Intel SGX

한성대학교
김경호

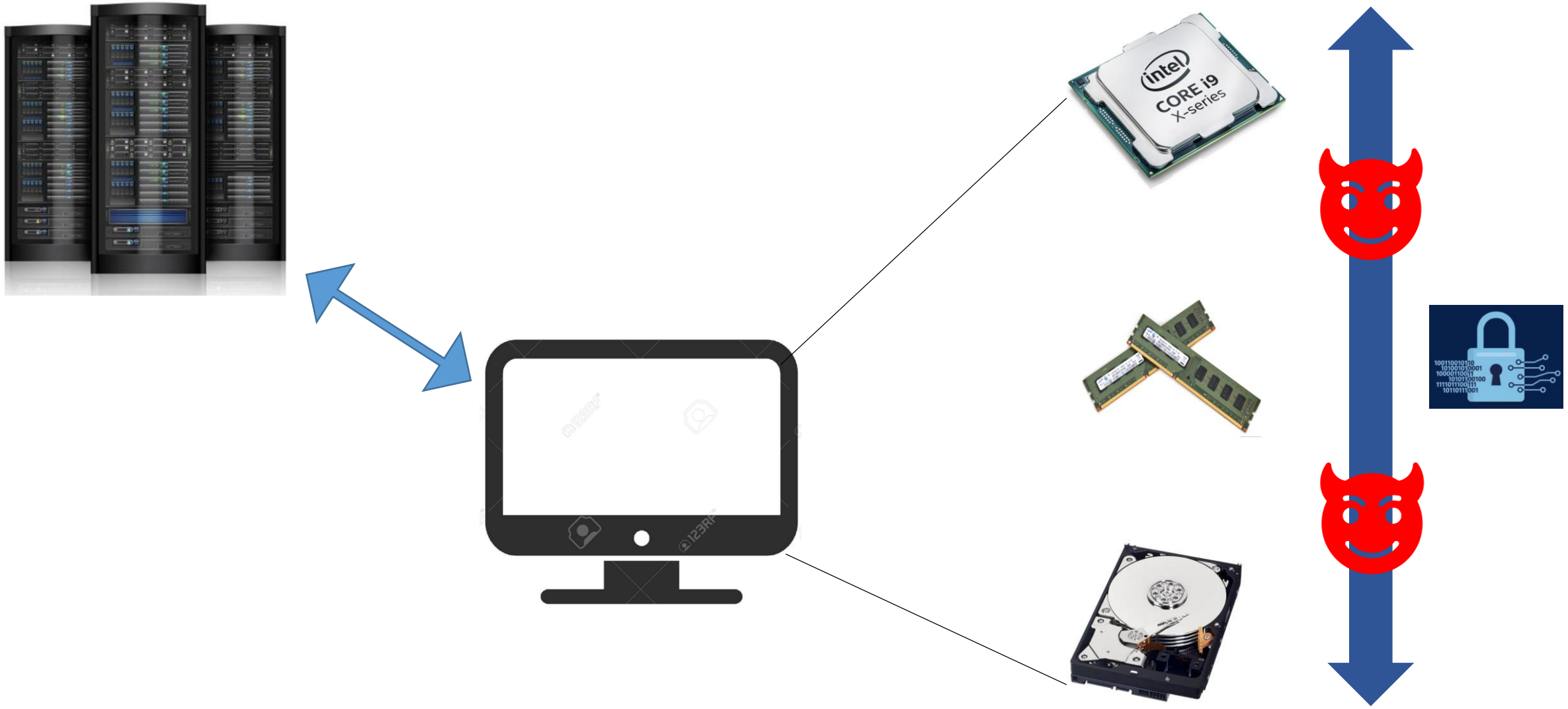
HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

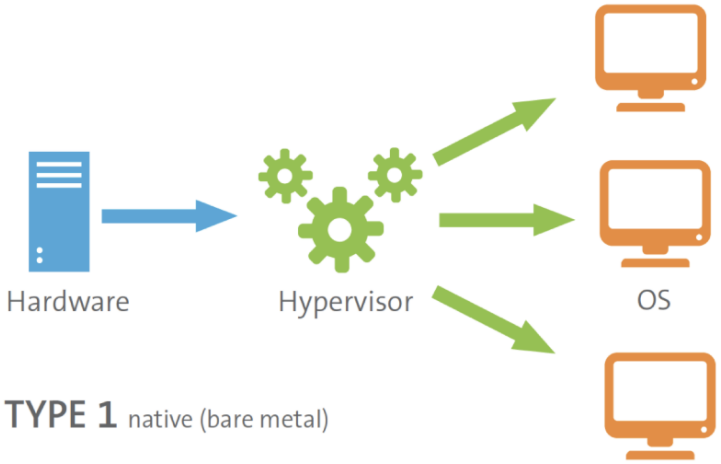
Contents

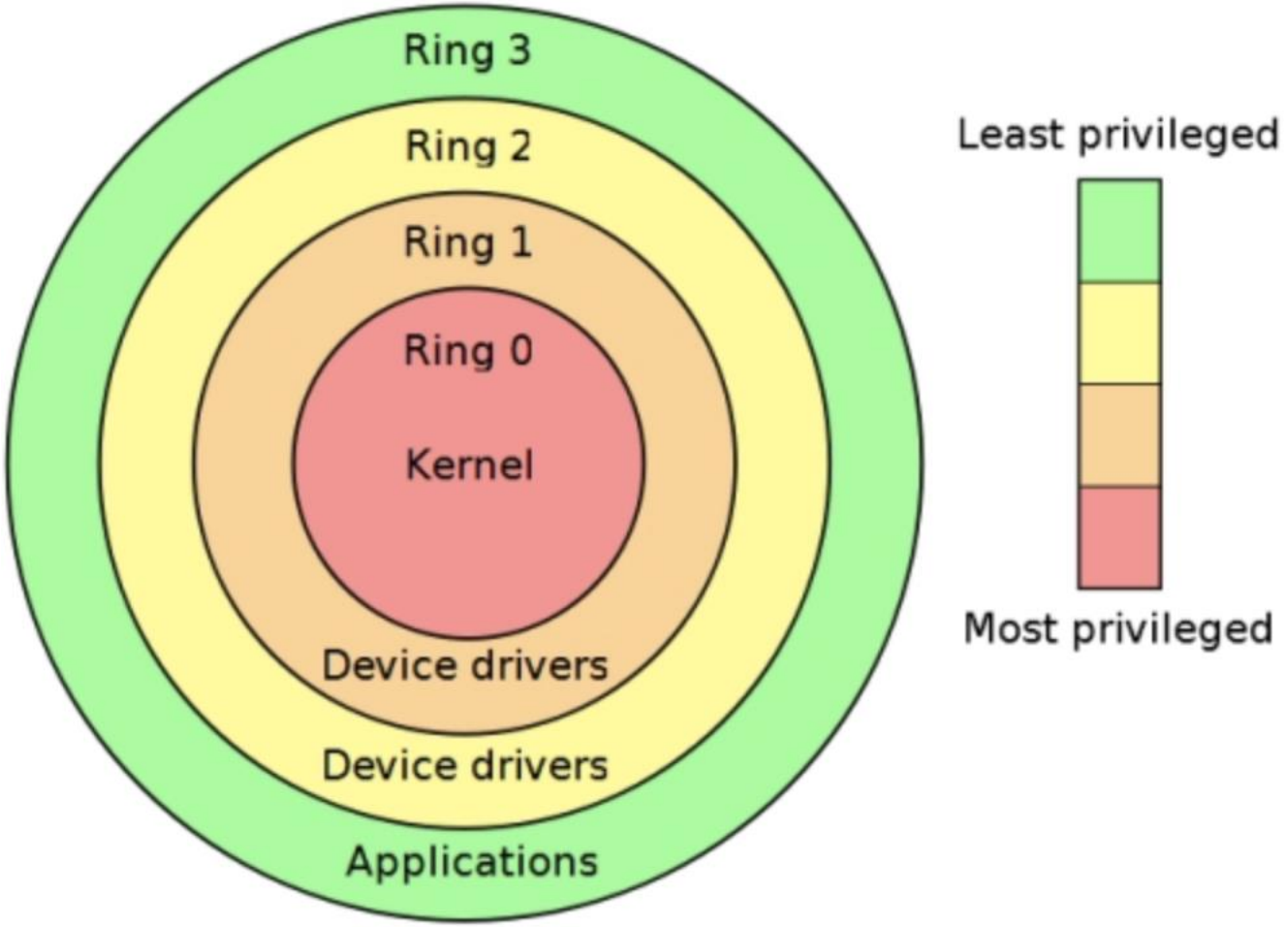
CryptoCraft LAB

1. What is Intel SGX



1. What is Intel SGX





1. What is Intel SGX | SE, TPM

- SE (Secure Elements)

