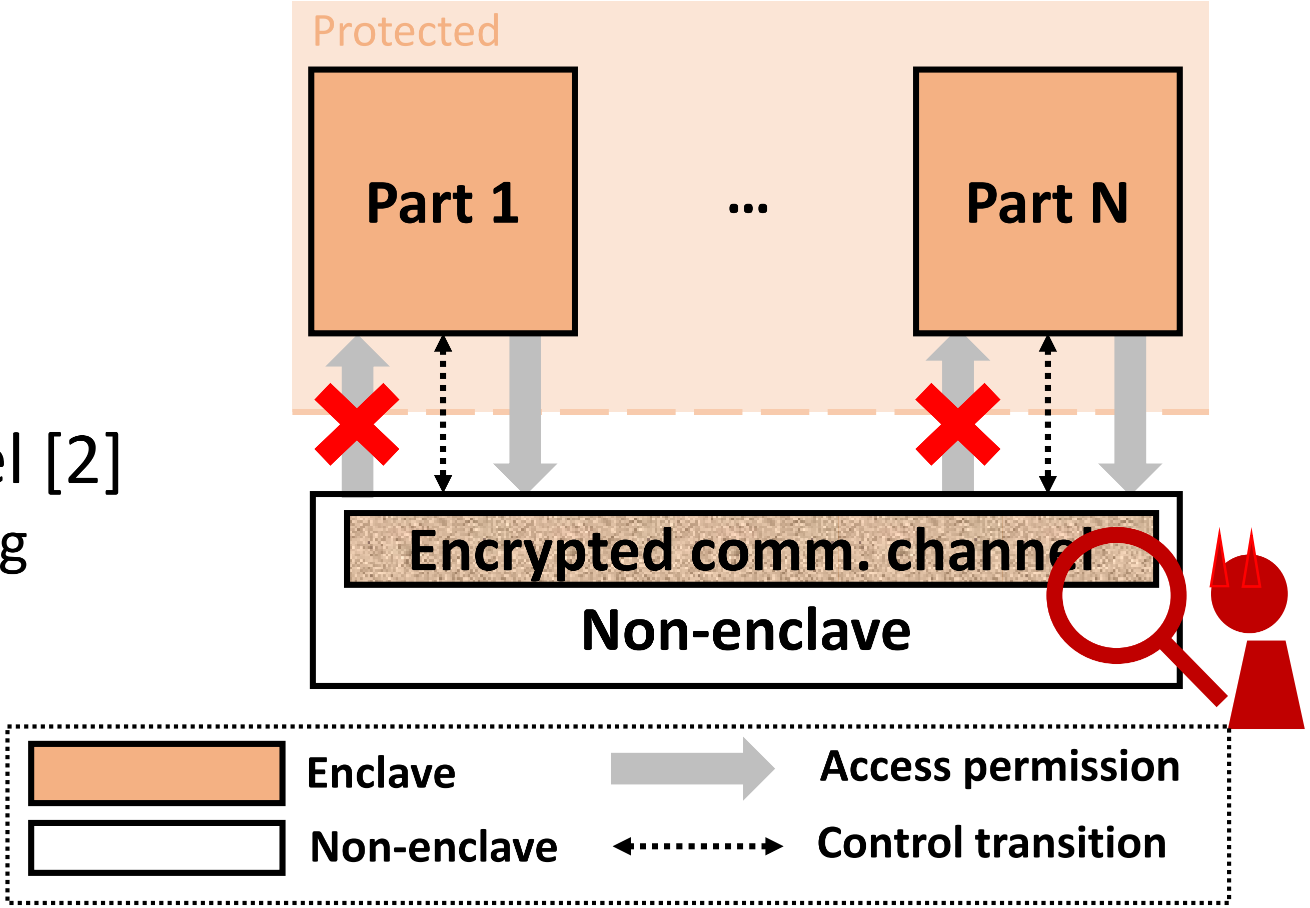
SGX Trusted Execution Environment

- Intel SGX
  - Provides trusted execution environments (TEE) called *Enclave*
  - Protected from malicious privileged SW
  - Guarantee confidentiality and integrity
  - **Monolithic design**



Need for Extension

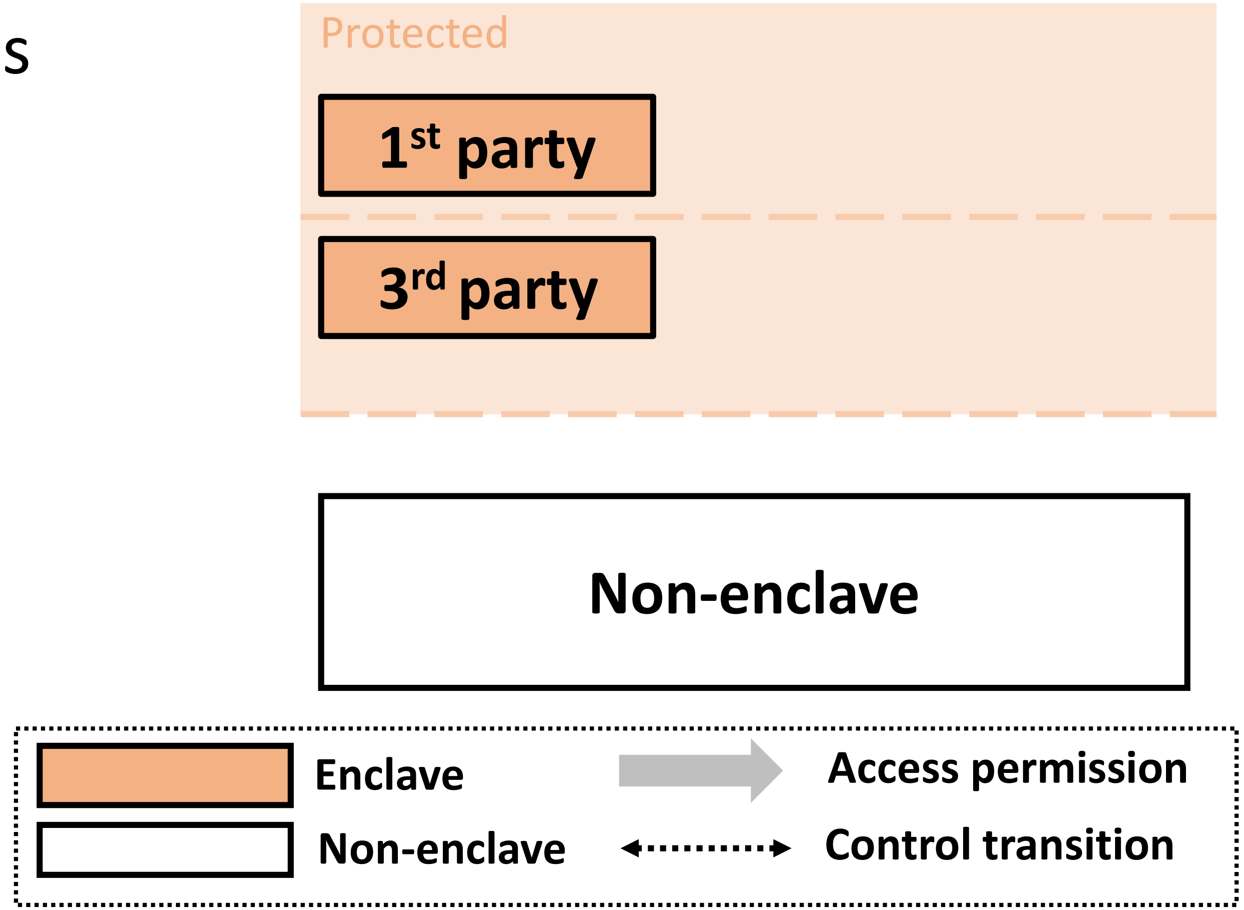
- Mutually distrustful parties [1]
  - Multiple parties are involved for building an application
  - Use multiple enclaves
- Exposed communication channel [2]
  - Must protect against eavesdropping
  - Must detect silent drops



[1] Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data [2] PANOPLY: Low-TCB Linux Applications with SGX Enclaves

Multi-level Security [3]

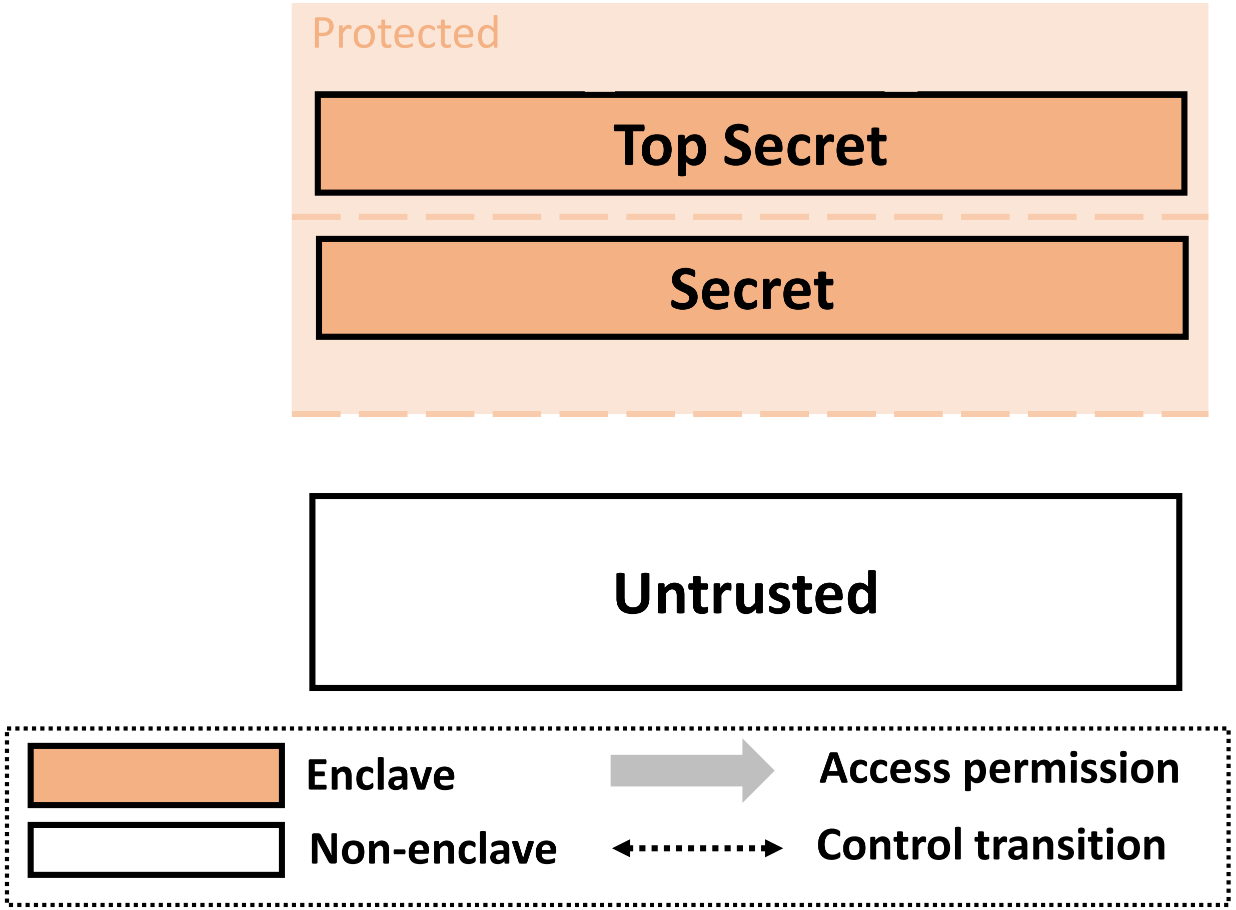
- Applications need multiple levels of security
  - Top secret / Secret
  - 1st party app / 3rd party library



[3] Secure Computer Systems: Mathematical Foundations

Compartmentalization

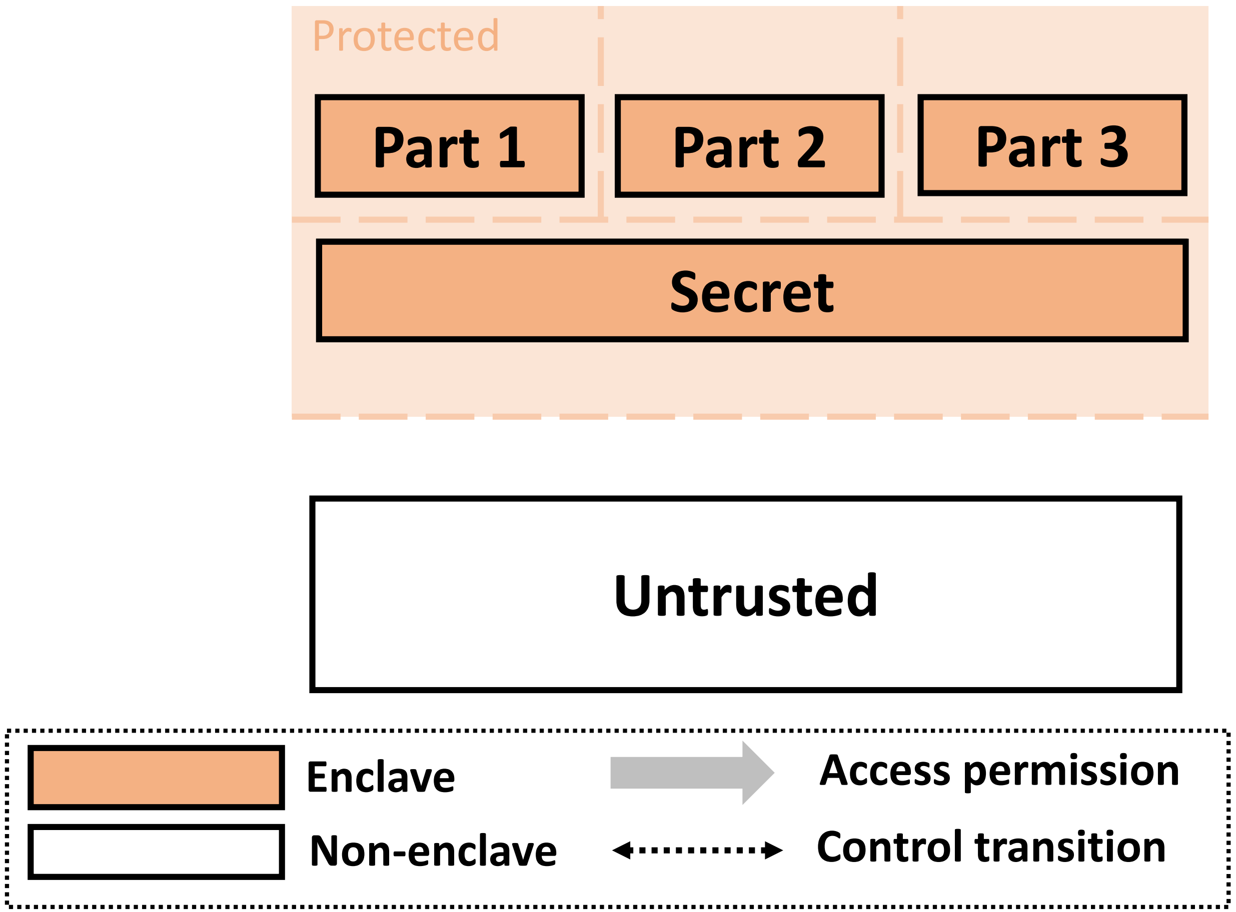
- Compartmentalization
  - Isolated peer compartments