  - 중요 데이터 저장 및 결제 같은 Secure App을 안전하게 실행할 수 있는 플랫폼
  - Malware Attack으로부터 데이터와 응용프로그램을 지켜주는 역할
  - 저장 용량 및 처리 속도가 제한적

- TPM (Trusted Platform Module)

  - 하드웨어 기반의 보안 관련한 기능만 제공하는 보안 암호화 프로세서
  - 다양한 변조 방지 메커니즘을 사용하여 안전한 암호화 연산 수행
  - 시스템 무결성 측정 및 키 생성 및 사용
  - 저장 용량이 제한적이고 암호화 기능만 수행함

1. What is Intel SGX | TEE

- TEE (Trusted Execution Environment)
  - 신뢰 할 수 있는 실행 환경
  - 메인 프로세서 내부의 보안 영역으로 격리된 환경에서 운영체제와 병렬 실행
  - 격리된 환경을 통한 응용프로그램의 무결성 및 기밀성 제공
  - H/W, S/W 측면에서 보안성 극대화
  - 신뢰 공간과 비신뢰 공간 (Secure , Normal)

  - Intel SGX(Software Guard eXtensions)
  - ARM TrustZone
  - AMD PSP(Platform Security Processor)

1. What is Intel SGX | Intel SGX

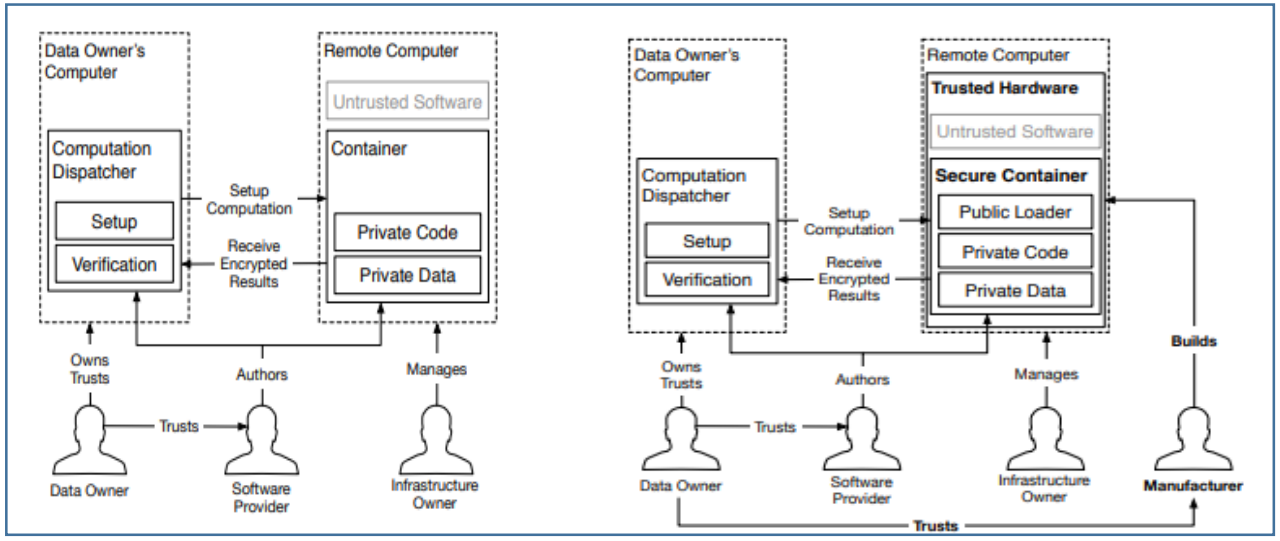
- Intel SGX

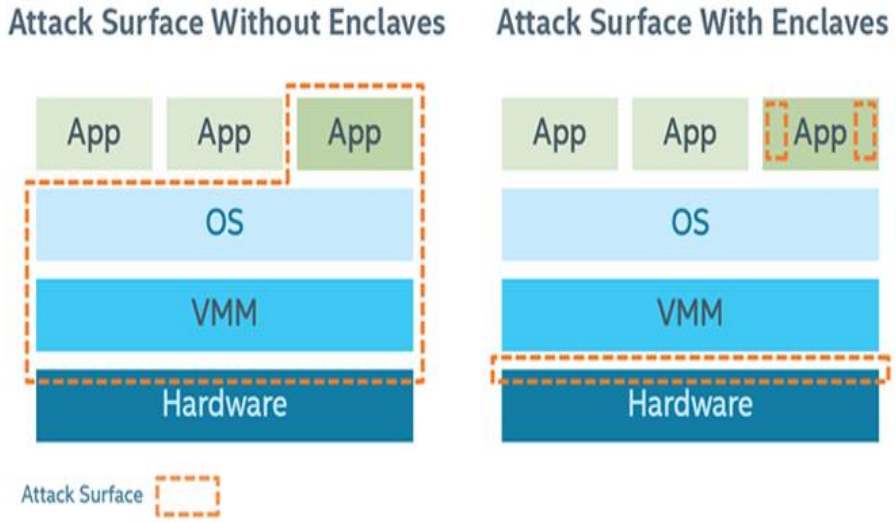
Intel에서 제공하는 CPU 명령어 코드

Enclave라는 Secure Container 제공

운영체제, 하이퍼바이저 포함 어떤 수준의 권한으로도 접근이 불가능

Enclave를 사용함으로써 공격 범위를 효과적으로 경감





1. What is Intel SGX | H/W Security

- MEE(Memory Encryption Engine) 을 이용하여 메모리 암호화

  (물리적 메모리 공격 보호)

- Enclave 실행 후 Disk 저장 파일 암호화

  (디스크에 저장된 중요 데이터 보호)

- 다양한 종류의 암호화 키 제공

- 부채널 공격을 방어하는 메커니즘은 존재하지 않음

  (개발자가 부채널 내성 프로그래밍을 해야함)

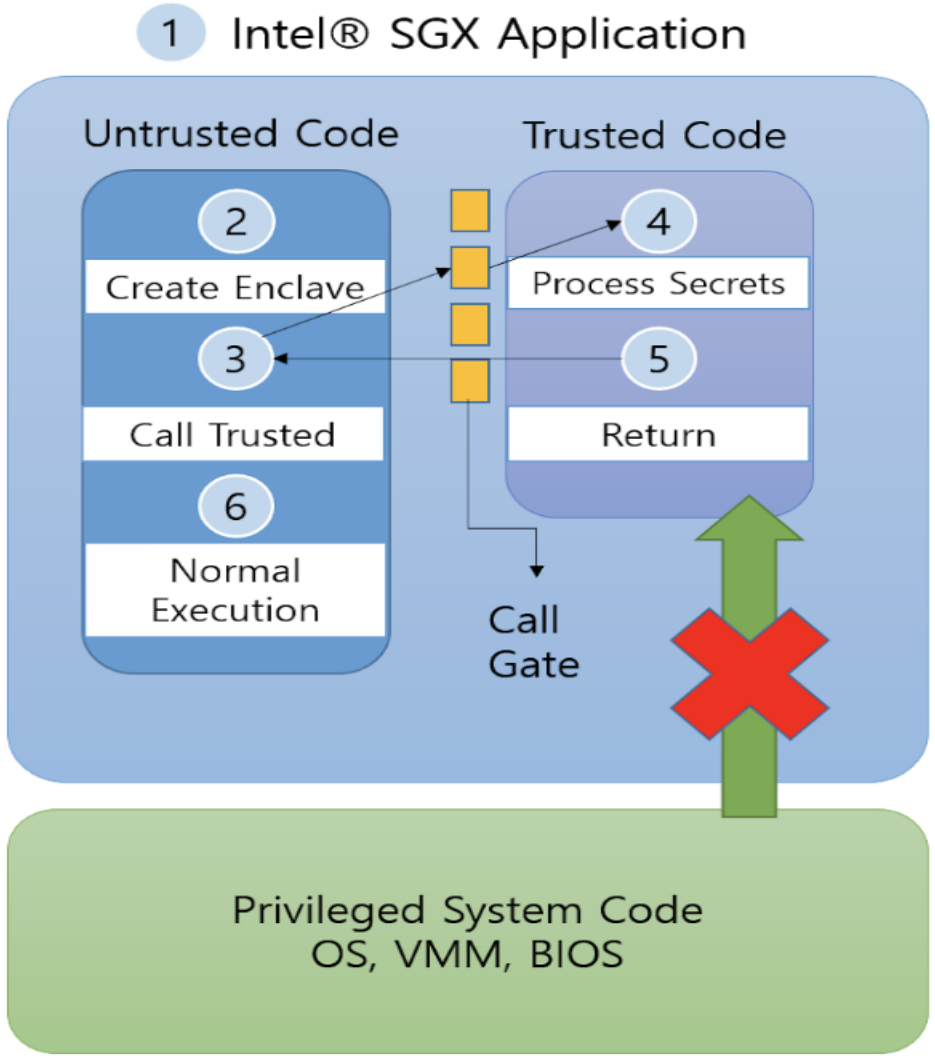
1. What is Intel SGX | S/W Security

- BIOS, VMM, OS 등의 높은 권한의 System S/W도 접근 불가능
  (System S/W 해킹시에도 저장된 데이터 무결성 및 기밀성 보장)

- jump, function call 등의 분기문으로 Enclave 메모리 접근 불가
  (StackOverflow 같은 비정상적 분기를 이용한 메모리 접근 방어)

- 리버싱을 방어하는 메커니즘은 존재하지 않음
  (Source Code 작성시 하드코딩 및 취약점 존재하는 코딩 자제)

1. What is Intel SGX | Mechanism

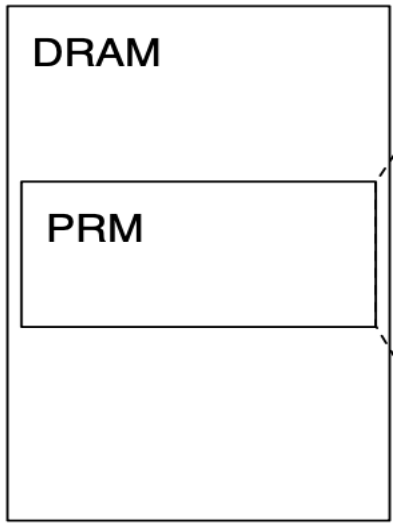


1. Trusted zone, Untrusted zone 설정
2. Enclave 생성 (Trusted Zone)
3. Enclave 실행 (ECALLS)
4. 프로세스 수행 (외부에서 접근 불가능)
5. Return으로 반환값 반환 (OCALLS)
   (Enclave 데이터는 여전히 Trusted memory에 저장)
6. 기존 연산 수행

2. SGX Memory Layout | PRM

- PRM (Processor Reserved Memory)

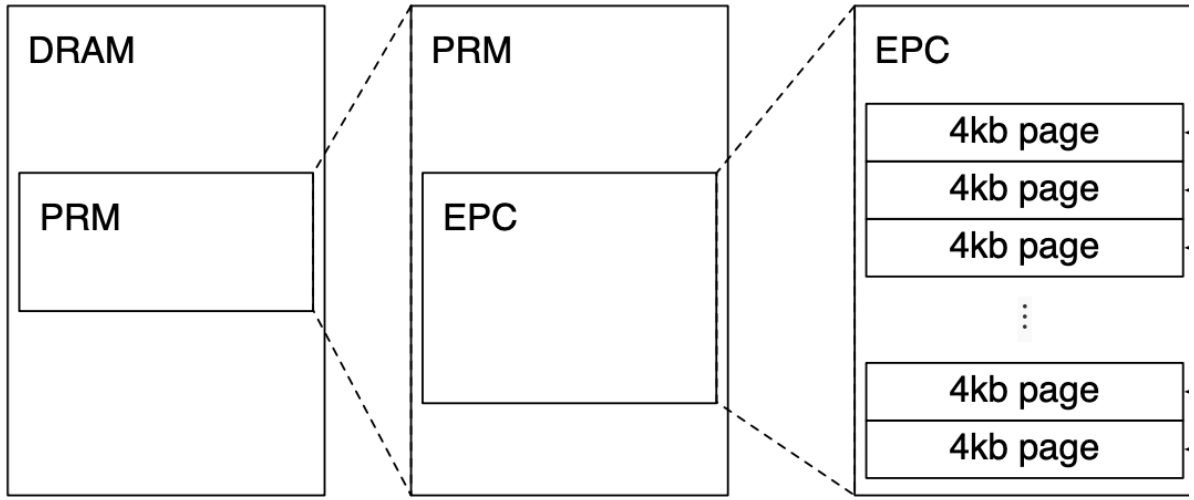
Enclave Code 및 Data가 저장되는 DRAM에서 분리된 메모리
System S/W 및 다른 소프트웨어에서 접근 불가능
CPU의 통합 메모리 컨트롤러가 다른 Device의 DMA(직접 메모리 접근) 거부
Memory 특성을 그대로 사용하여 2^n 의 크기를 가짐



2. SGX Memory Layout | EPC

- EPC (Enclave Page Cache)

Enclave Code, Data를 저장하는 4KB Page들로 구성된 PRM의 Subset
Architecture의 주소 변환 기능과 동일한 Page size 사용
Non-Enclave S/W는 EPC에 접근 불가능
System S/W에 의해 EPC를 관리 (SGX 명령어 사용하여 할당 및 해제)

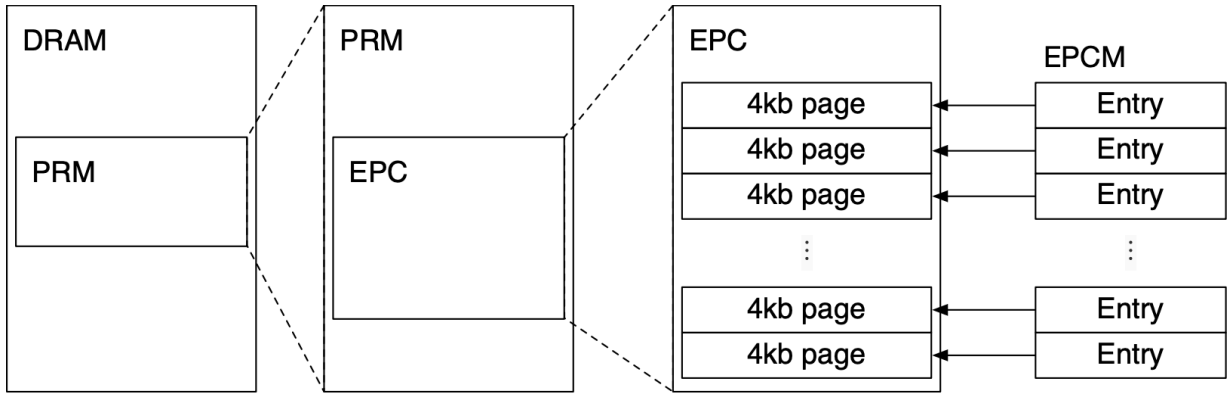


2. SGX Memory Layout | EPCM

- EPCM (Enclave Page Cache Map)

  System S/W는 믿을 수 없는 S/W로 간주하여 확인을 위해 Map을 사용
  SGX의 보안 점검에서만 사용
  한 EPC page에 2개 이상의 Enclave를 할당했는지 확인 (VALID -> 1)
  PT -> Page type (PT_REGS, PT_SECS, etc..)
  ENCLAVESECS -> Enclave SECS 주소 (Enclave에서 다른 Enclave 접근 방지)

Field	Bits	Description
VALID	1	0 for un-allocated EPC pages
PT	8	page type
ENCLAVESECS		identifies the enclave owning the page