KAIST School of Computing

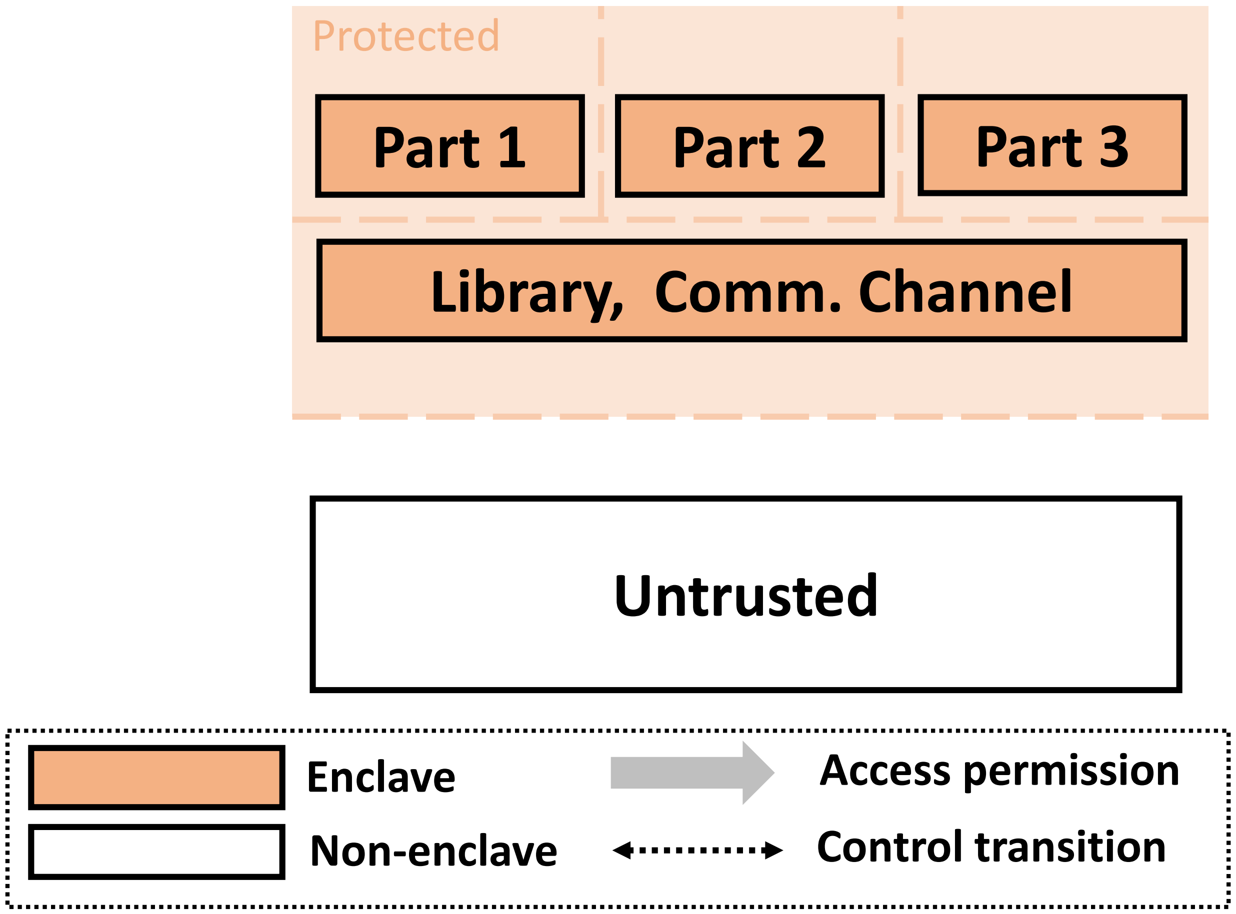
Compartmentalization

- Compartmentalization
  - Isolated peer compartments



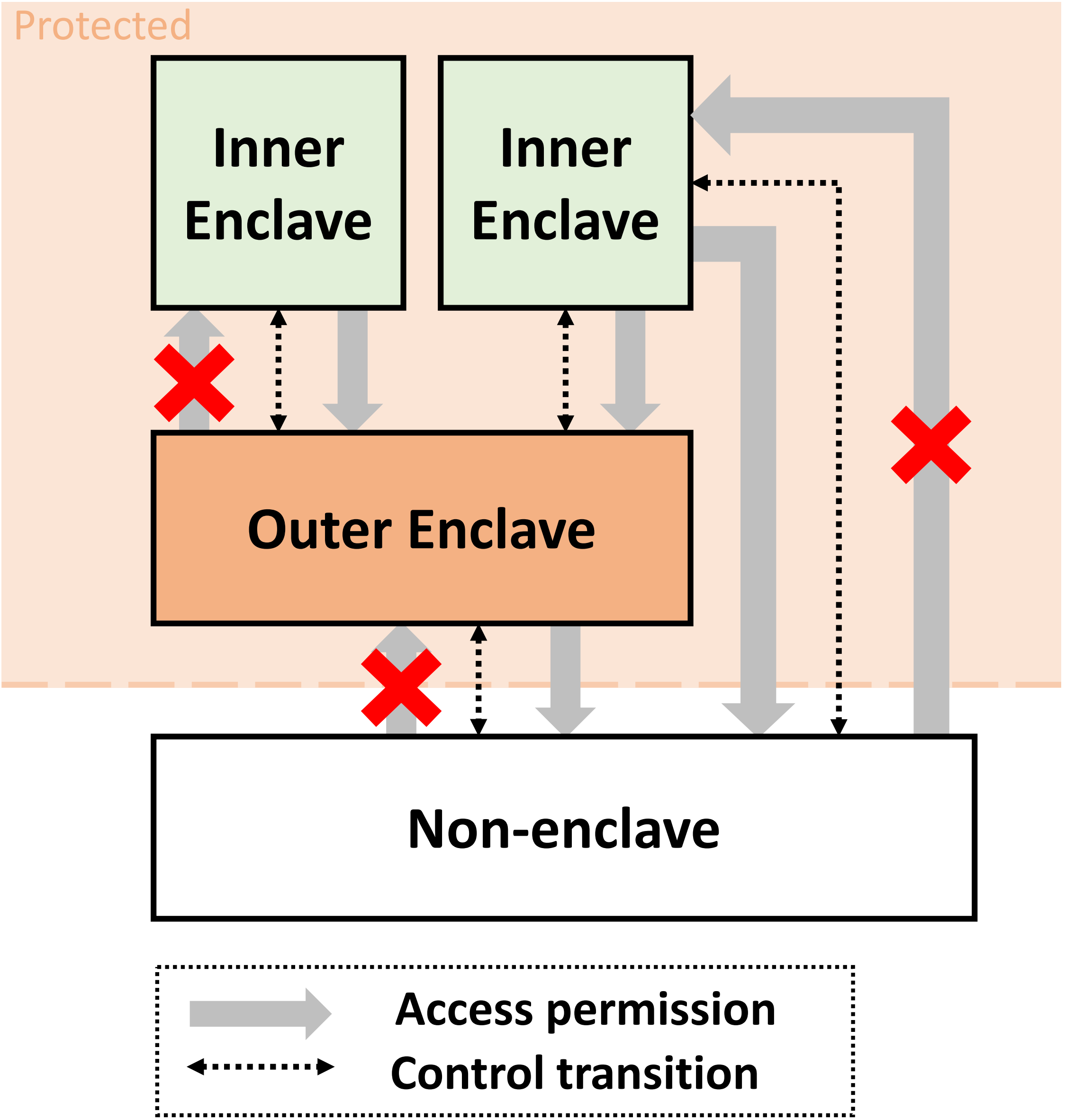
Compartmentalization

- Compartmentalization
  - Isolated peer compartments
- Sharing lower compartment
  - Shared library
  - Communication channel



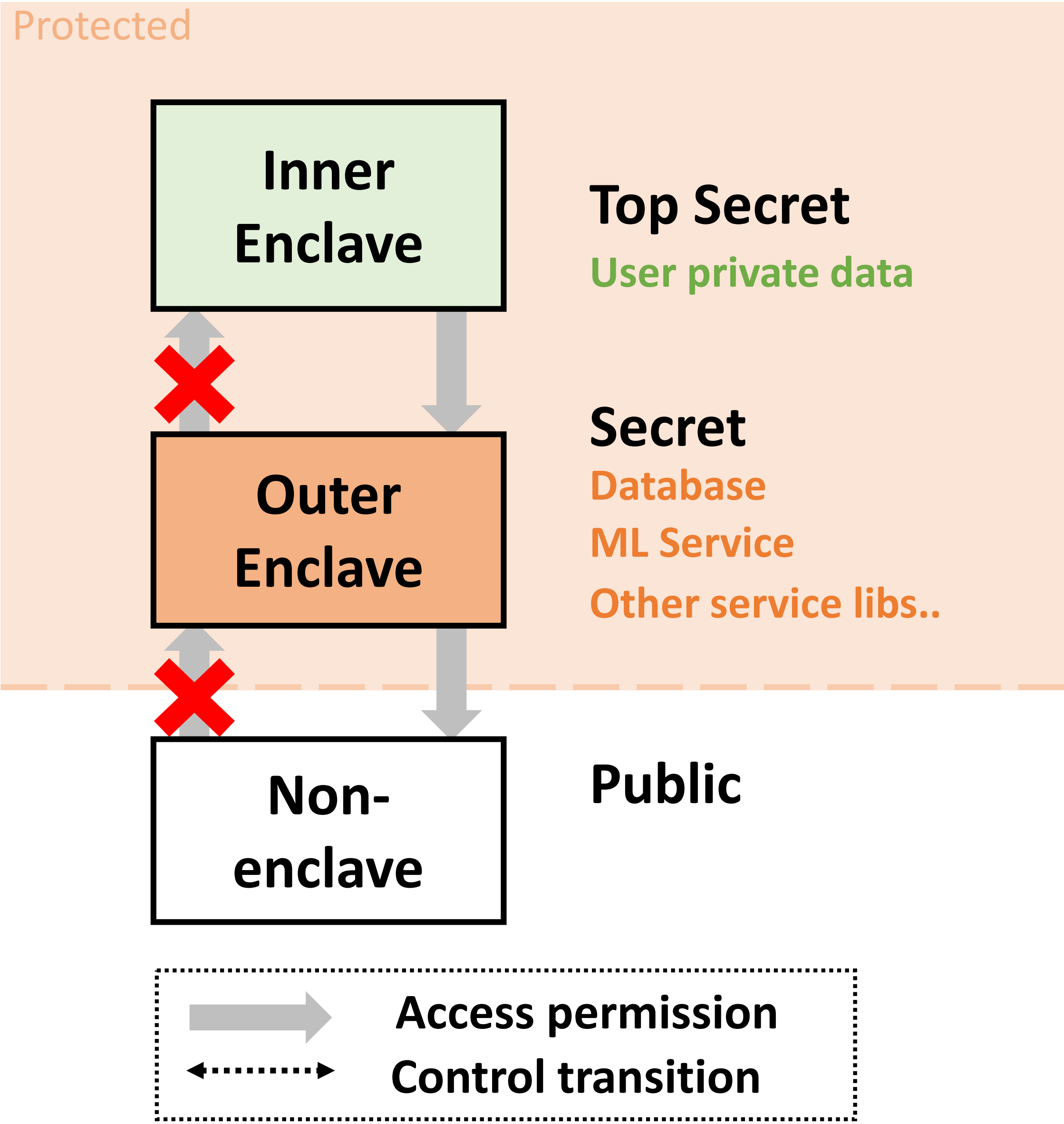
Nested Enclave

- Nested enclave is hardware extension to SGX
  - Inner Enclave
  - Outer Enclave



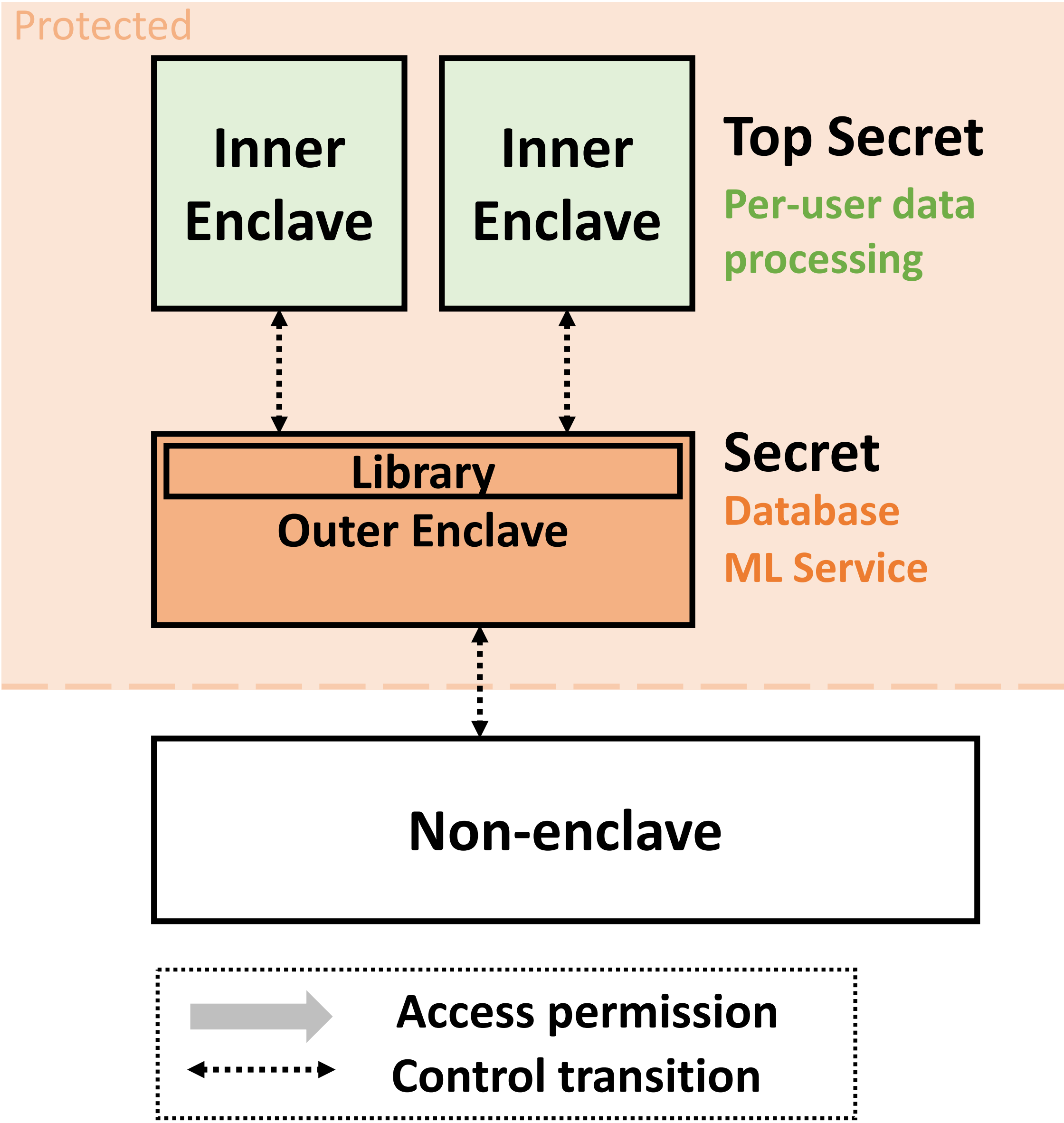
New semantics

- Hierarchical Isolation
  - Non-enclave context doesn't have access permission to both enclaves
  - Outer enclave doesn't have access permission to inner enclaves
  - Inner enclave has access permission to lower levels
- Supporting multi-level security



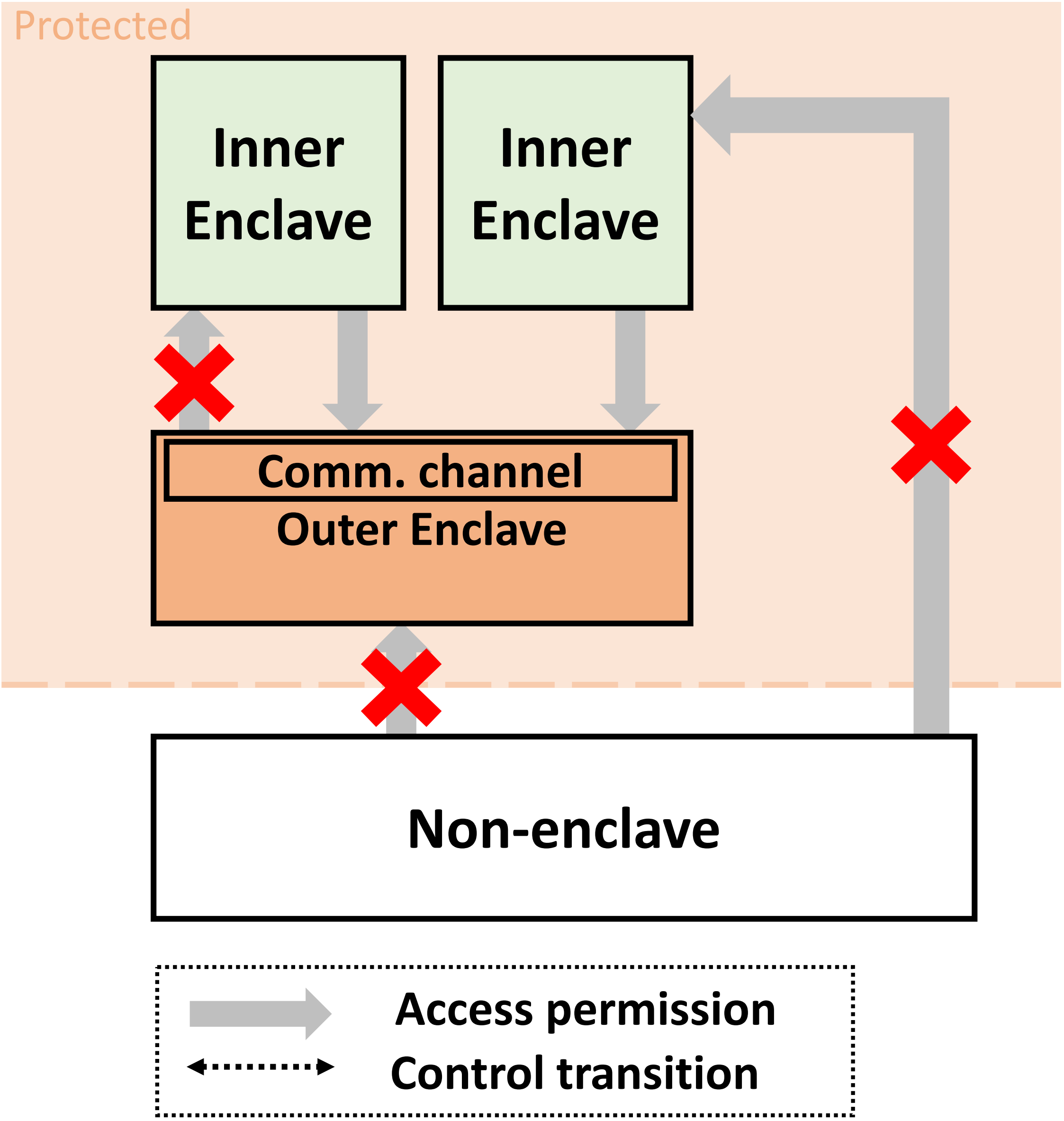
New semantics (Cont'd)

- Compartmentalization
  - Isolation among Inner Enclaves
- Shared library with Outer Enclave
  - Reduced total memory usage
  - Inner enclave is protected from shared library



New semantics (Cont'd)

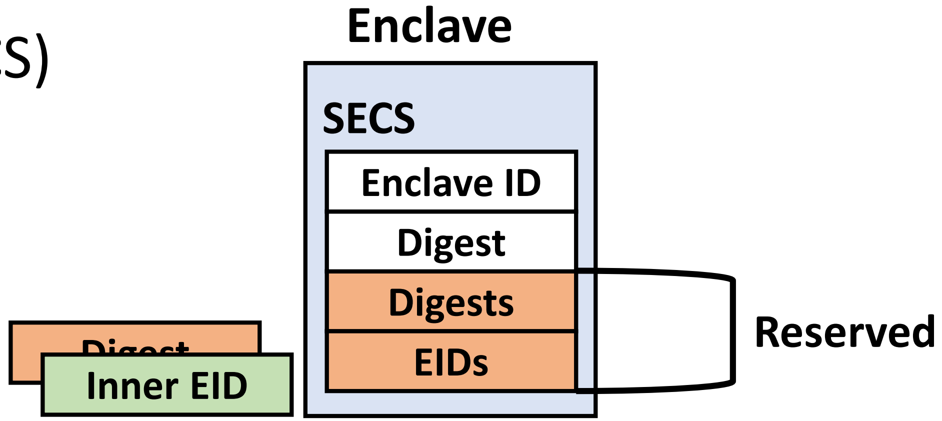
- Secure communication channel in outer enclave
  - Hardware based protection via Memory Encryption Engine (MEE)
- Faster communication through caches
  - No encryption for data in caches
  - Enables faster data transfer [4]



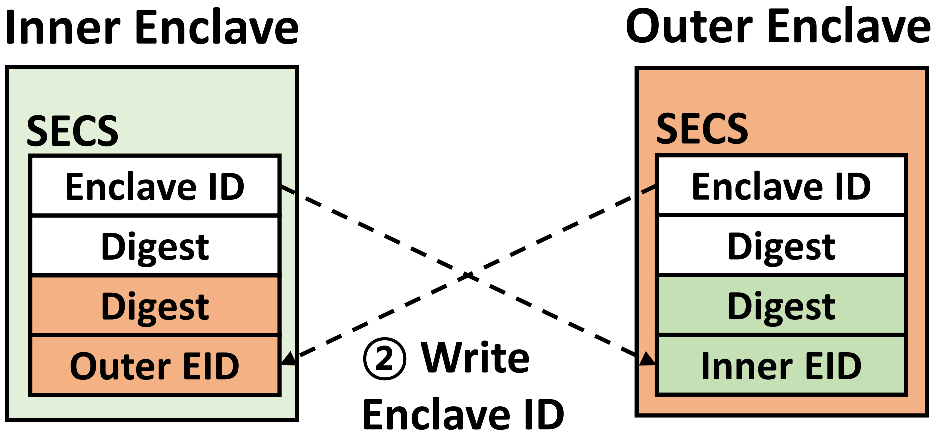
[4] The multikernel: a new OS architecture for scalable multicore systems

States and Association

- SGX Enclave Control Structure (SECS)
  - The state of an enclave
  - Use reserved fields to contain
    - Inner or outer enclave's digest
    - Inner or outer enclave's ID (EID)



- Association of inner-outer enclaves
  - Attest the pair enclave with digest
  - Write verified Enclave ID (EID) in SECS

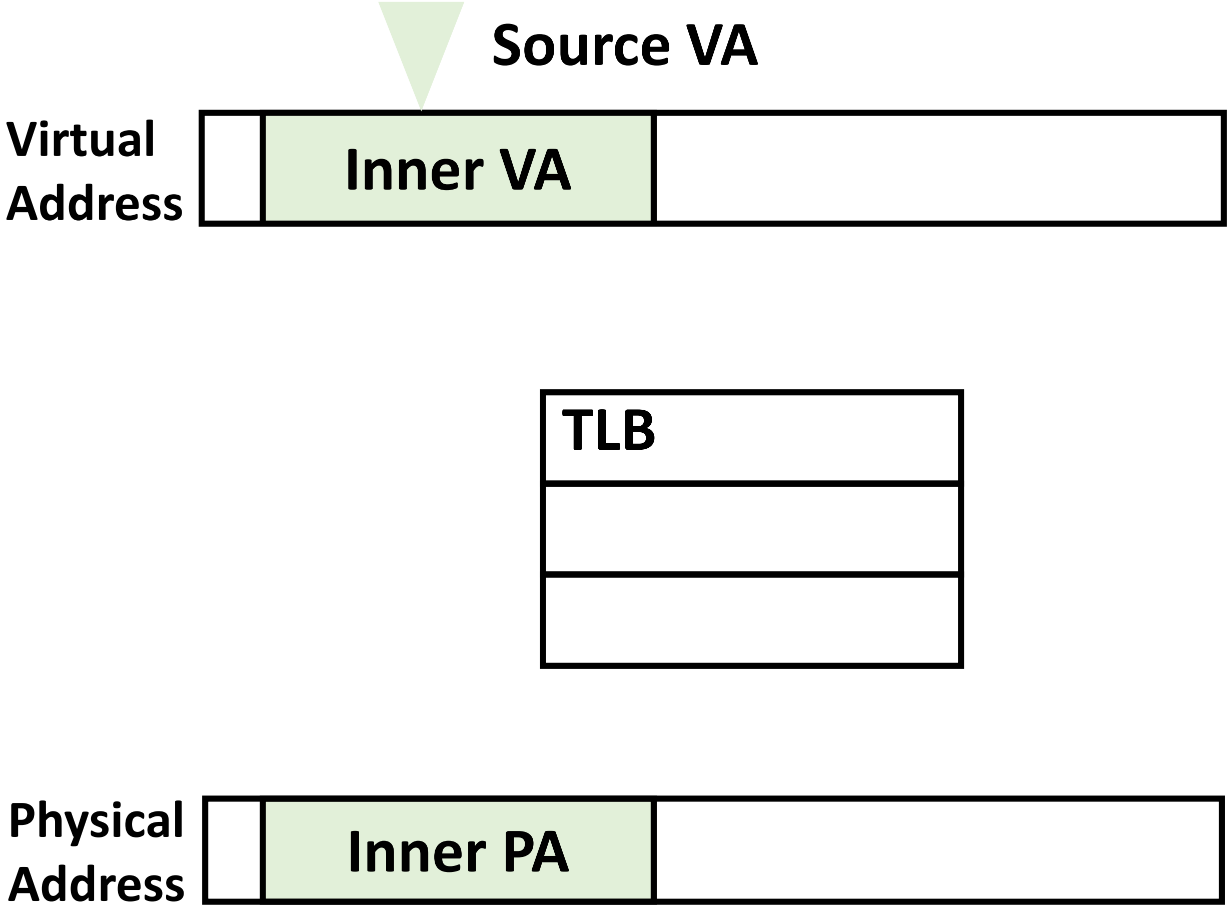


Memory Access Validation

- SGX validates memory access during TLB miss handling
- *EPCM (Enclave Page Cache Map)* in SGX:
  - Meta data for each physical frame
  - Owner EID, VPN (virtual page number), etc

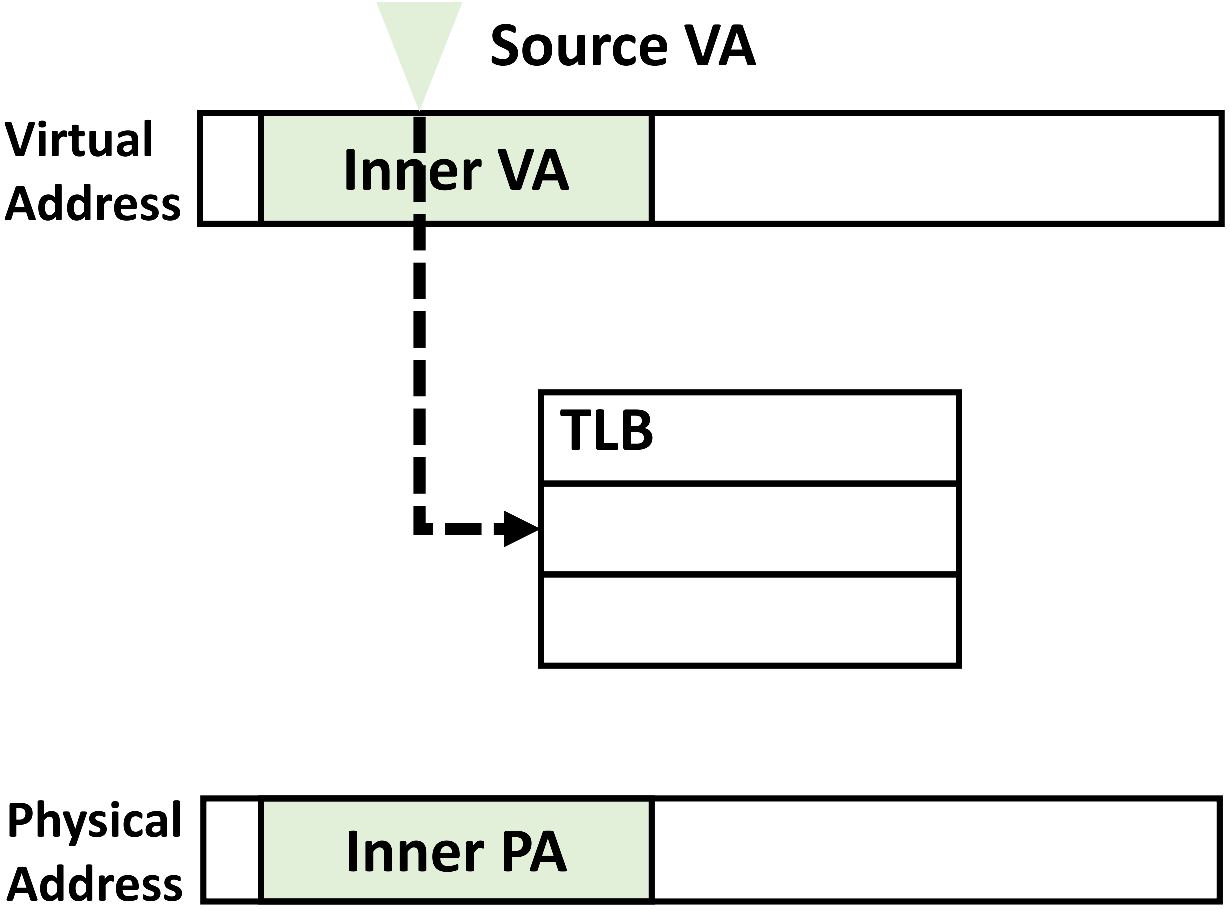
- Invariant for security: TLB must contain only validated translation
- Cases
  - (A) Inner enclave accesses its enclave region
  - (B) Inner enclave accesses its outer enclave region
  - Others

Access Validation (A)



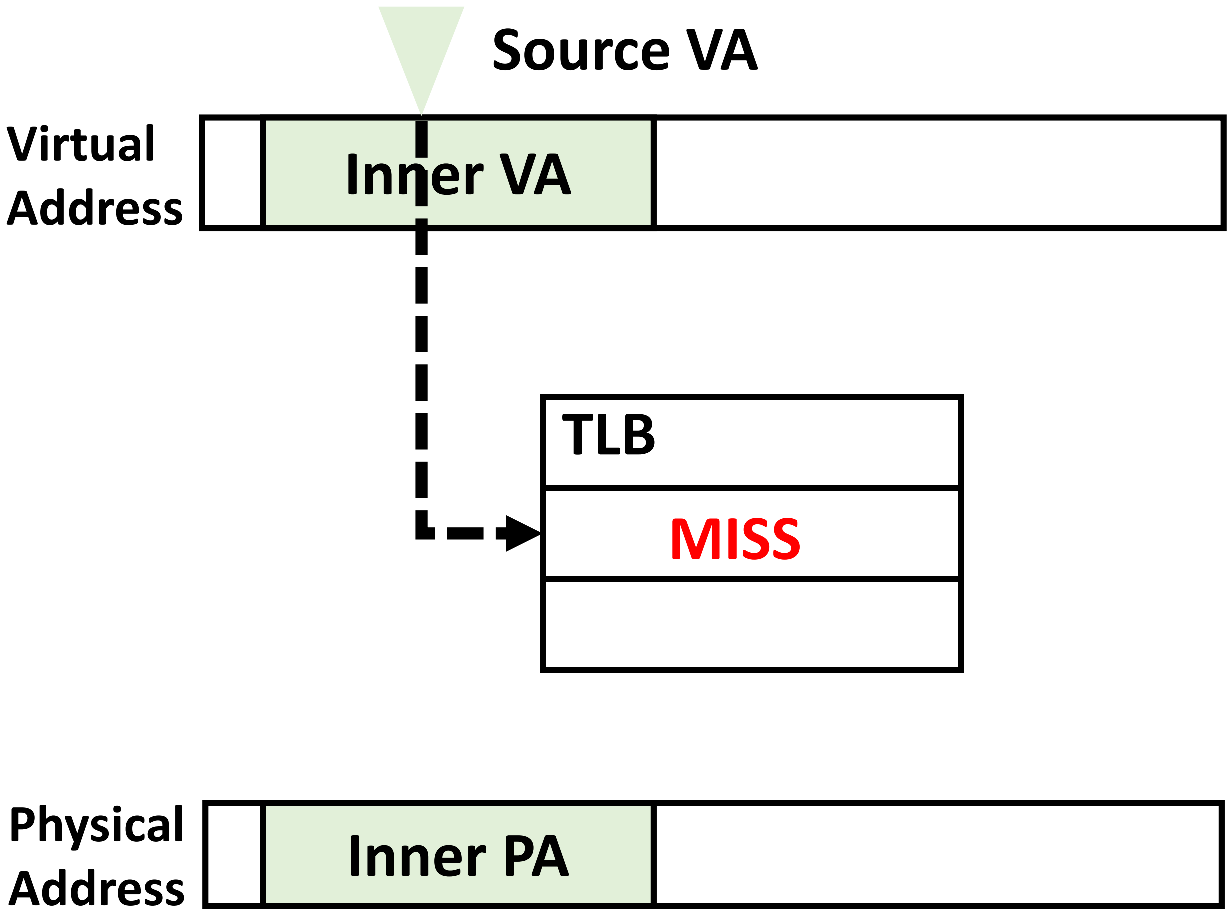
(A) Inner enclave accesses its enclave region

Access Validation (A)



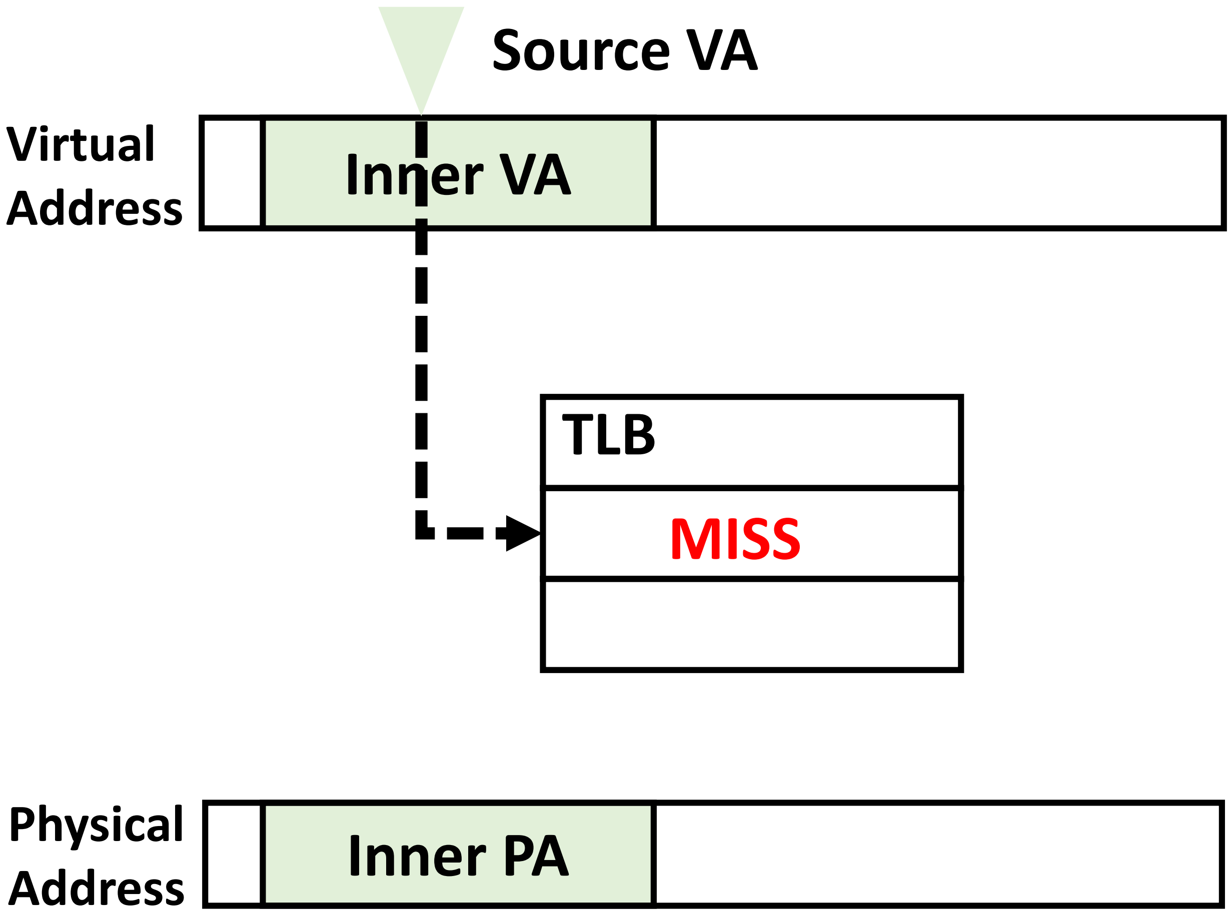
(A) Inner enclave accesses its enclave region

Access Validation (A)



(A) Inner enclave accesses its enclave region

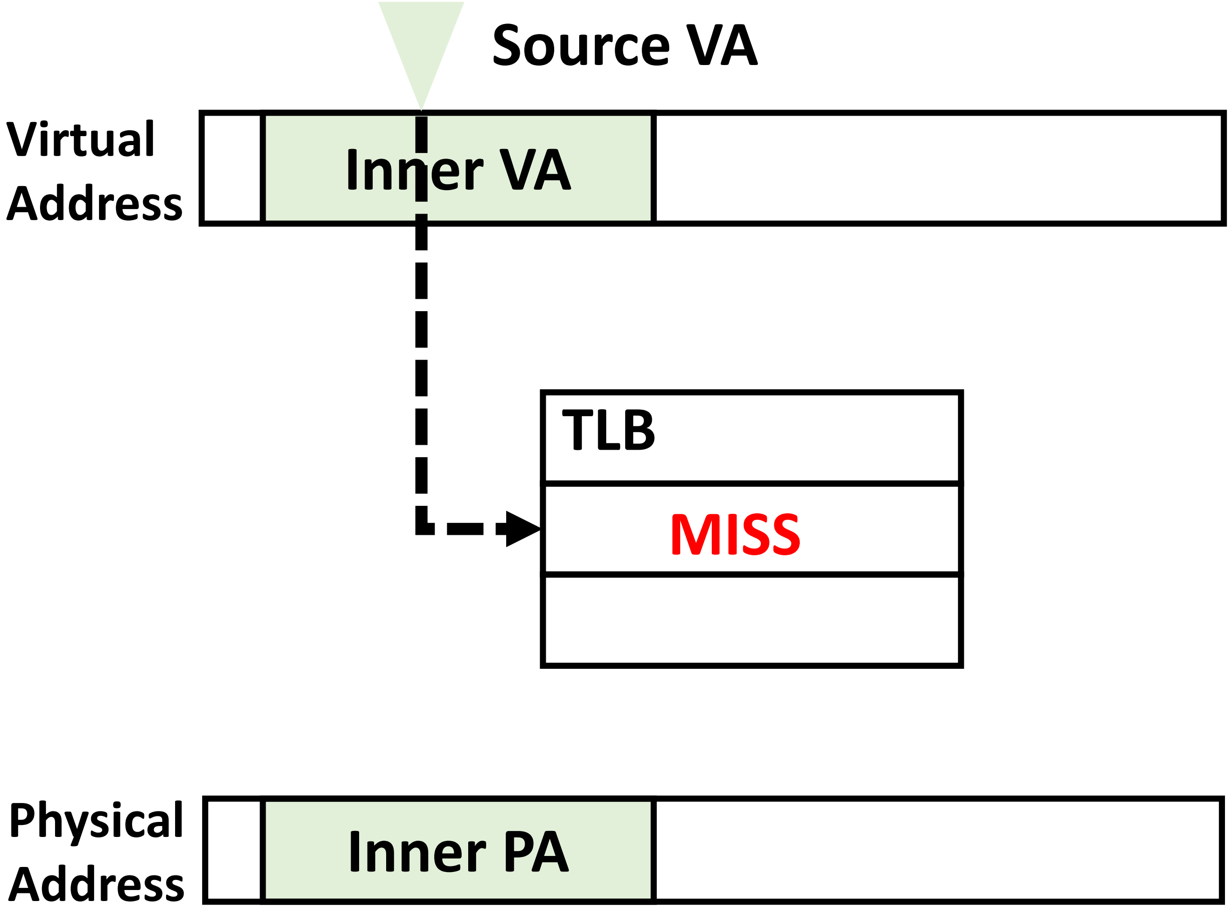
Access Validation (A)