2. SGX Memory Layout | SECS

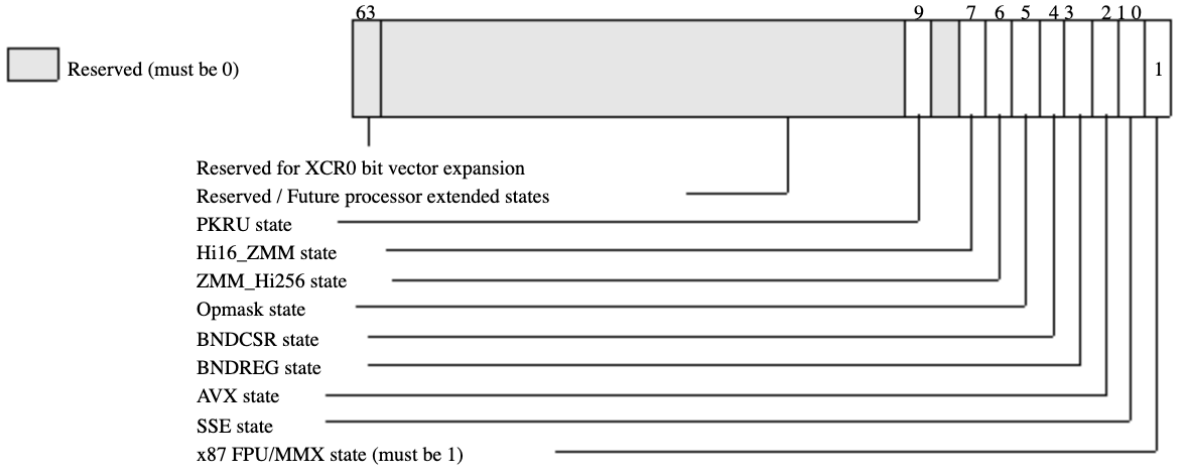
- SECS (The SGX Enclave Control Structure)

Enclave에 대한 Metadata
EPCM의 PT field에 PT_SECS 로 저장
Enclave를 생성할 때 최초로 SECS page 할당
Enclave를 해제할 때 마지막으로 SECS Page 해제
특정 Enclave의 가상 주소에 Mapping되는게 아니라 SGX가 독점 사용
(Enclave code에서 접근이 불가능)

2. SGX Memory Layout | SECS -> Attributes

- SGX Enclave Attributes

Enclave SECS의 ATTRIBUTES field
DEBUG -> 1이면 디버깅 가능 (SGX 디버깅은 개발 단계에서만 사용)
XFRM -> CPU의 XCR0 register 값으로 설정
(XCR0 -> 응용 프로그램에서 사용하는 기능 비트맵)
MODE64BIT -> 64bit CPU인 경우 1 설정 (32bit 역호환성을 위해)



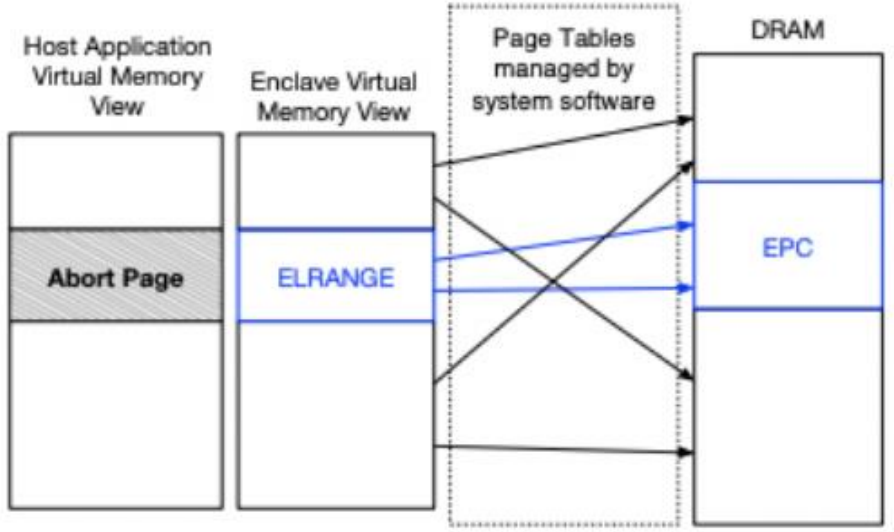
Field	Bits	Description
DEBUG	1	Opts into enclave debugging features.
XFRM	64	The value of XCR0 (§ 2.6) while this enclave's code is executed.
MODE64BIT	1	Set for 64-bit enclaves.

2. SGX Memory Layout | ELRANGE

- ELRANGE(The Enclave Linear Address Range)

Enclave의 Virtual Address Space (EPC page의 Code 및 Data를 Mapping)
ELRANGE 내부의 데이터는 무결성 보장 외부는 신뢰 X
Enclave가 동적 라이브러리를 나타내는 경우 Loader가 ELRANGE 내부로 예약 메모리 설정

Enclave Code -> Non-Enclave Code O
Non-Enclave Code -> Enclave Code X(fault)



2. SGX Memory Layout | Address Translation

- Address Translation for SGX Enclaves

Enclave는 System S/W의 Page Table 및 EPC Page 관리 및 Host 애플리케이션과 동일한 주소 변환 프로세스 사용

신뢰 할 수 없는 System S/W가 관리하기 때문에 기존의 메모리 취약점 존재

각 EPC Page가 특정 가상 주소에만 Mapping 되도록 설정

(EPCM->ADDRESS field에 예상 가상 주소 저장 후 실제 변환 가상 주소와 같은지 비교)

EPC Page의 액세스 권한을 항상 일치 시킴으로써 보안성 확장

Field	Bits	Description
ADDRESS	48	the virtual address used to access this page
R	1	allow reads by enclave code
W	1	allow writes by enclave code
X	1	allow execution of code inside the page, inside enclave

2. SGX Memory Layout | TCS

- TCS (The Thread Control Structure)

SGX는 멀티 코어 프로세서를 지원(멀티 쓰레드를 이용한 동일 Enclave 실행 가능)
각 논리 프로세서마다 TCS를 이용해서 Enclave 코드 실행
EPCM의 PT field에 PT_TCS 로 저장
SECS Page와 마찬가지로 Enclave Code에서 접근 불가능(Debug 때는 가능)
OENTRY -> Enclave의 명령어 시작 주소 (Entry point)
OFSBASGX -> FS segment register 값
OGSBASGX -> GS segment register 값
(FS 와 GS는 일반적으로 Thread Local Storage(TLS)를 가르킴)

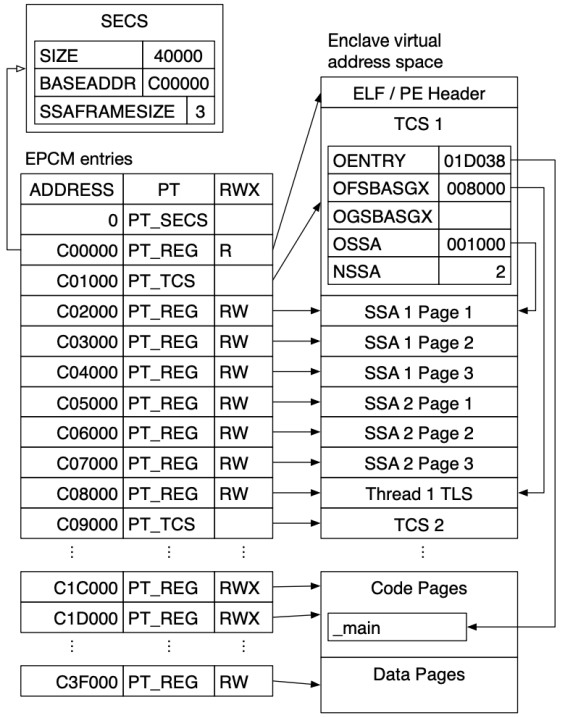
2. SGX Memory Layout | SSA

- SSA (The State Save Area)