(A) Inner enclave accesses its enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID) || (Owner EID == Outer EID)

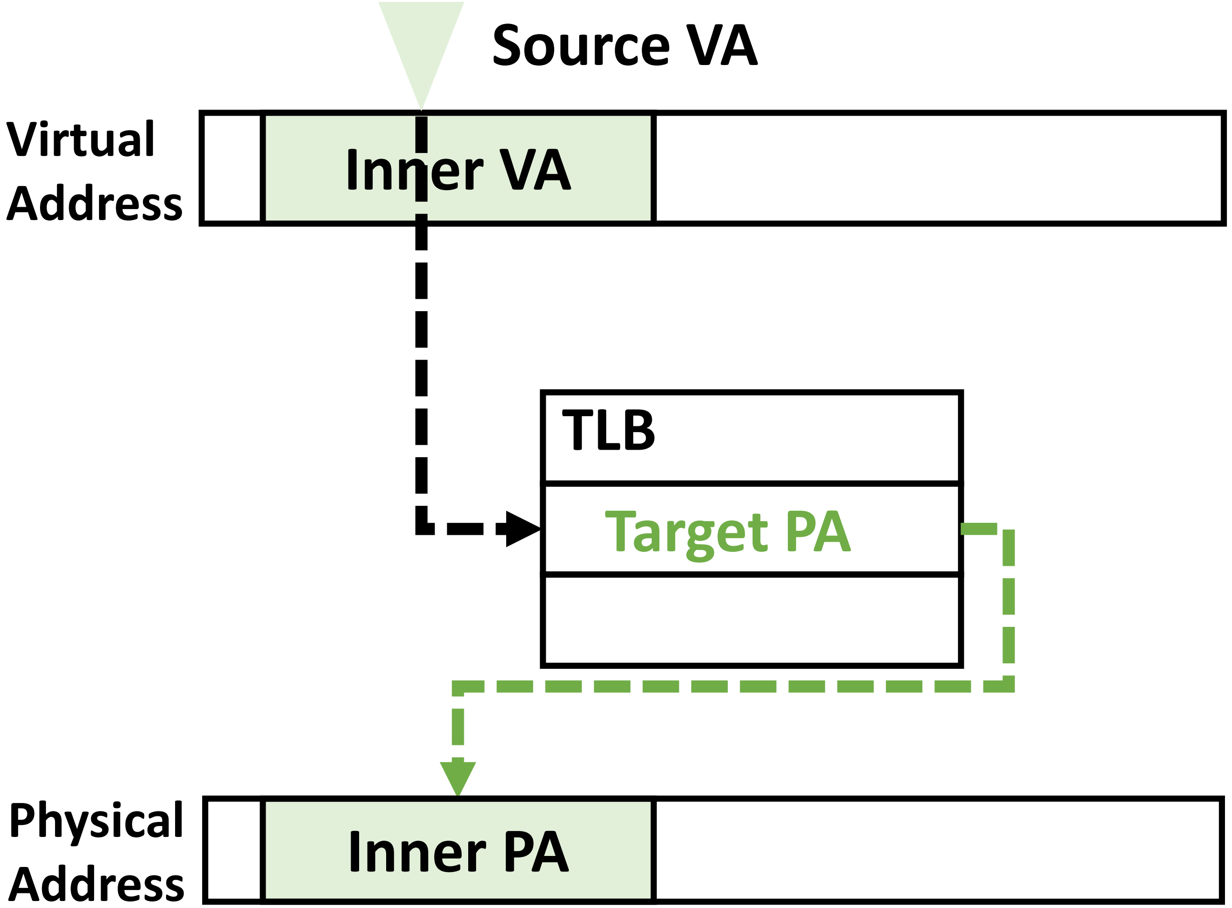
Access Validation (A)



(A) Inner enclave accesses its enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == Outer EID)

Access Validation (A)

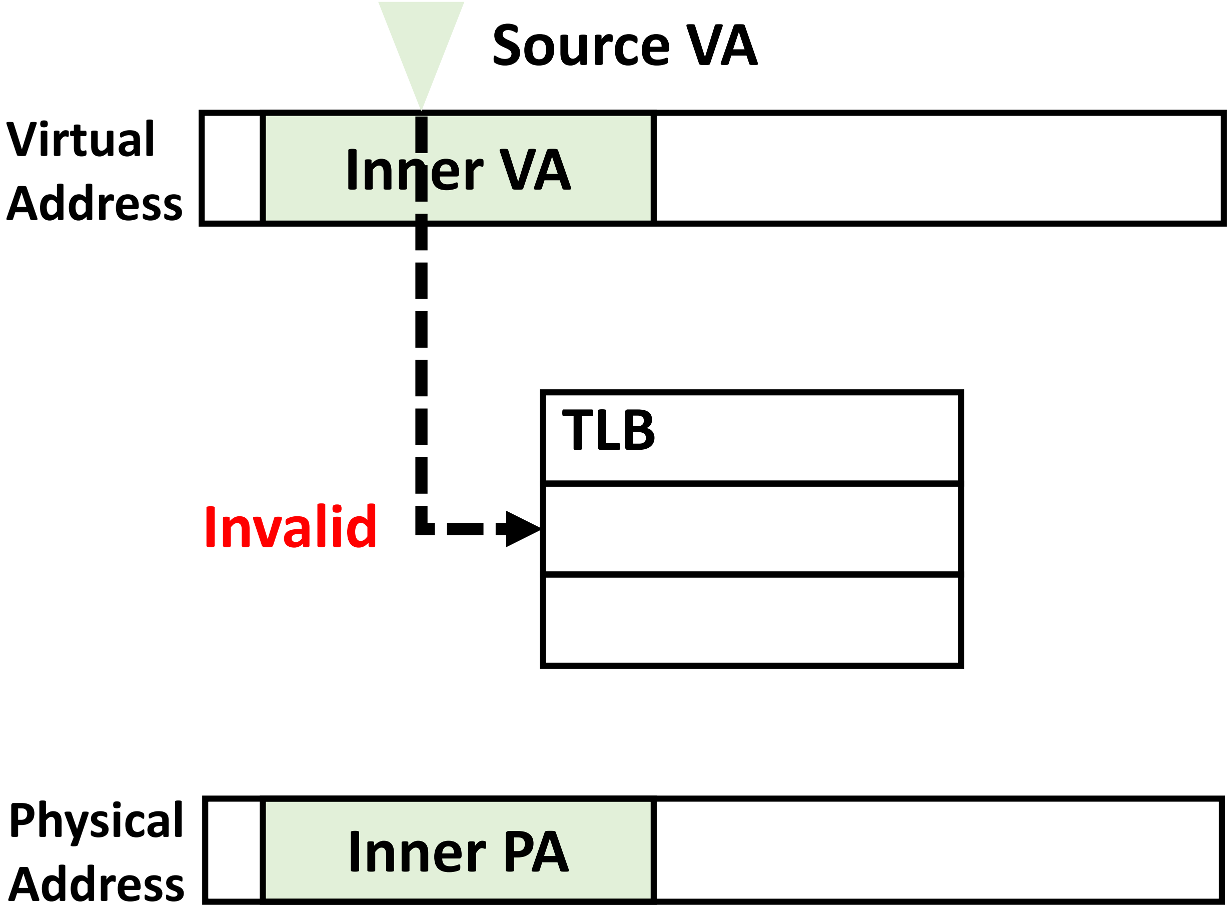


(A) Inner enclave accesses its enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- **Correct VPN?**
- **(Owner EID == current EID)**
  || (Owner EID == Outer EID)

YES => Insert TLB entry

Access Validation (A)



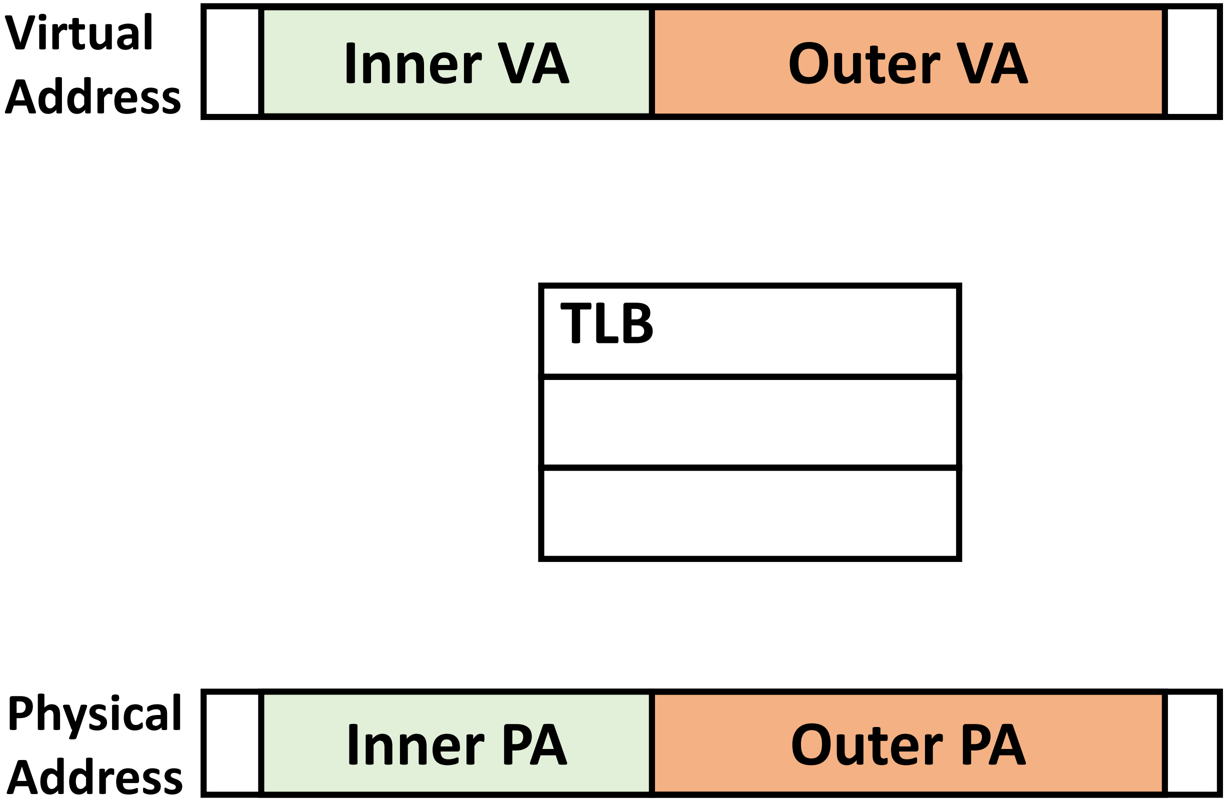
(A) Inner enclave accesses its enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == Outer EID)

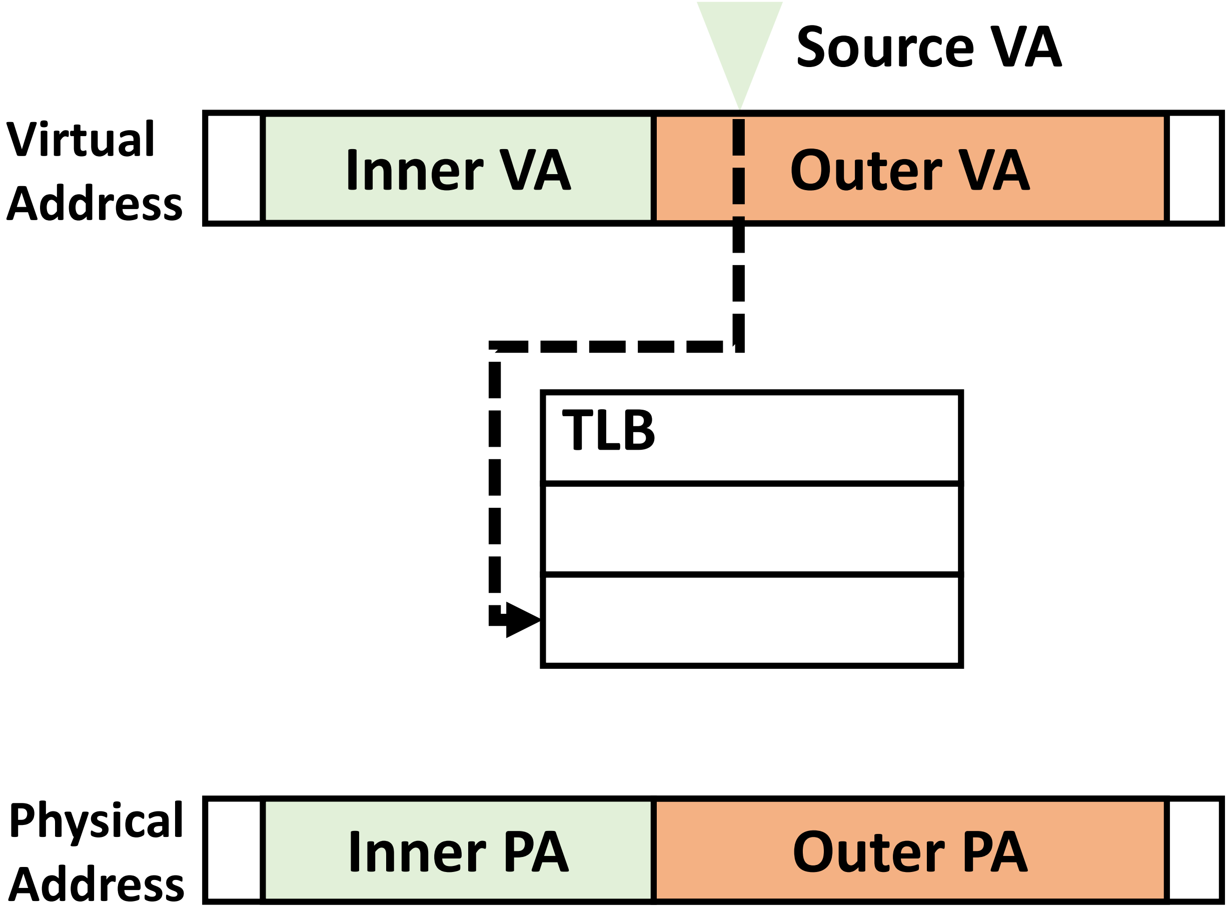
YES => Insert TLB entry
No => Invalid

Access Validation (B)

(B) Inner enclave accesses its outer enclave region



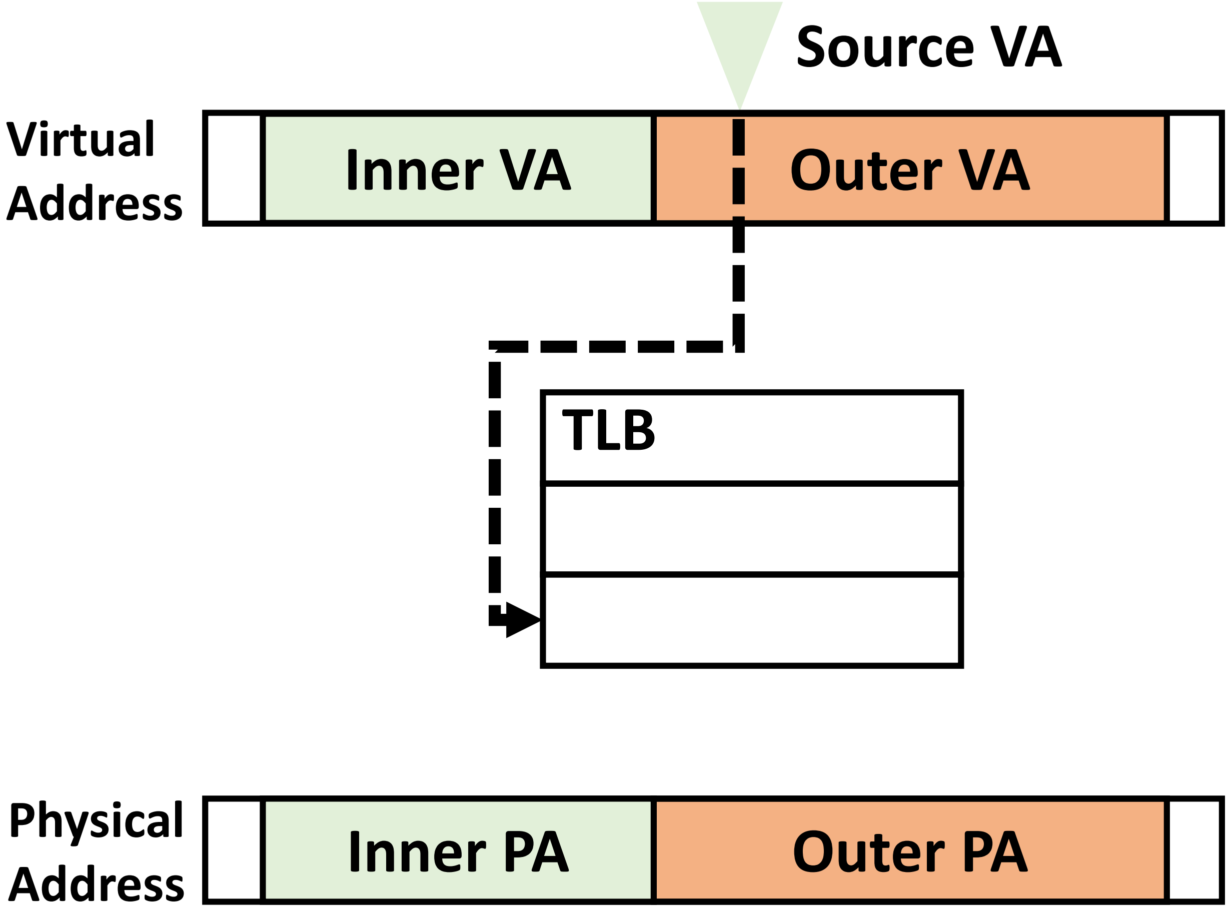
Access Validation (B)



(B) Inner enclave accesses its outer enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == *Outer EID*)

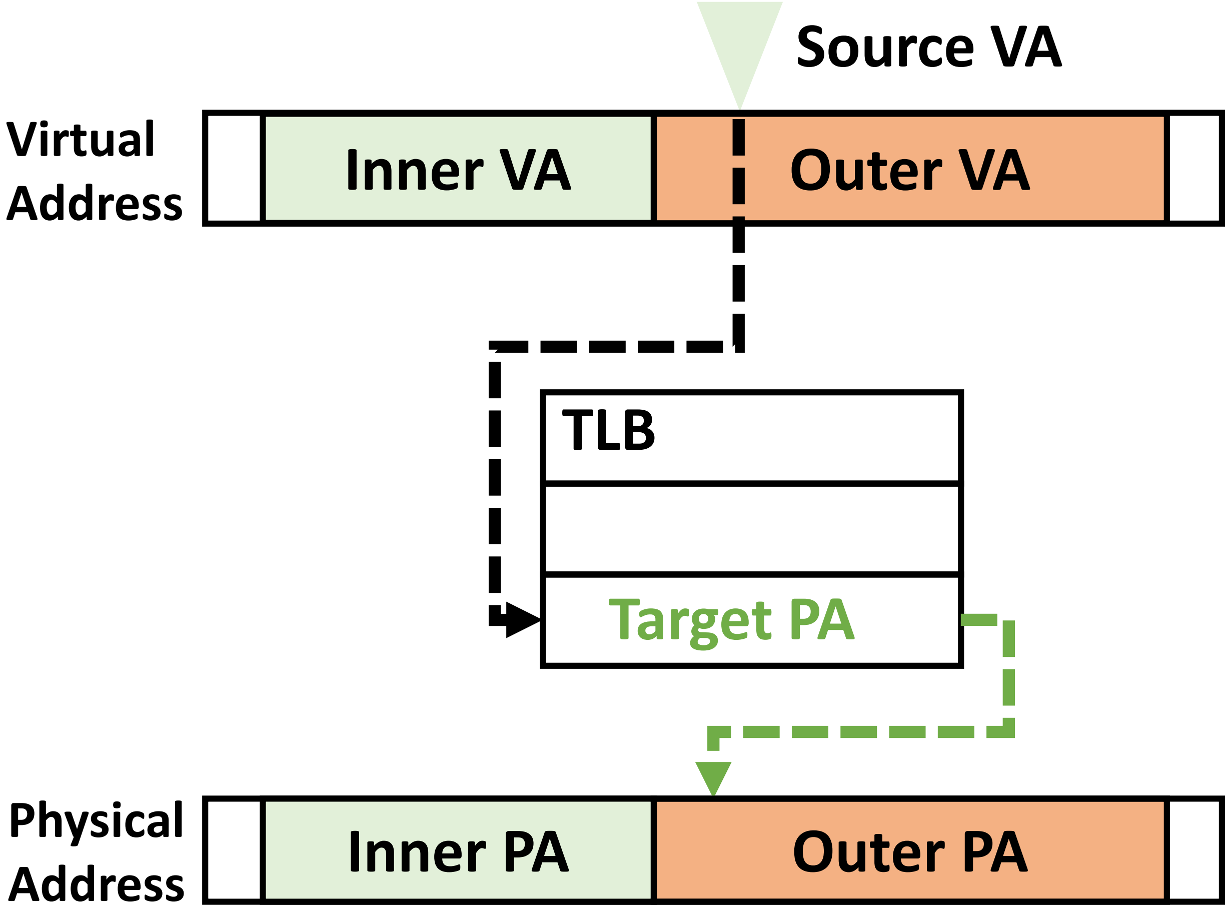
Access Validation (B)



(B) Inner enclave accesses its outer enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == *Outer EID*)

Access Validation (B)

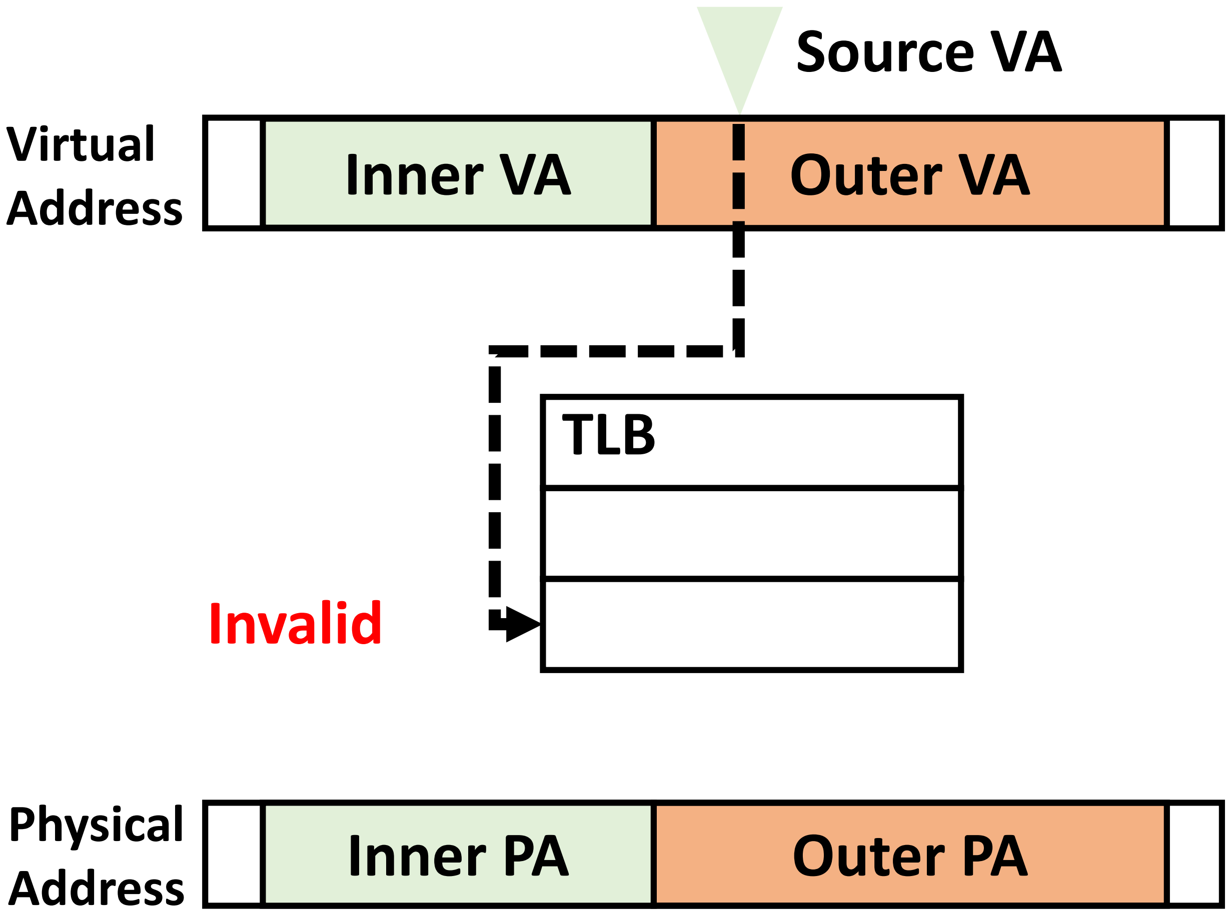


(B) Inner enclave accesses its outer enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == *Outer EID*)

YES => Insert TLB entry

Access Validation (B)

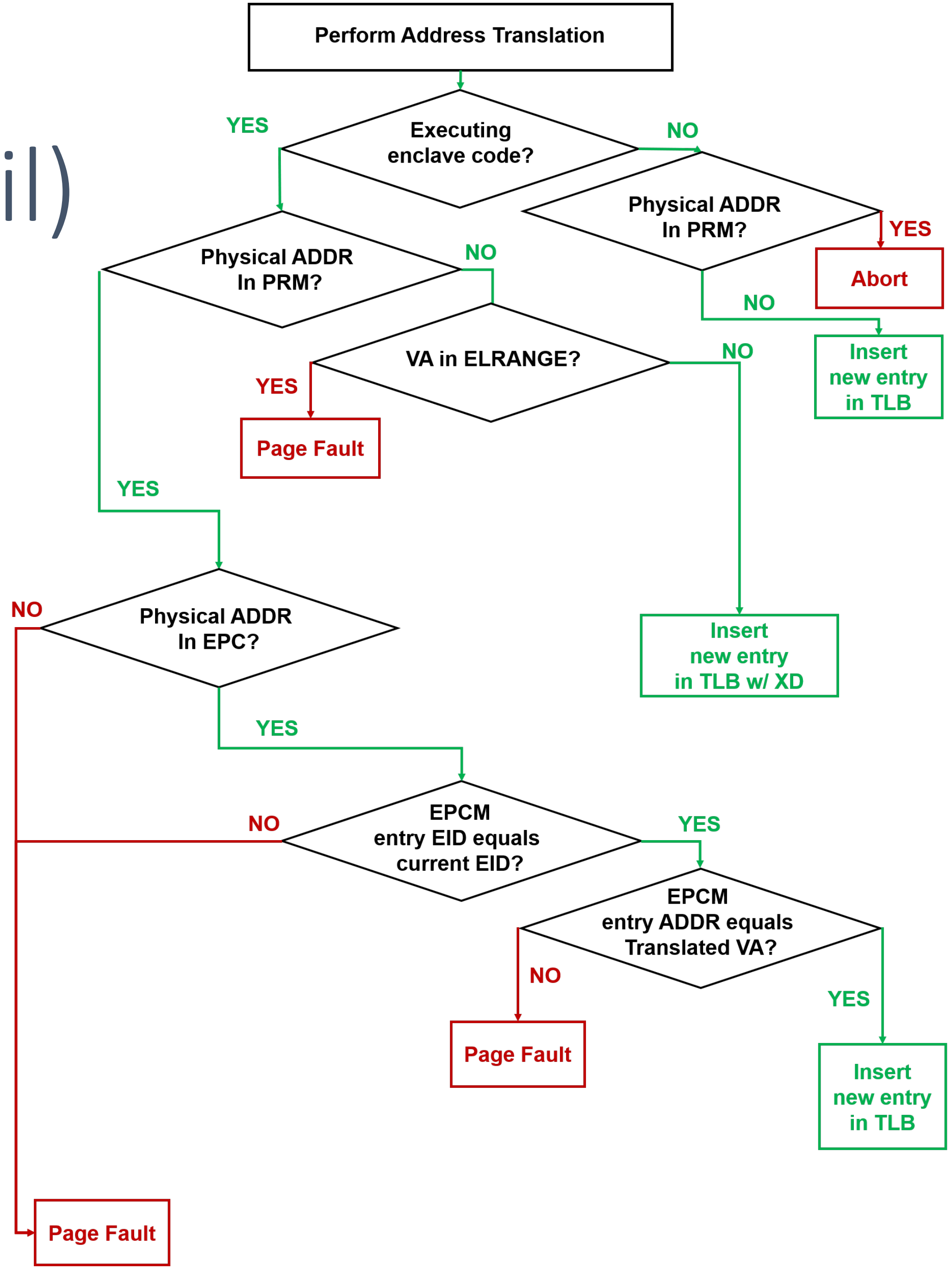


(B) Inner enclave accesses its outer enclave region

(1) Target PA is Enclave PA?
(2) Check EPCM
- Correct VPN?
- (Owner EID == current EID)
|| (Owner EID == *Outer EID*)

YES => Insert TLB entry
No => Invalid

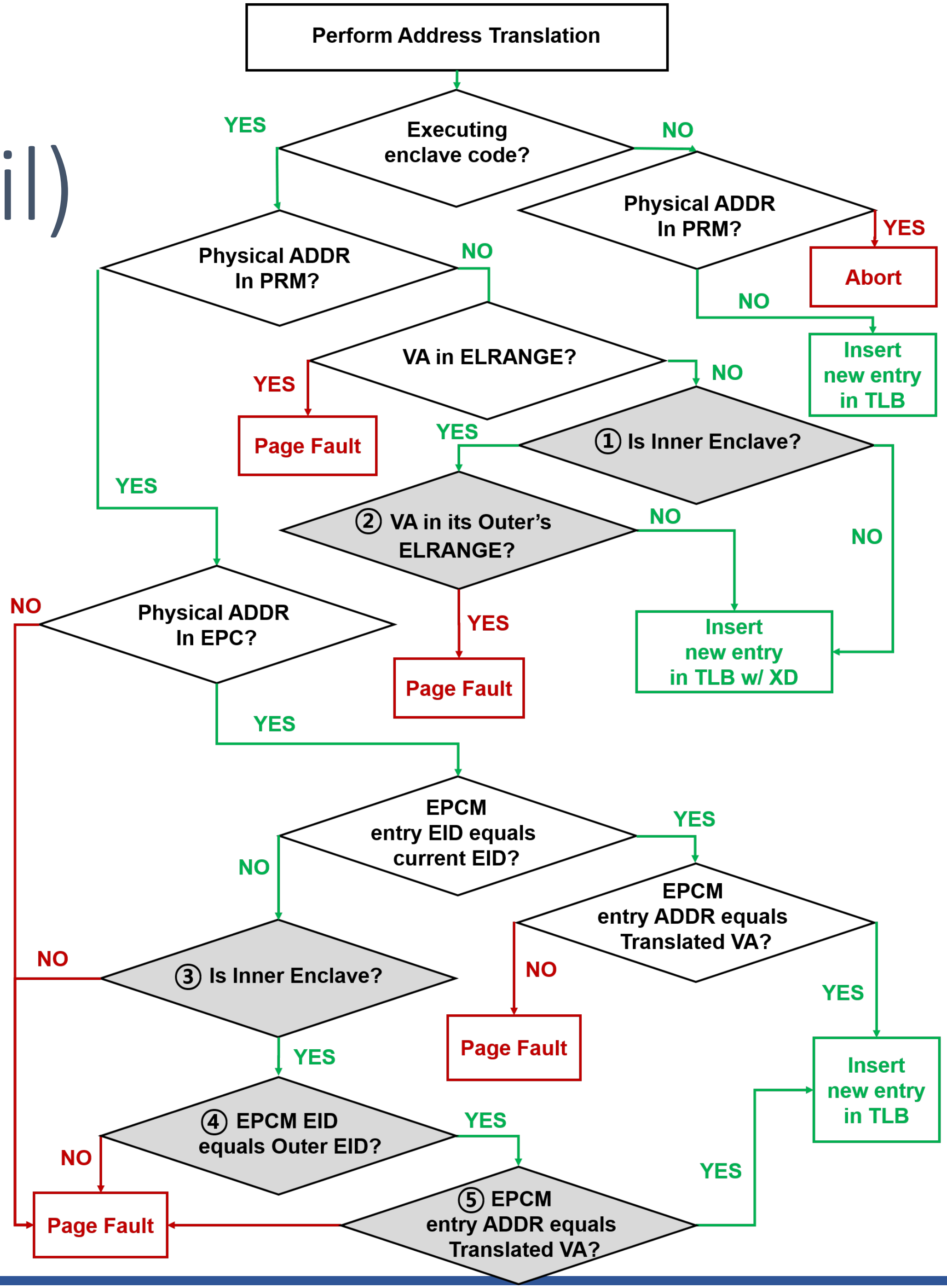
Access Validation (detail)



[4] Intel SGX Explained.

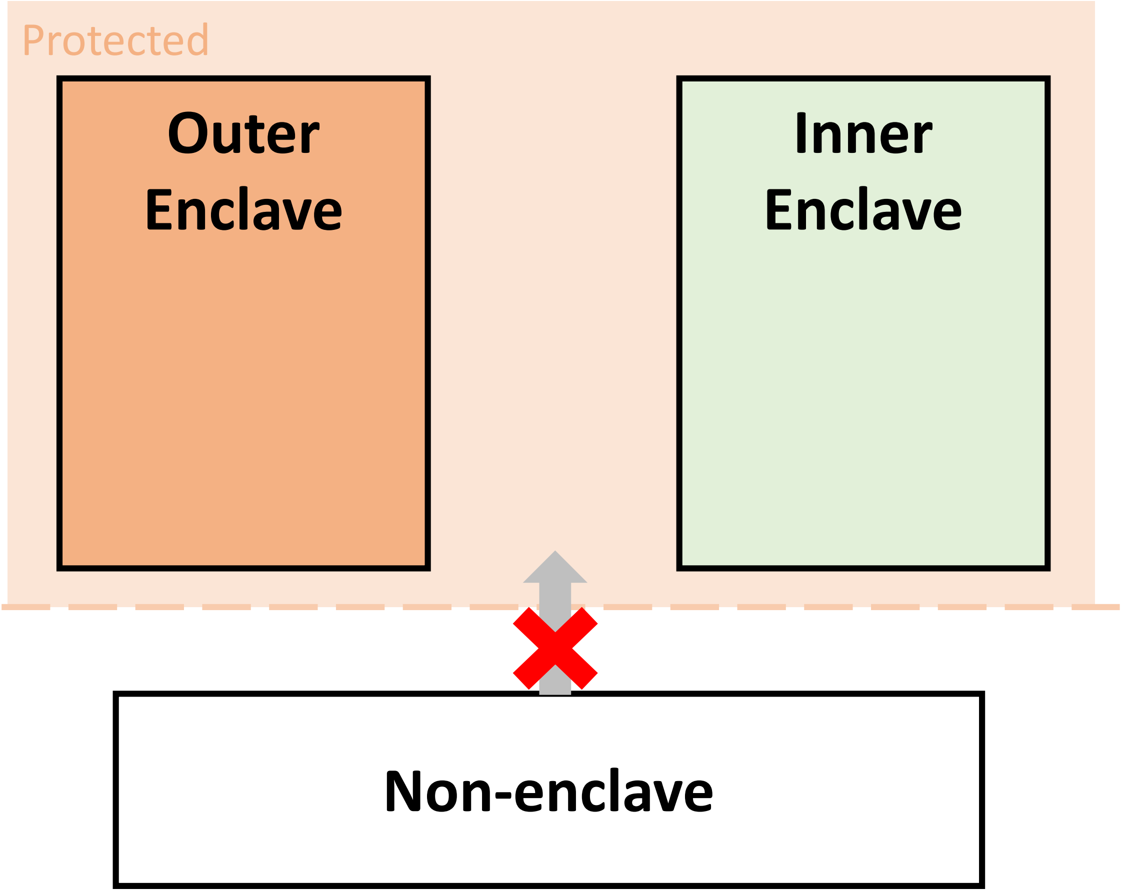
Access Validation (detail)