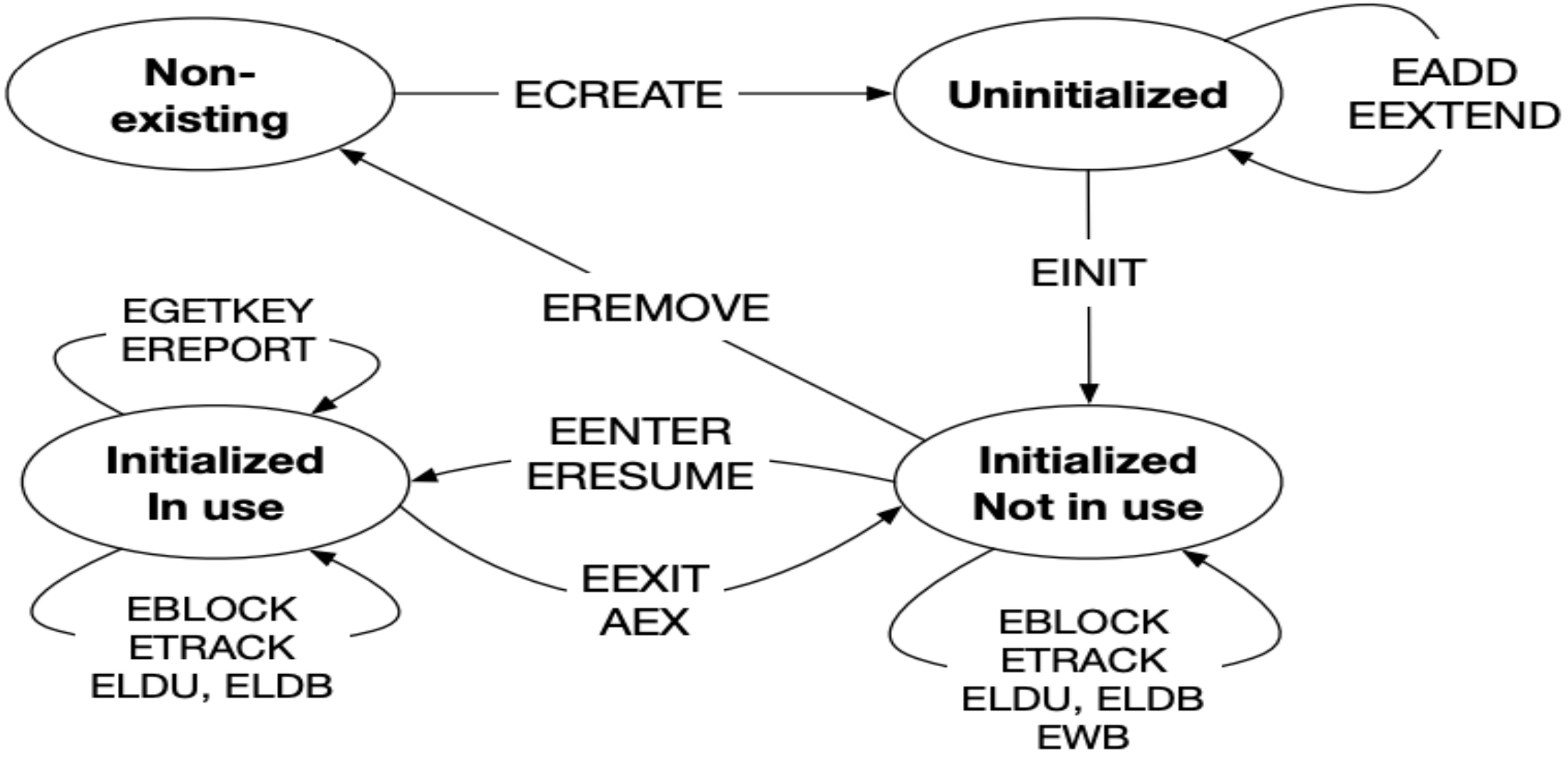
Enclave 실행 중 H/W Exception 발생 시 Thread의 Context를 저장하는 안전한 영역
(Enclave 데이터는 비밀이기 때문에 외부에 노출되는 곳에 저장할 수 없음)
OSSA(Offset of the SSA Array) -> SSA Array의 첫번째 주소
NSSA(The Number of SSAs) -> 사용 가능한 SSA의 수
EPCM의 PT field에 PT_REG 로 저장

※ Thread's Execution Context
범용 레지스터 및 XSAVE 명령어 결과

※ XSAVE
requested-feature bitmap(RFBM)을 메모리에 저장



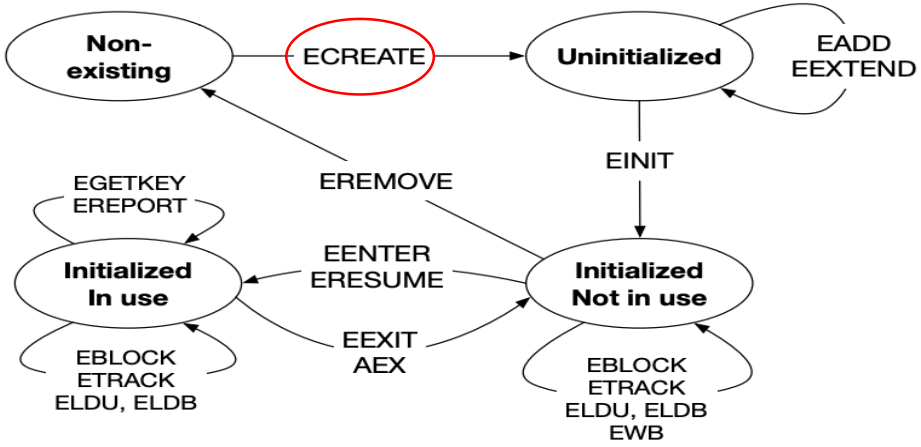
3. SGX Enclave Life Cycle



3. SGX Enclave Life Cycle | ECREATE

- ECREATE

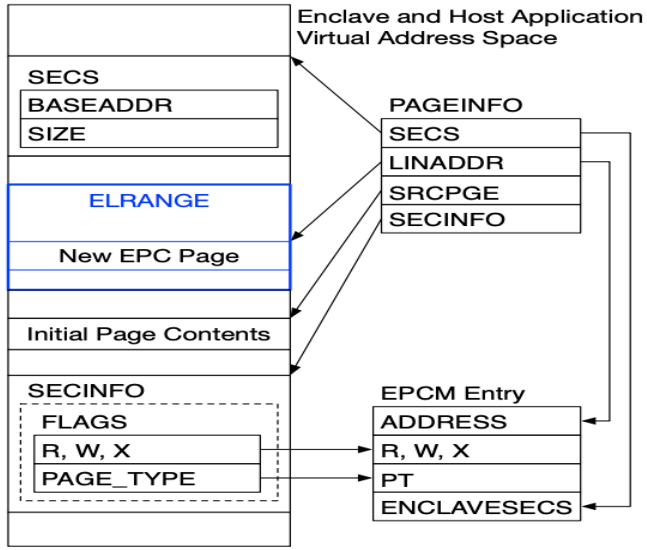
System S/W가 ECREATE 실행 시 Enclave 생성
Free EPC Page를 new Enclave의 SECS로 변환
non-EPC Page 정보를 이용해서 SECS field를 초기화
Architecture Layout을 사용하여 SECS field 값 지정(SIZE, BASEADDR)
생성된 EPC Page 유효성 검사 (2^n)
SECE->ATTRIBUTES->INIT 값을 false로 설정