- Modifications are marked in grey



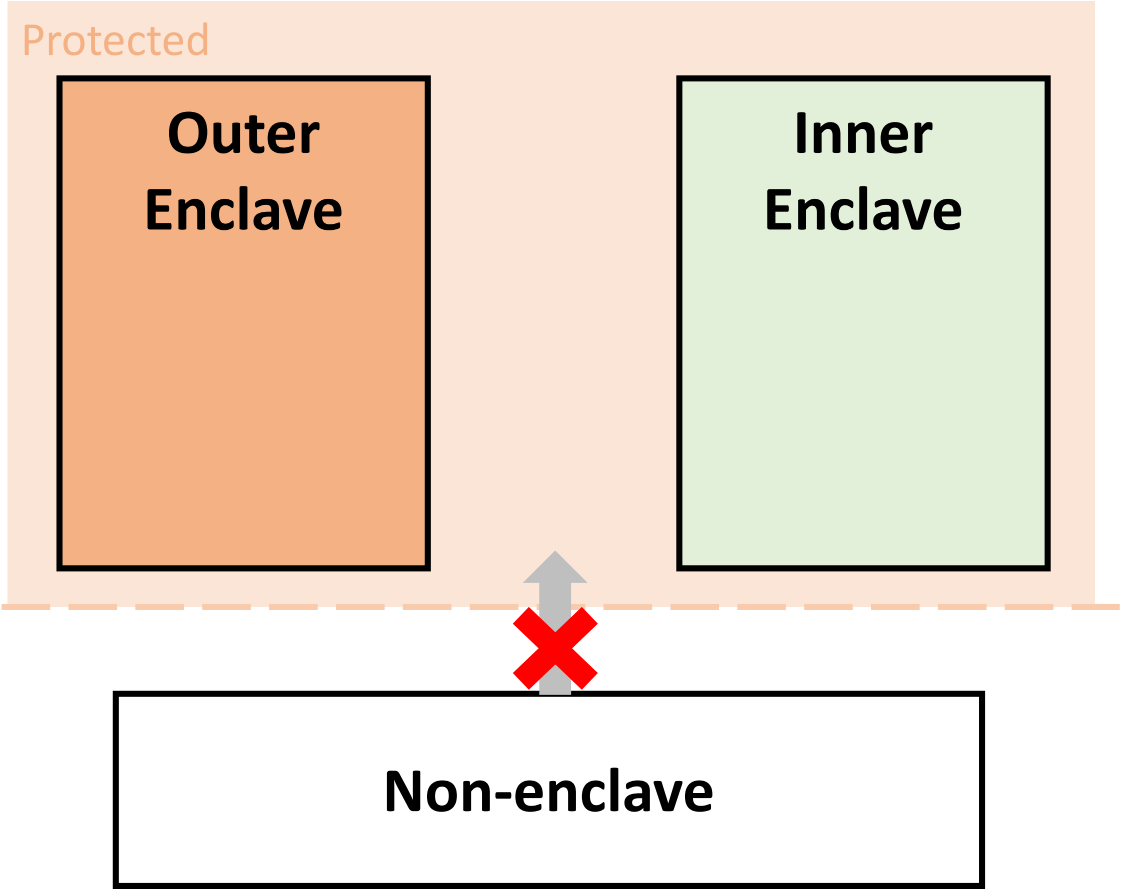
[4] Intel SGX Explained.

Secure Transition



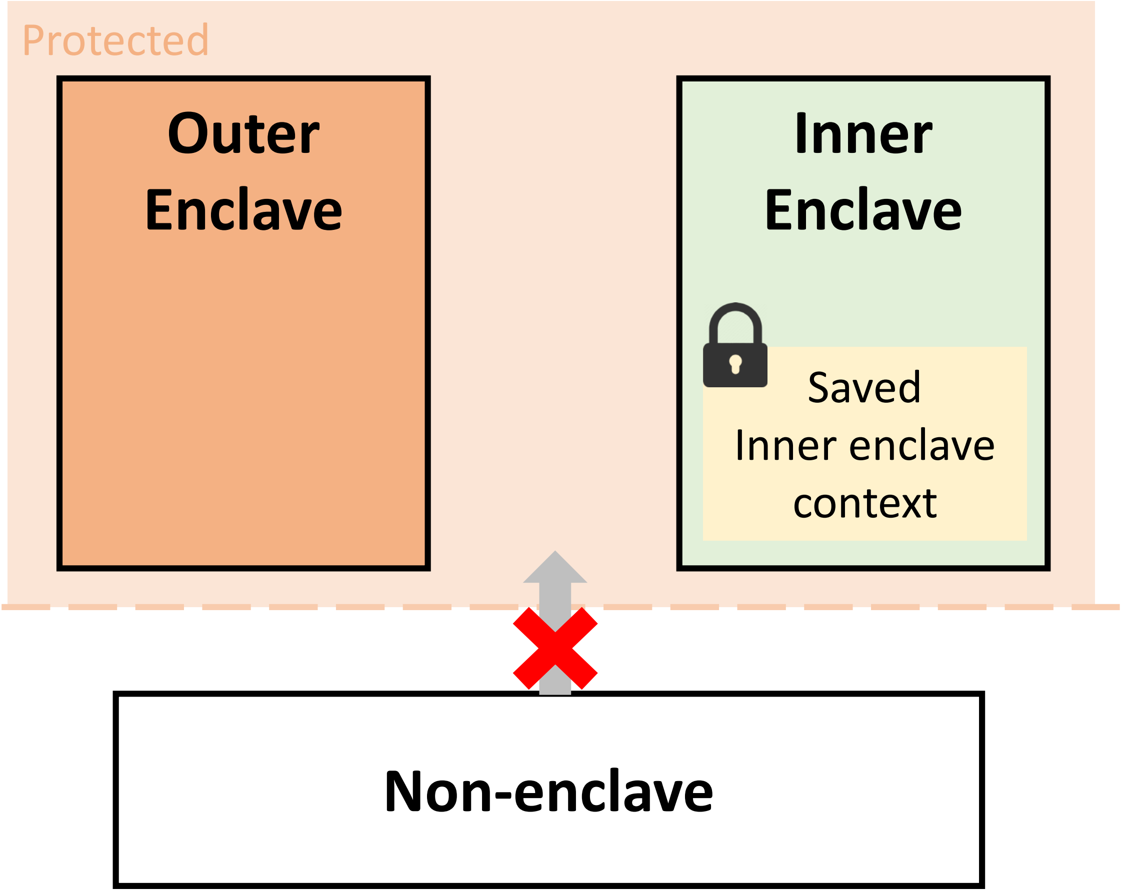
Secure Transition

- Direct transition between inner and outer enclaves



Secure Transition

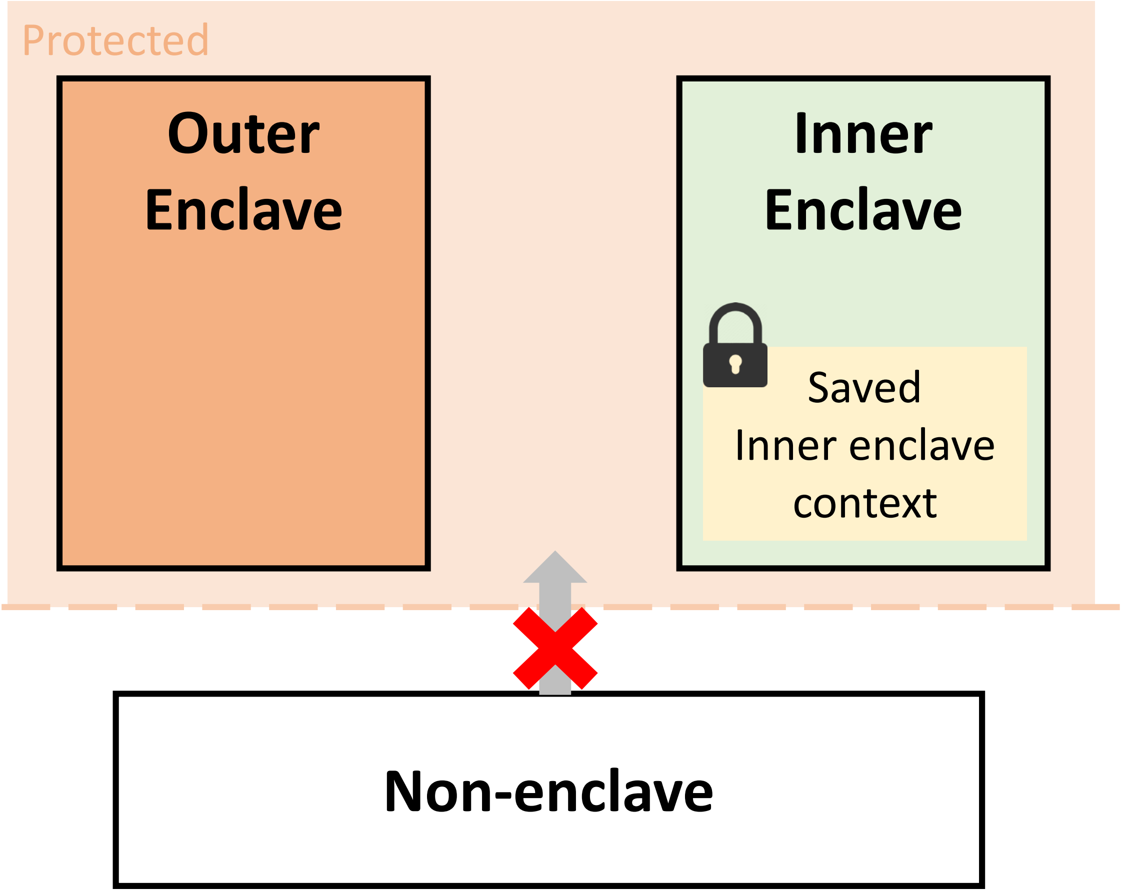
- Direct transition between inner and outer enclaves
- Transition between Inner and outer enclaves
    - Save running context



KAIST School of Computing

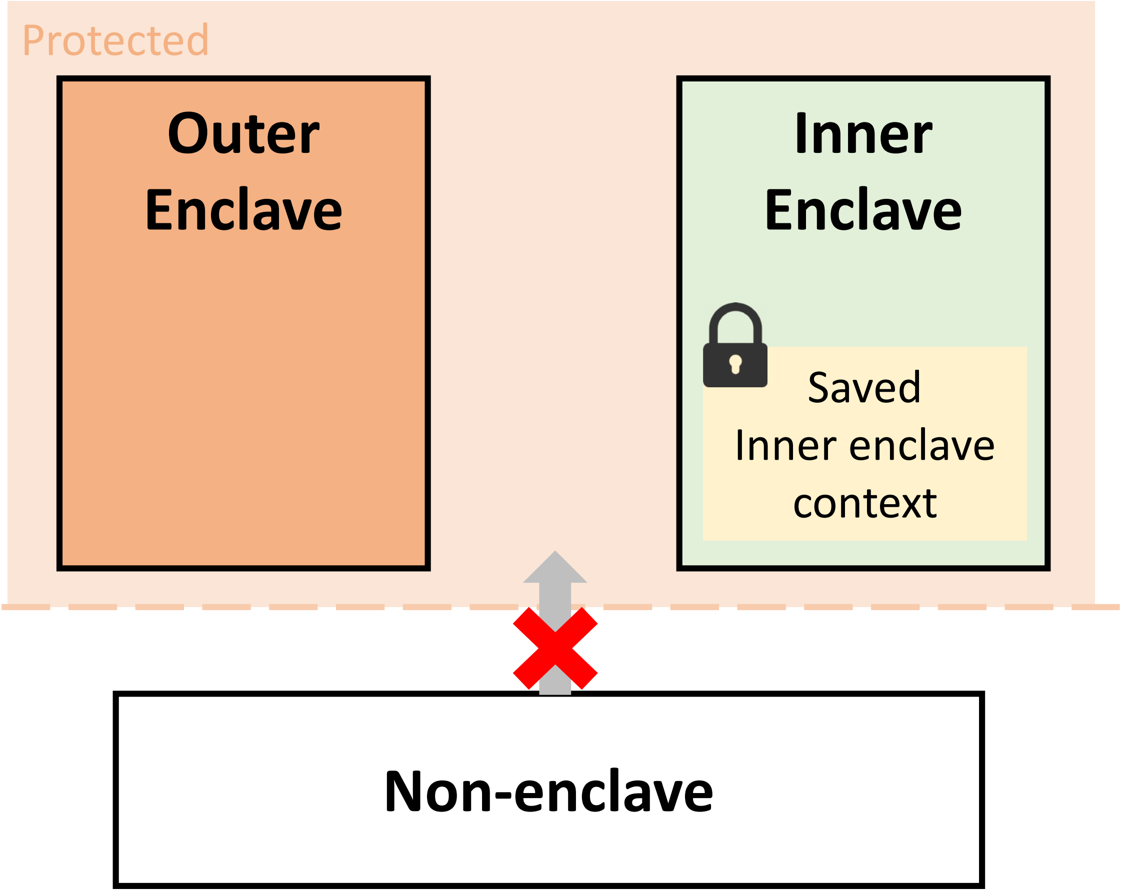
Secure Transition

- Direct transition between inner and outer enclaves
- Transition between Inner and outer enclaves
  - Save running context
  - Flush flags, register, and TLB



Secure Transition

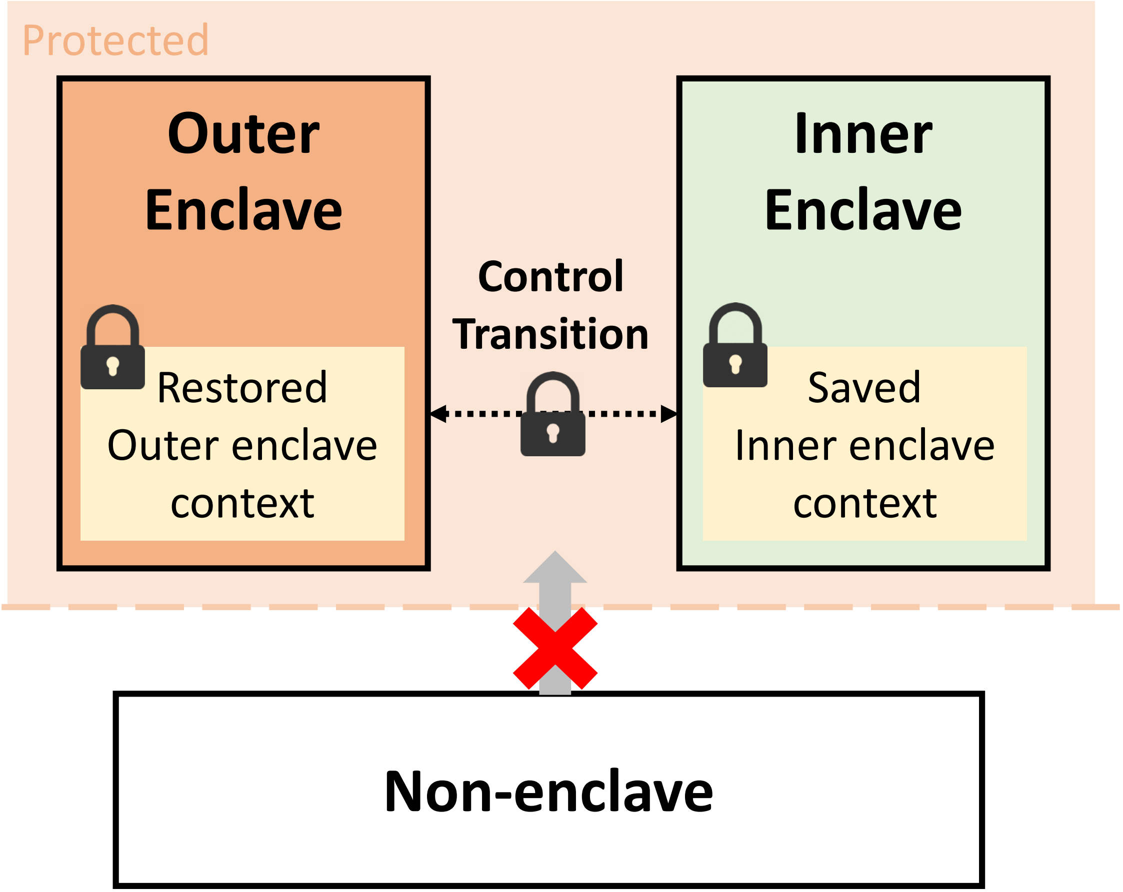
- Direct transition between inner and outer enclaves
- Transition between Inner and outer enclaves
  - Save running context
  - Flush flags, register, and TLB
  - Check & sanitize parameters



Secure Transition

- Direct transition between inner and outer enclaves

- Transition between Inner and outer enclaves
  - Save running context
  - Flush flags, register, and TLB
  - Check & sanitize parameters
  - Restore target context if exists



Evaluation Methodology

Implementation

- New instructions to SDK emulation
  - NEENTER, NEEXIT, NASSO, NEREPORT
- APIs
  - Nested ecall/ocall
  - Association