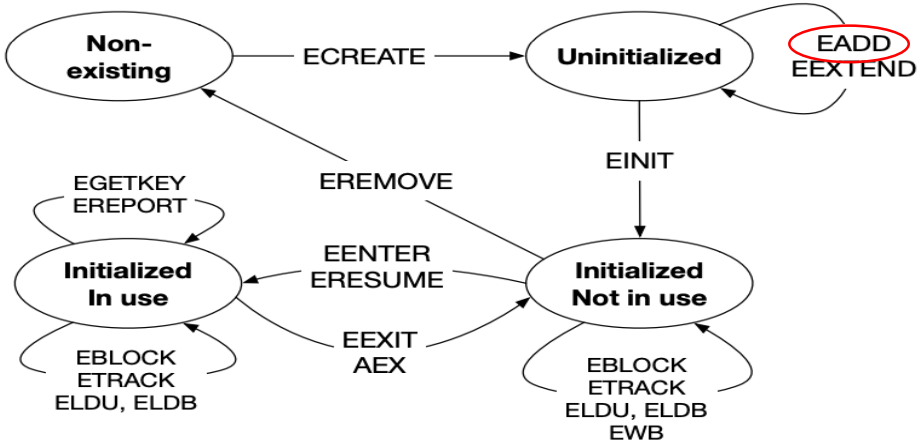


3. SGX Enclave Life Cycle | EADD

- EADD

초기 Enclave Code & Data 로드
PAGEINFO 구조에서 입력 Data를 읽어옴
입력값 검증
SECS가 이미 존재할 때 EADD 사용하면 fault
이미 할당된 EPC Page에 EADD 사용시 fault
Page의 가상주소가 ELRANGE내에 있는지 확인





LINADDR : 할당된 EPC Page 가상 주소
SRCPGE : 내용을 EPC Page로 복사할 non-EPC Page 가상 주소
SECS : Page를 소유한 Enclave의 SECS 가상 주소
SECINFO : EPCM 의 가상 주소

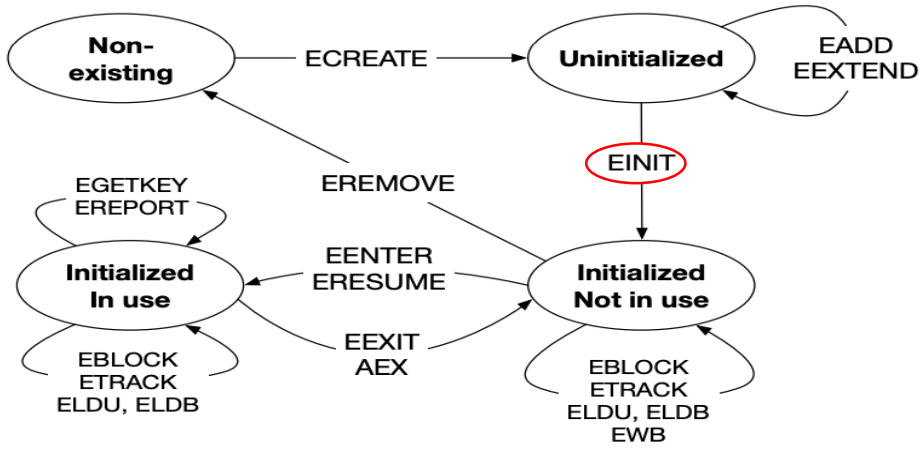
3. SGX Enclave Life Cycle | EINIT

- EINIT

Enclave 초기화
EINIT 명령어 끝난 뒤 SECE->ATTRIBUTES->INIT 값을 true로 설정
INIT이 true 설정된 후 EADD 호출 X
EINIT에 필요한 TOKEN을 Launch Enclave에서 받아야함
Measurement 값 확정 (변경 불가)

Launch Enclave(LE)

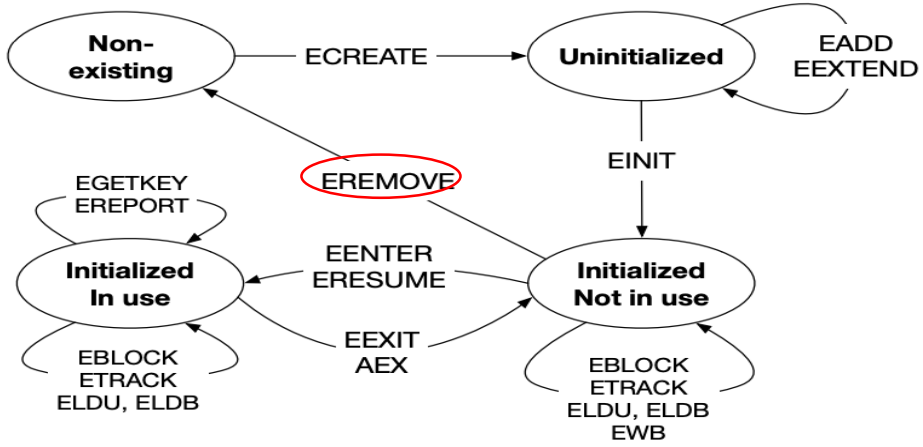
Intel에서 제공하는 Privileged Enclave
하드코딩된 특수 Key를 이용하여 서명



3. SGX Enclave Life Cycle | EREMOVE

- EREMOVE

사용된 EPC Page 할당 해제
EPCM->VALID 값 0으로 설정
Enclave를 실행하는 Thread 확인 후 없으면 SECS 할당 해제를 끝으로 해제



3. SGX Enclave Life Cycle | EENTER

- EENTER

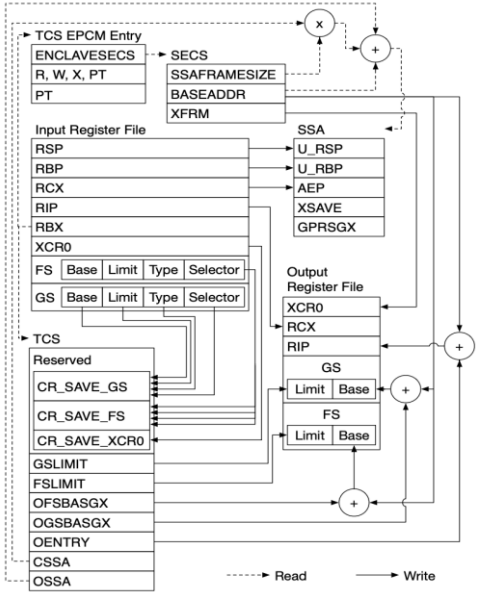
Enclave 코드에 제어된 Jump 실행
Enclave Mode로 전환

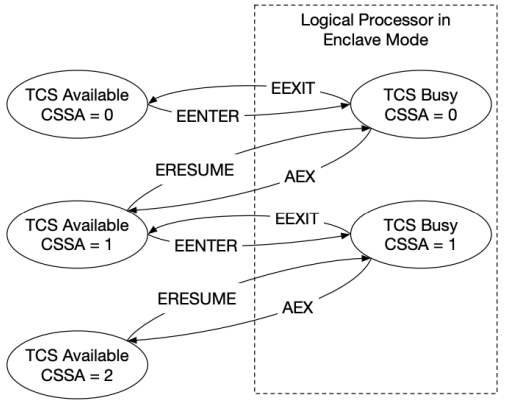
ring 0가 아닌 ring 3의 Enclave Mode

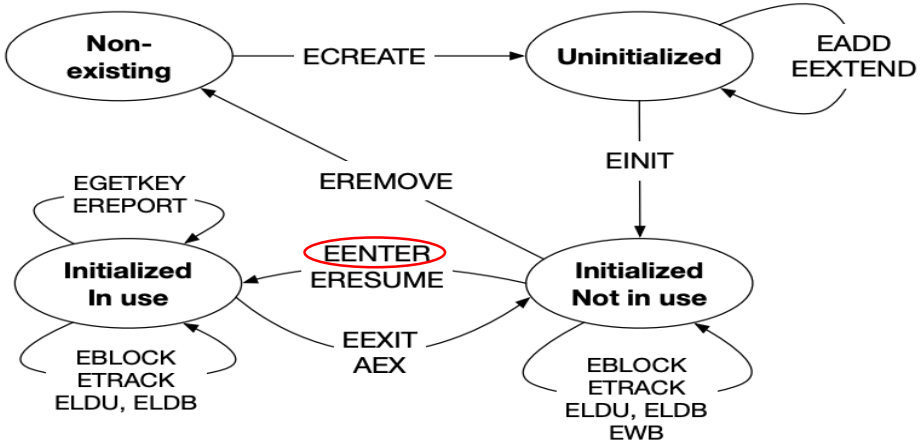
TCS를 입력으로 받고, TCS가 not busy, NSSA가 1개 이상
RIP를 OENTRY로 설정(Entry Point)
XFRM로 XCR0 설정

Architecture feature 제어

FS, GS를 이용해서 TLS 구현
EENTER 다음 명령어를 RCX 저장
레지스터 이전 값 백업 (DRAM??)