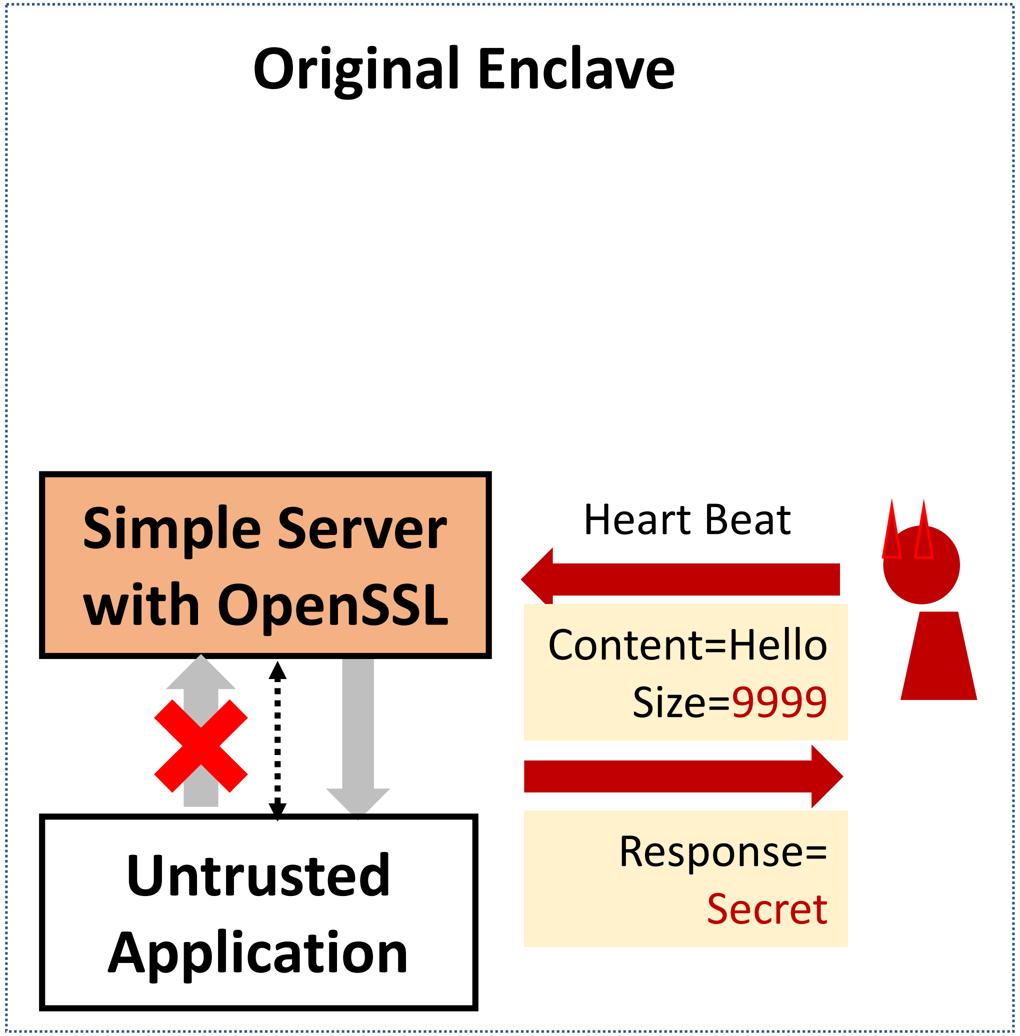
Application porting

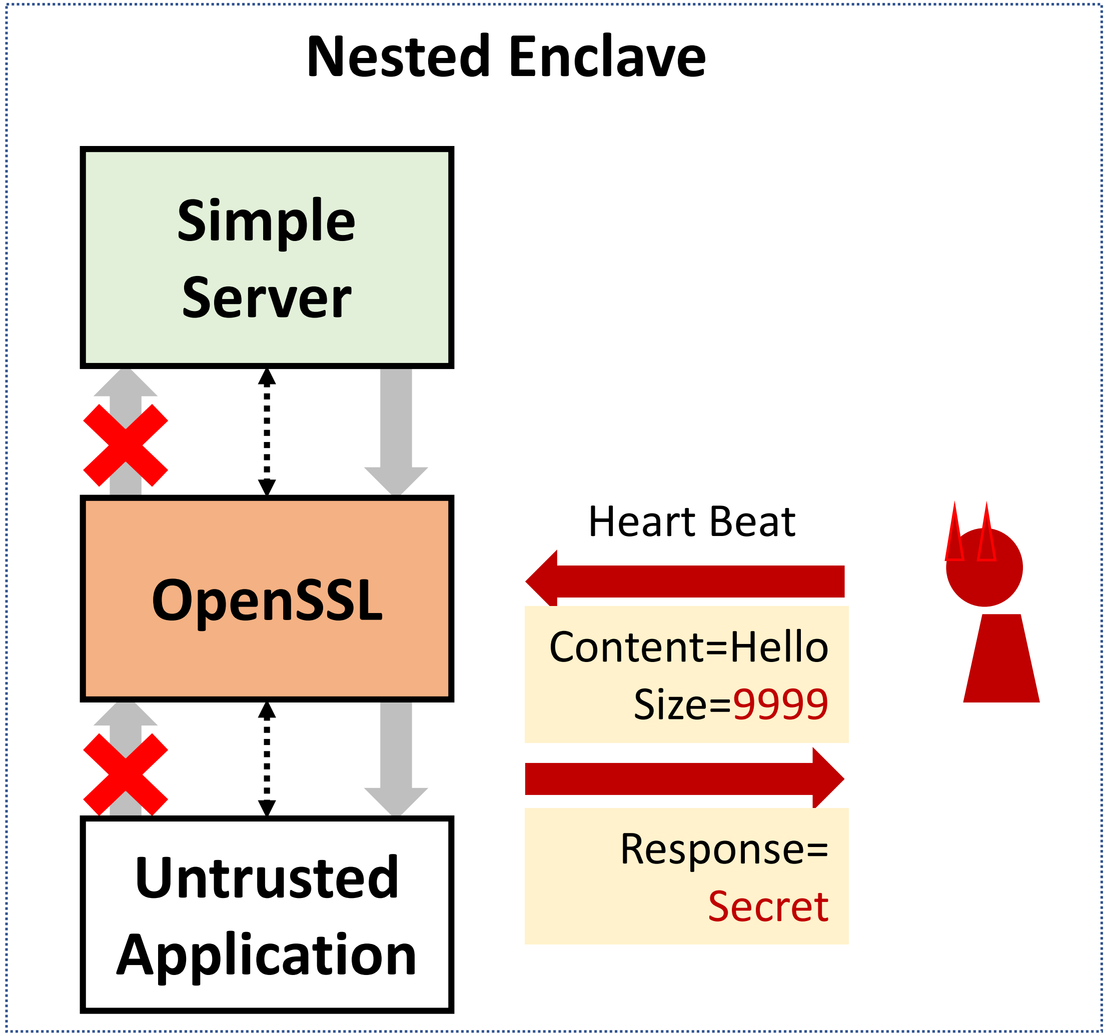
- Echo server with OpenSSL
- Query server with SQLite
- LibSVM (training, prediction)

Evaluation Environment

- Intel i7-7700 64 GB DRAM
- Ubuntu 16.04, Linux kernel 4.13.0.
- Intel SGX SDK / driver v1.9

CASE 1: Heartbleed Attack

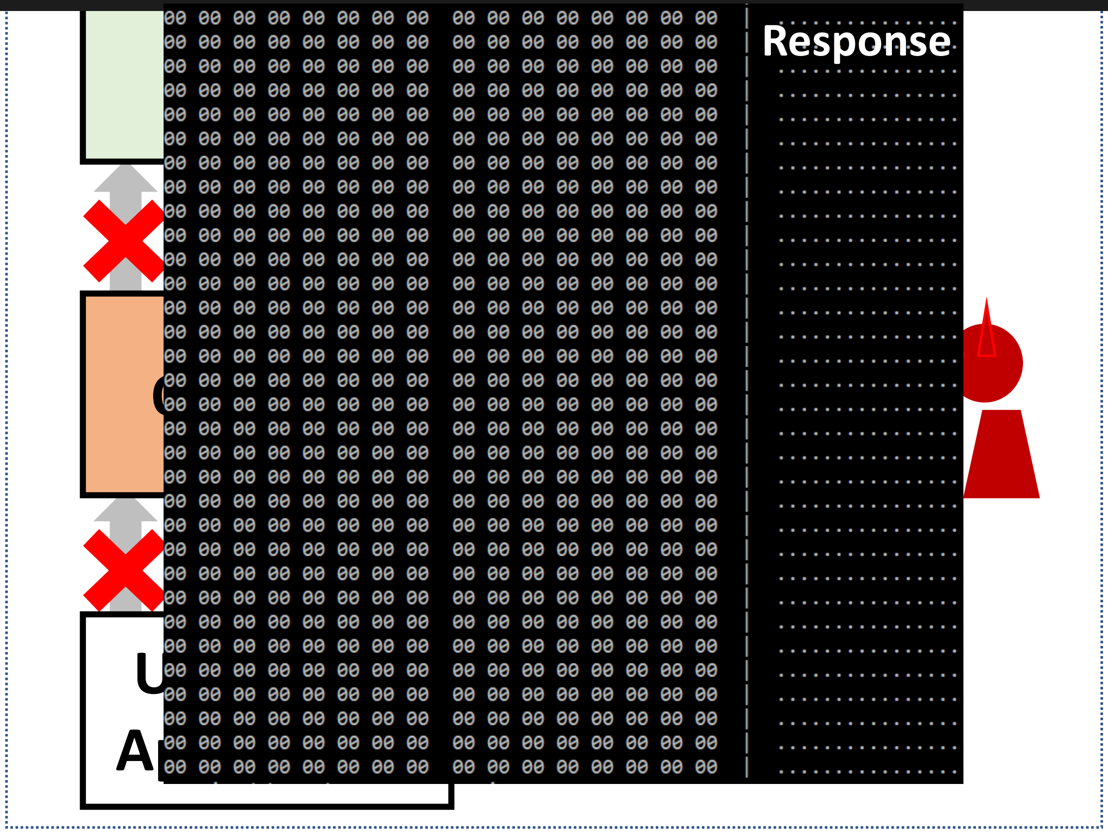




CASE 1: Heartbleed Attack

In Simple Server

```
void info_leak()
{
    char *secret;
    int size = 0x8000;

    secret = (char *) malloc (size);
    for (int i = 0; (i + 1) * 0x100 < size; i++)
        strlcpy (secret + i * 0x100, "ID=admin;PASSWORD=admin_secure_password_1337;EX=deadbeef;TOPSECRET=THISISACONFIDENTIALSTRING;INFO=YOUCANNOTREADITBECAUSEIFREEITAFTERUSETHIS;", 0x100);
    printf ("secret's address: %p, size: %x\n", secret, size);
    free(secret);
}
```

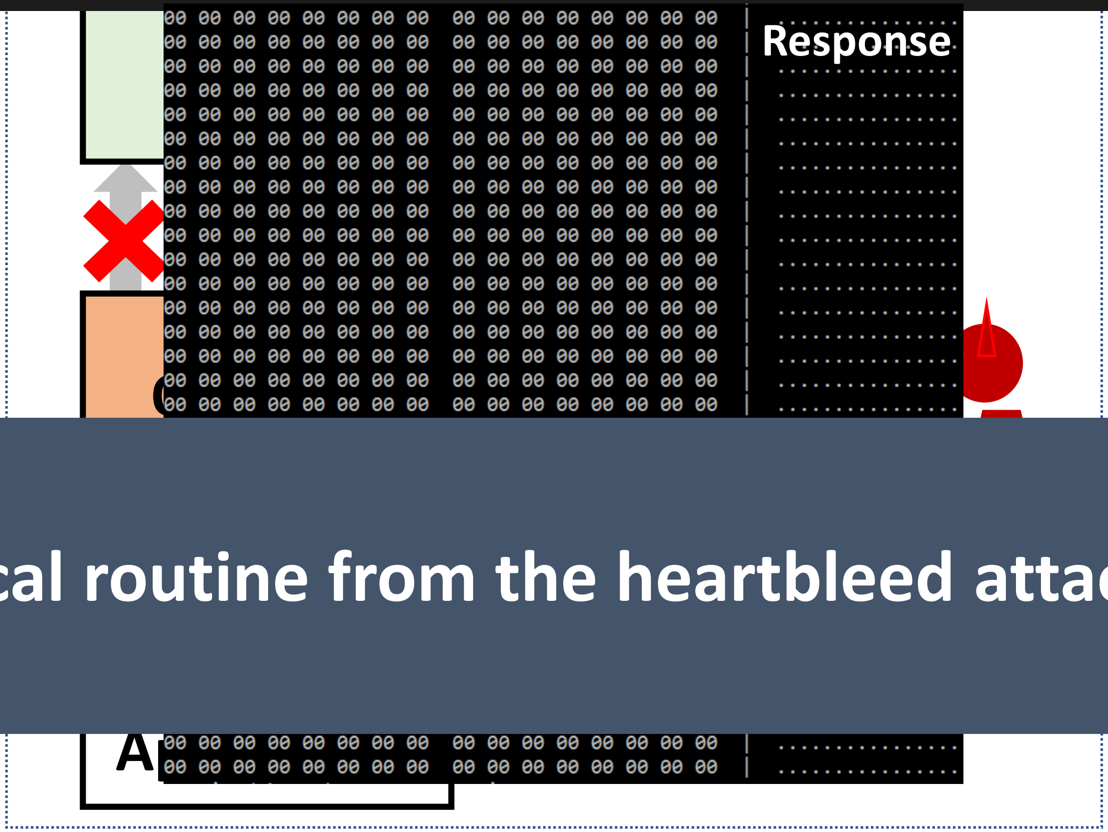


KAIST School of Computing

CASE 1: Heartbleed Attack

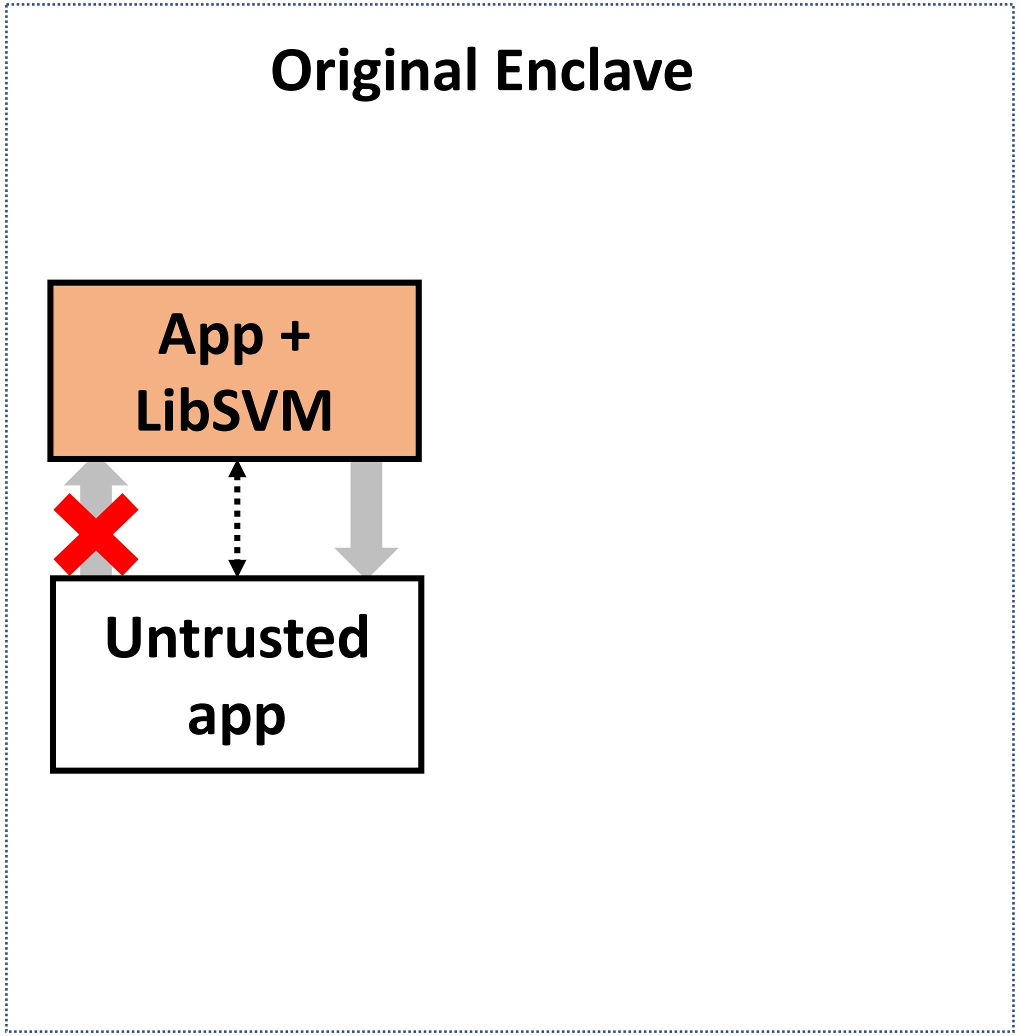
In Simple Server

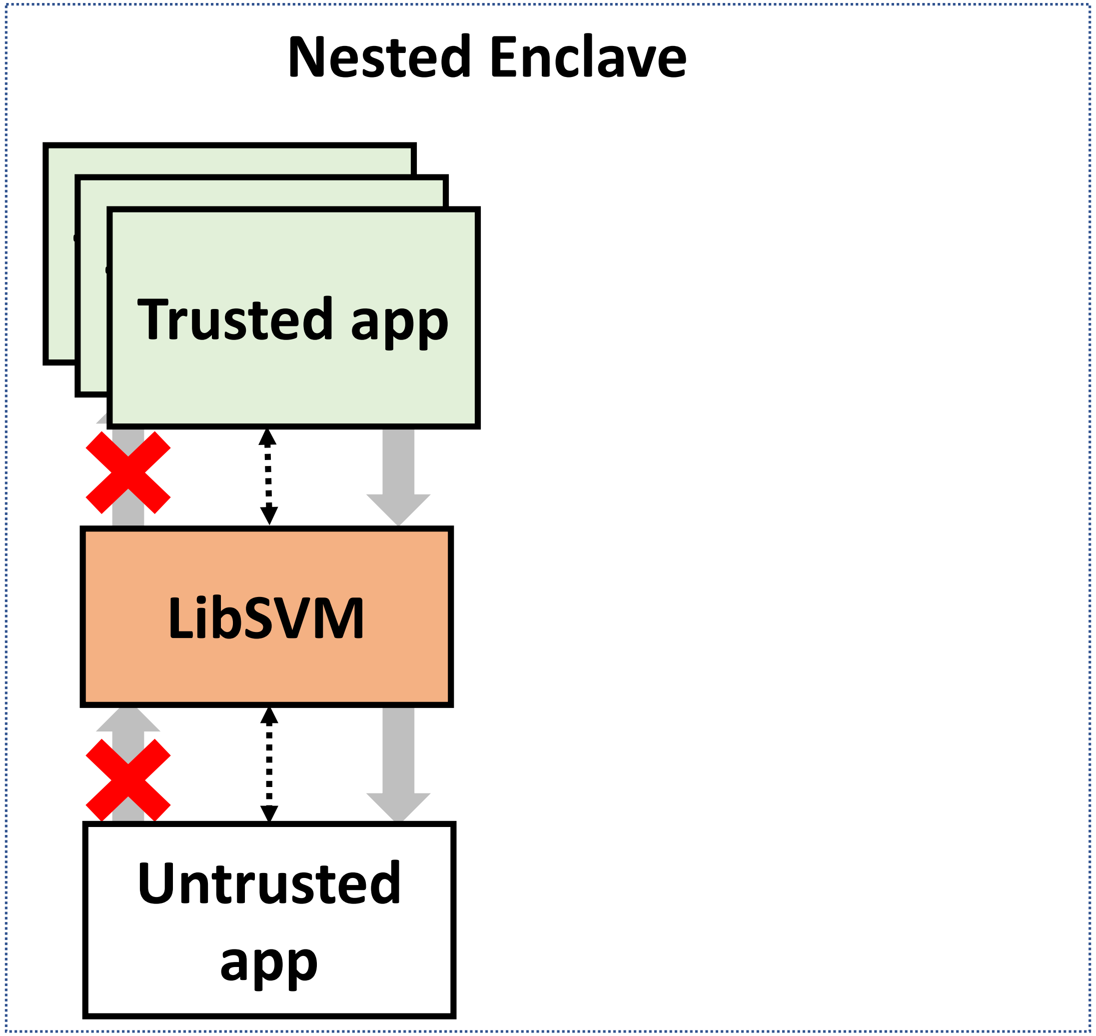
```
void info_leak()
{
    char *secret;
    int size = 0x8000;

    secret = (char *) malloc (size);
    for (int i = 0; (i + 1) * 0x100 < size; i++)
        strlcpy (secret + i * 0x100, "ID=admin;PASSWORD=admin_secure_password_1337;EX=deadbeef;TOPSECRET=THISISACONFIDENTIALSTRING;INFO=YOUCANNOTREADITBECAUSEIFREEITAFTERUSETHIS;", 0x100);
    printf ("secret's address: %p, size: %x\n", secret, size);
    free(secret);
}
```