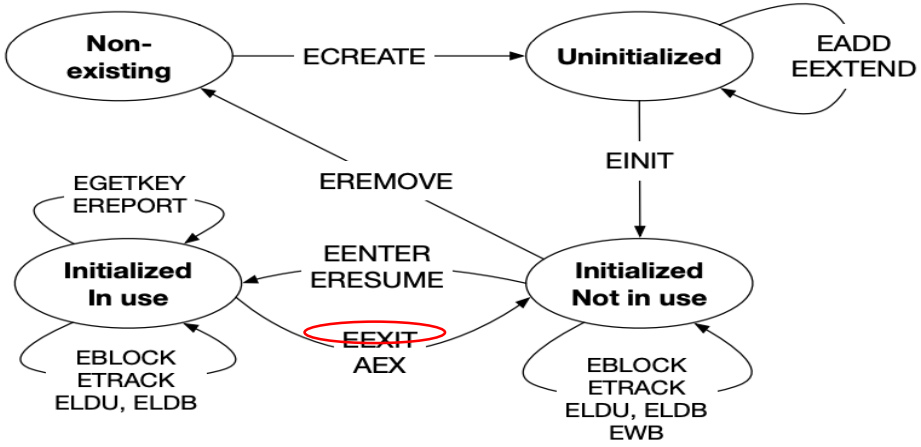


3. SGX Enclave Life Cycle | EEXIT

- EEXIT

Enclave Mode일 경우만 실행
RBX에서 읽은 값으로 RIP(명령어 레지스터) 설정
EENTER 수행 시 저장한 레지스터 값 불러옴

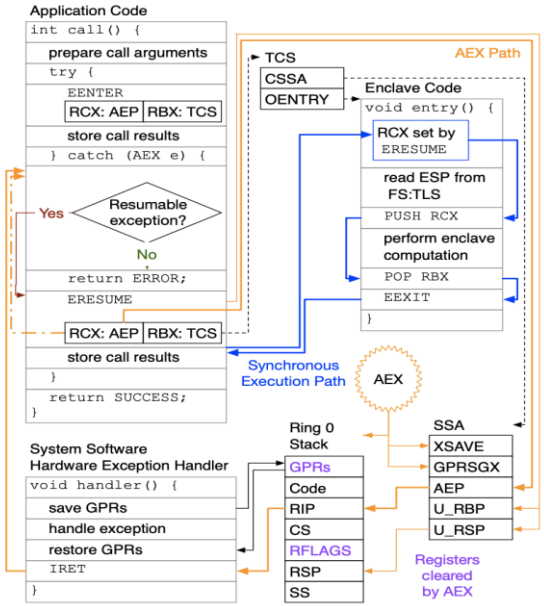
!! EEXIT은 레지스터 수정을 따로 안하기 때문에 EENTER가 끝날 때 레지스터에 남은 비밀 제거해야함 !!



3. SGX Enclave Life Cycle | AEX

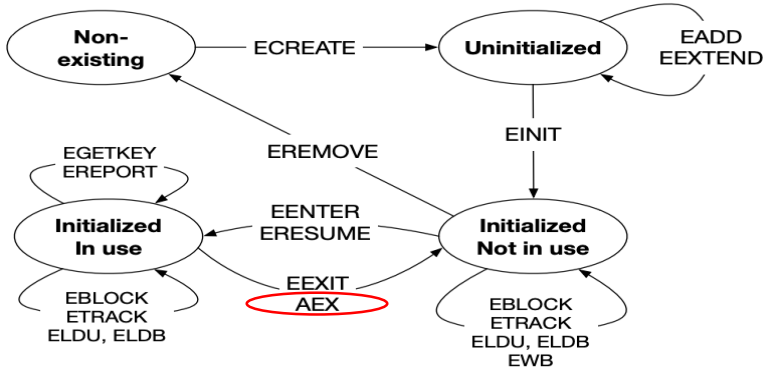
- AEX (Asynchronous Enclave Exit)

Enclave Mode 중 H/W Exception 발생 시 바로 Handler 호출 X
AEX를 이용하여 Enclave Execution Context 저장 (SSA)
SSA에 저장 후 남은 비밀들 삭제 후 Enclave Mode 종료



EENTER 명령어 수행 시

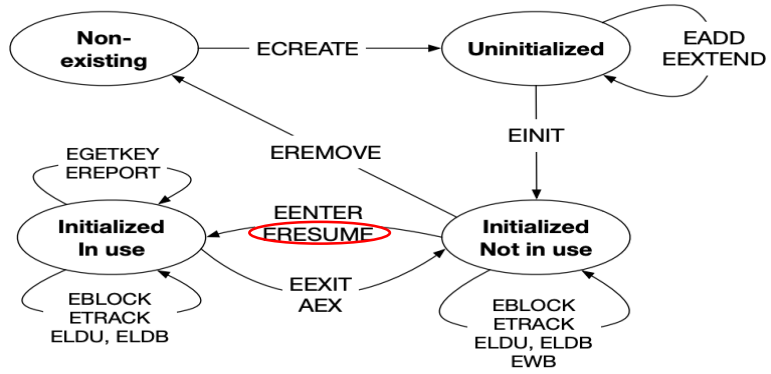
H/W Exception Handler가 Asynchronous Exit Handler로 return
AEP(handler Pointer)를 RCX 값에 저장 후 SSA AEP로 전달
RSP, RBP를 SSA에 저장



3. SGX Enclave Life Cycle | ERESUME

- ERESUME

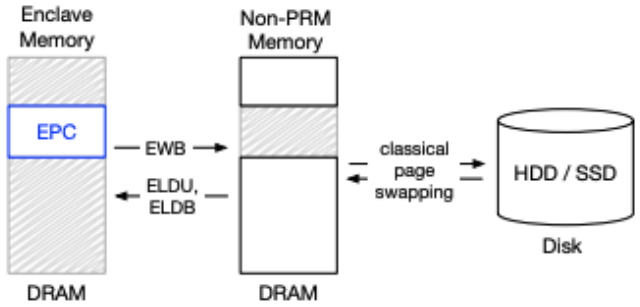
H/W Exception 실행 후 AEP에 저장한 Asynchronous exit handler로 분기
Asynchronous exit handler는 ERESUME 실행
Enclave Mode로 전환 및 중단된 계산 다시 시작
SSA에 백업된 Execution Context 복원
EENTER와 ERESUME 명령은 SSA의 내용 유무를 제외하고 동일한 메커니즘
EENTER -> CSSA < NNSA
ERESUME -> CSSA != 0



4. EPC Page Eviction | Page Swapping, Eviction

- Page Swapping (Paging)

  OS Kernel이 비활성화 어플리케이션의 메모리를 Disk로 이동
  부족한 DRAM 자원을 효과적으로 사용 가능한 메커니즘



- EPC Page Eviction

  OS와 동일한 방법으로 Paging 불가능(EPC는 보호되는 데이터이기 때문)
  PRM 범위의 EPC Page를 PRM 외부로 evict 후 OS의 Paging 메커니즘 사용

4. EPC Page Eviction | TLB

- TLB (Translation Lookaside Buffer)

  가상 주소를 물리 주소로 변환하는 과정에서 빠른 변환을 위한 캐쉬
  최근 일어난 주소 변환 테이블을 저장
  CPU는 1차적으로 TLB 확인 후 없으면 MMU 확인
  Page eviction시 TLB를 이용한 EPC 접근 공격 방지 메커니즘 존재

  TLB를 이용한 EPC 접근 공격 방어 메커니즘
    - EEXIT, AEX로 Enclave 종료될 때 TLB flush
    - EPC Page가 Enclave에서 해제될 때 Enclave 코드 사용하는 모든 Thread가 Enclave 종료

4. EPC Page Eviction | TLB 이용 공격 방지

- TLB Flush

  EEIXT, AEX를 통해 Enclave 종료할 때 TLB Flush

  System S/W가 Flush -> SGX가 검증

  주소변환 결과가 Blocked인 경우 Page fault

  EBLOCK -> Page table에서 EPC Page 매핑 제거(OS 가 수행)