Nested enclave protects the critical routine from the heartbleed attack

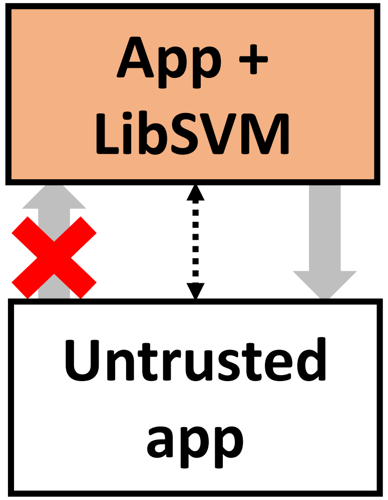
CASE 2: Machine Learning as a Service





CASE 2: Machine Learning as a Service

Original Enclave

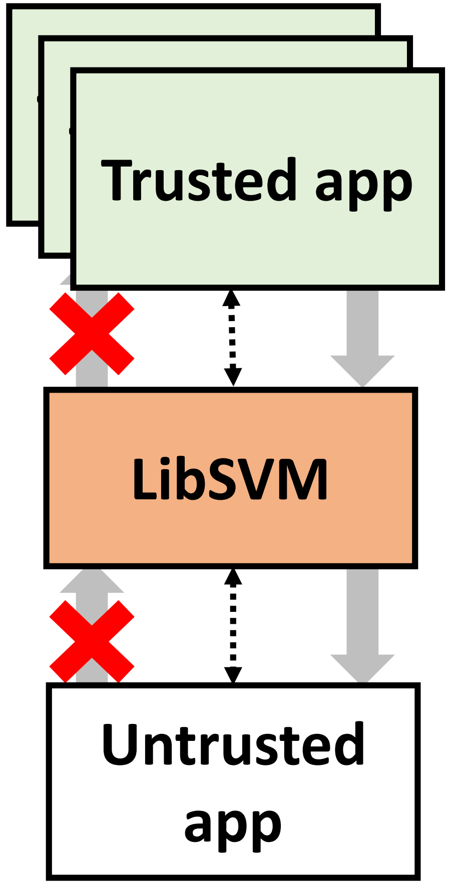


Single enclave:
Private processing+
ML library

User Private Data
- Patient info
- Account info

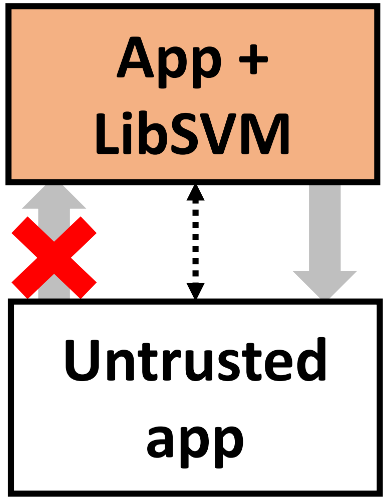
Data for ML
- X-ray pic
- DNA sequence
- Email …

Nested Enclave



CASE 2: Machine Learning as a Service

Original Enclave

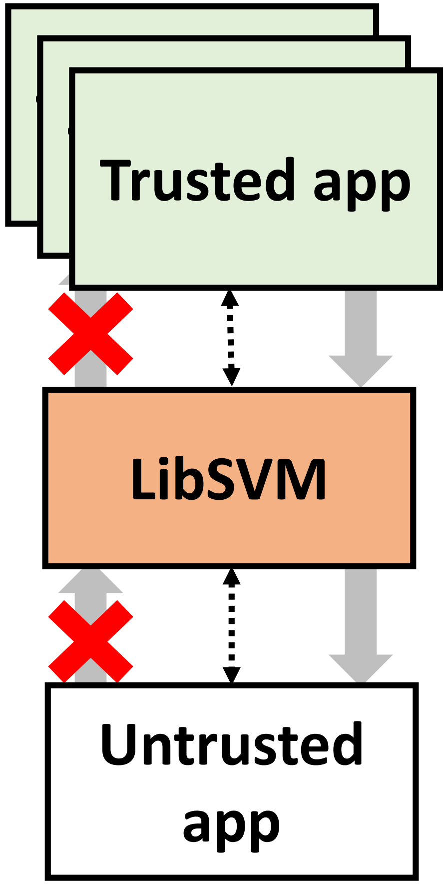


Single enclave:
Private processing+
ML library

User Private Data
- Patient info
- Account info

Data for ML
- X-ray pic
- DNA sequence
- Email …

Nested Enclave



Per-user inner enclave

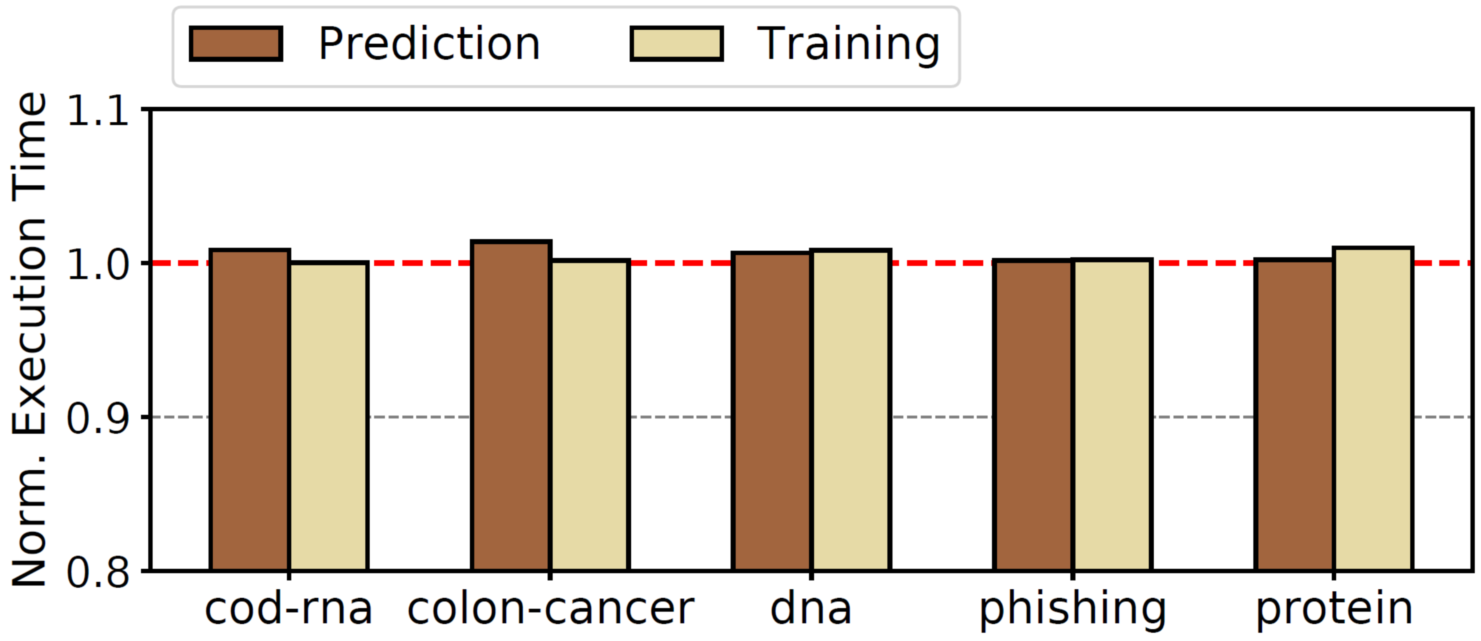
User Private Data
- Patient info
- Account info

Outer enclave: ML library

Data for ML
- X-ray pic
- DNA sequence
- Email …

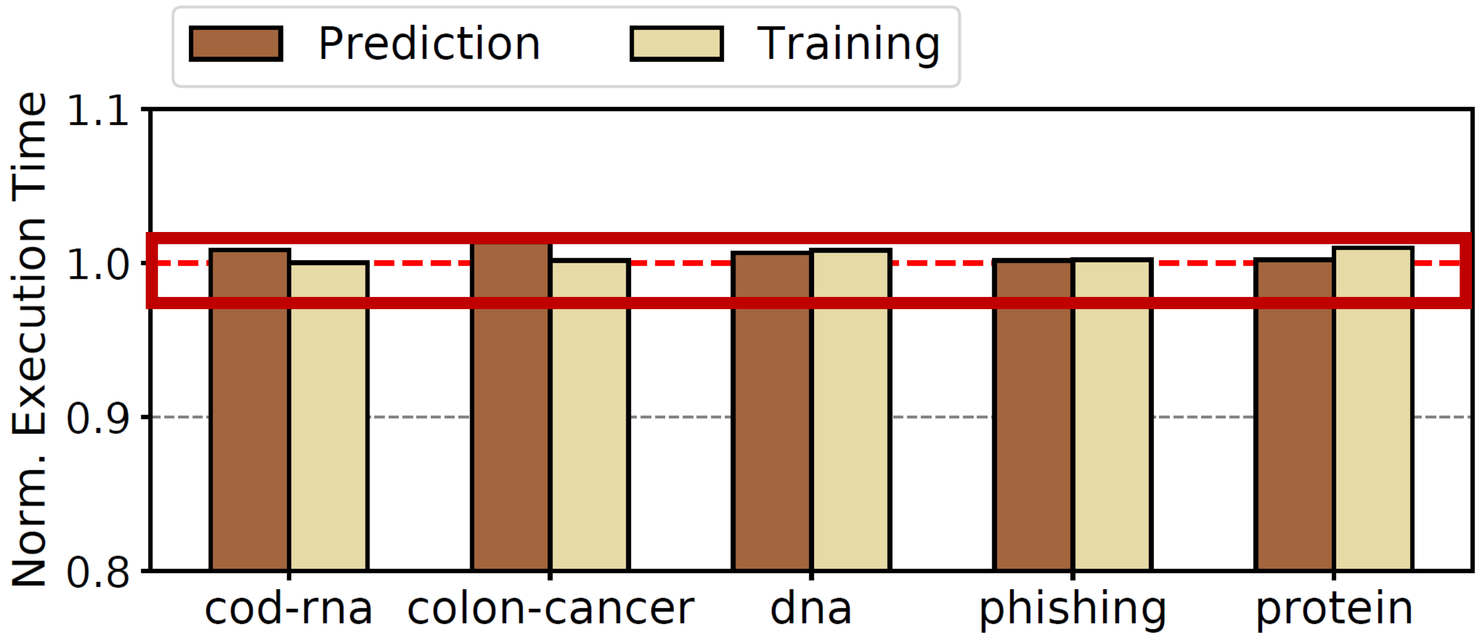
CASE 2: Machine Learning as a Service

- **Normalized execution time to the original enclave**
- **Overhead is relatively small (< 2%)**



CASE 2: Machine Learning as a Service

- **Normalized execution time to the original enclave**
- **Overhead is relatively small (< 2%)**



More case studies in paper..

- Library sharing and time to loading enclaves
- Shared SQLite server
- Communication with intra- vs inter enclave channels

Porting effort

Name	Modification	Modified LOC	Original
Echo server	C/C++ code	34	883
	EDL	10	28
	SGX-OpenSSL	0	507k
SVM-predict	C/C++ code	27	208
	EDL	10	49
	SGX-LibSVM	0	152k
SVM-train	C/C++ code	24	333
	EDL	10	41
	SGX-LibSVM	0	152k

KAIST School of Computing

Porting effort

Name	Modification	Modified LOC	Original
Echo server	C/C++ code	34	883
	EDL	10	28
	SGX-OpenSSL	0	507k
SVM-predict	C/C++ code	27	208
	EDL	10	49
	SGX-LibSVM	0	152k
SVM-train	C/C++ code	24	333
	EDL	10	41
	SGX-LibSVM	0	152k

KAIST School of Computing

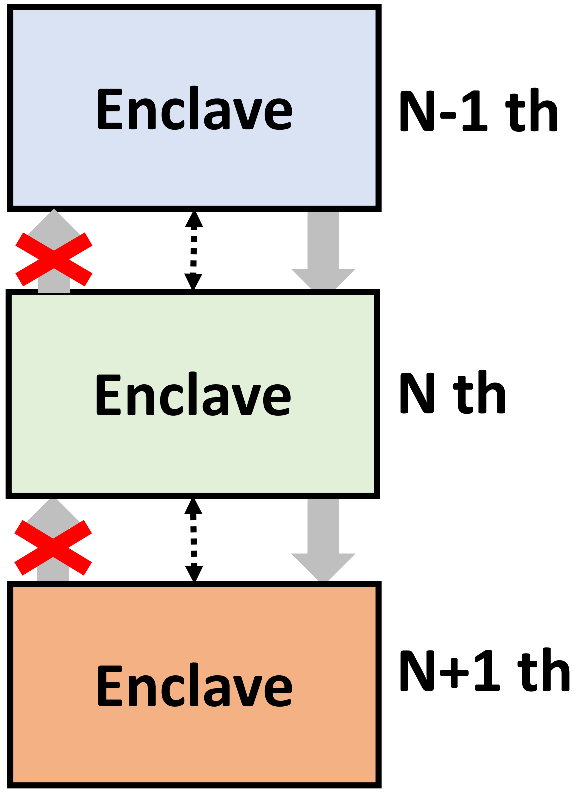
Porting effort

Name	Modification	Modified LOC	Original
Echo server	C/C++ code	34	883
	EDL	10	28
	SGX-OpenSSL	0	507k
SVM-predict	C/C++ code	27	208
	EDL	10	49
	SGX-LibSVM	0	152k
SVM-train	C/C++ code	24	333
	EDL	10	41
	SGX-LibSVM	0	152k

Extending Nested Enclave

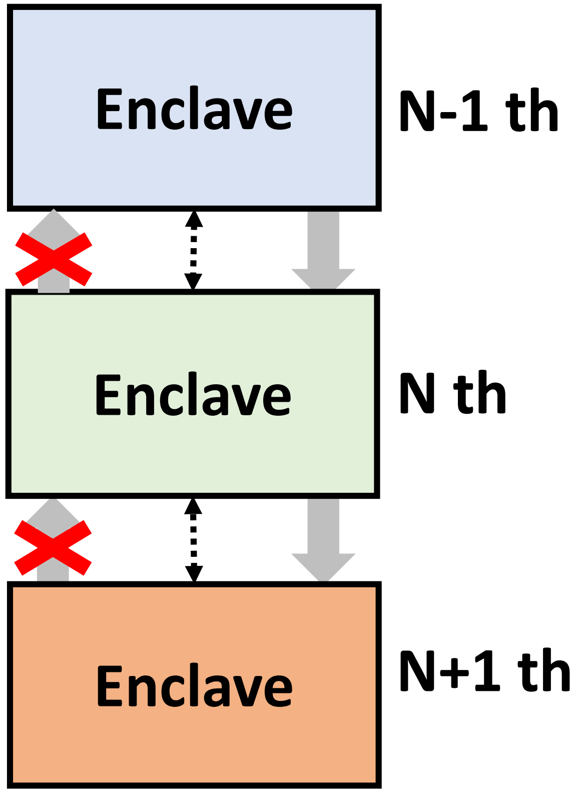
Extending Nested Enclave

N-depth nesting

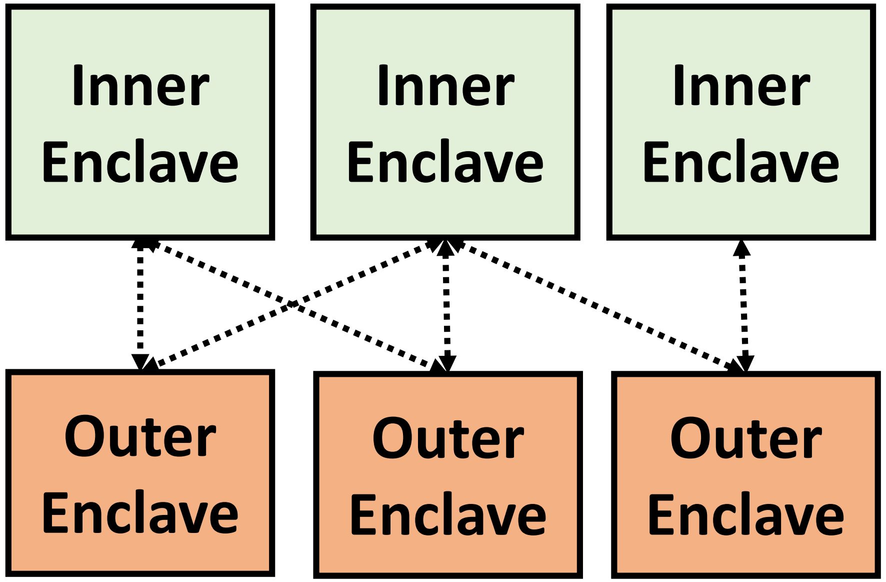


Extending Nested Enclave

N-depth nesting



N to M relation



Conclusion

- A new extension to the trusted execution environments for supporting multi-level security within TEE.
- The required hardware and software extensions for the nested enclave are small.
- Case studies on the nested enclave emulation framework.

Thank you