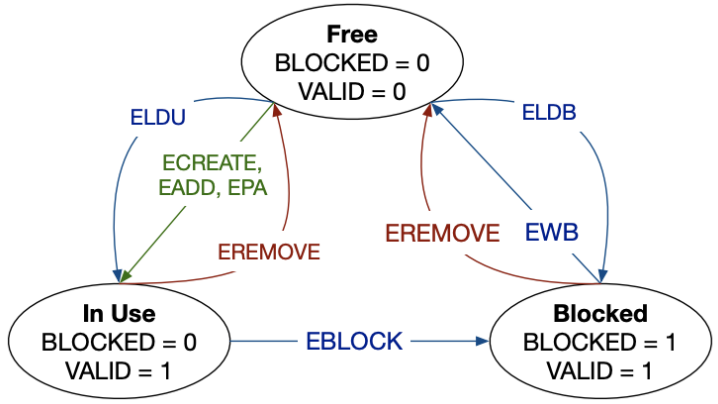
  ETRACK -> 어떤 Thread가 Flush 한지 추적 및 AEX 발생(IPI)

  EWB -> EPC Page Evict 수행

  EPC Page 암호화 버전 작성

  VALID, BLOCKED 비트 삭제

  ELDU, ELDB -> Evict된 Page 복구(B -> BLOCKED 1)



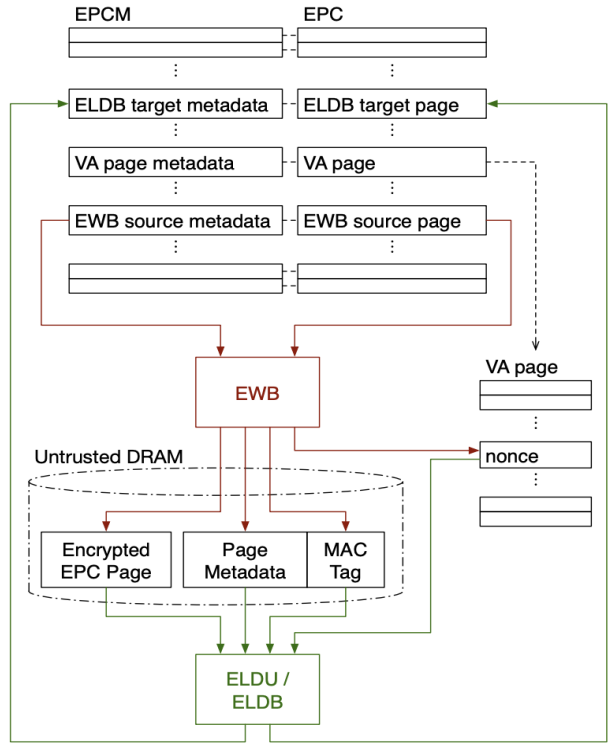
4. EPC Page Eviction | VA & Enclave ID

- The Version Array (VA)
  - SGX는 Evict된 데이터의 Freshness 보장하기 위해 Nonce 사용 (EWB)
  - 해당 Nonce 값을 EPC Page에 저장 (PT->PT_VA)
  - VA Page 제한 없이 할당 취소 가능하지만 취소 시 Evict 복원 불가

- Enclave ID
  - Evict된 Enclave를 식별하기 위한 ID
  - SECS같은 경우 EPC에 남아 있는 경우만 식별 가능

4. EPC Page Eviction | EWB & ELDU,ELDB

- EWB
  - EWB 출력은 암호화된 EPC Page version, Page metadata, MAC tag, Nonce
  - PRM 외부로 evict 후 OS에 의해 Disk로 이동 가능
  - EPC Page 내용 제외 나머지는 암호화 되지 않음
  - MAC 은 나머지 데이터의 진위성 검사
- ELDU, ELDB
  - Evict된 Enclave Page 복원
  - 두 명령어의 차이는 B는 BLOCKED를 1로 set



5. SGX Enclave Measurement

- Measurement

  Enclave의 Measurement를 이용하여 자기 자신을 인증
  Enclave가 생성되면서 정해진 방식에 의해 Measurement 생성
  예상되는 Measurement 값과 실제 Measurement가 같으면 인증
  SHA-2 Hash 알고리즘 사용하여 Measurement 생성
  중간에 값이 하나만 수정돼도 완전히 다른 Hash값 출력
  SECS의 MRENCLAVE field에 저장

5. SGX Enclave Measurement | ECREATE

- Measuring ECREATE

SECS->MRENCLAVE field 초기화
정해진 64byte 데이터를 SHA-2 알고리즘으로 Hashing
SSAFRAMSIZE -> 최대 사용할 수 있는 SSA 개수
BASEADDR이 없는 이유는 ELRANGE 내부 메모리 어디든 로드 가능하기 때문

Offset	Size	Description
0	8	"ECREATE\0"
8	8	SECS.SSAFRAMESIZE (§ 5.2.5)
16	8	SECS.SIZE (§ 5.2.1)
32	8	32 zero (0) bytes

Enclave Attribute

Measurement에는 Attribute 값이 사용되지 않음
단일 Measurement를 이용하여 아키텍쳐 확장의 이점 (XFRM의 feature 별로 만들 수 없음..?)

5. SGX Enclave Measurement | EADD

- Measuring EADD

Offset	Size	Description
0	8	"EADD\0\0\0\0"
8	8	ENCLAVEOFFSET
16	48	SECINFO (first 48 bytes)

ENCLAVEOFFSET -> EPC Page의 예상되는 가상 주소
(작성자의 가상 메모리 레이아웃 -> 변환 방지)
SECINFO -> EPCM의 PT, R,W,X field 값 포함 (47바이트는 0)

- Measuring EEXTEND

Offset	Size	Description
0	8	"EEXTEND\0"
8	8	ENCLAVEOFFSET
16	48	48 zero (0) bytes
64	64	bytes 0 - 64 in the chunk
128	64	bytes 64 - 128 in the chunk
192	64	bytes 128 - 192 in the chunk
256	64	bytes 192 - 256 in the chunk

Enclave EPC Page에 로드된 Data를 Measurement
EEXTEND로 Measurement하지 않은 Enclave는 EENTER 불가
EADD와 분리한 이유는 지연시간 때문

5. SGX Enclave Measurement | EINIT

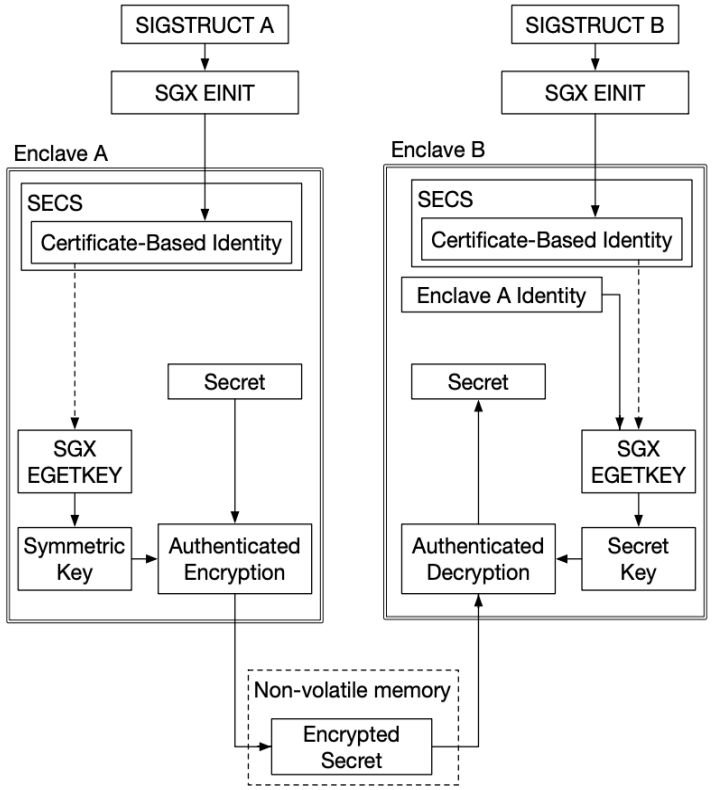
- Measuring EINIT

  Enclave 생성 과정 마무리
  Enclave 모든 내용을 Sealing
  MRENCLAVE에 SHA2 최종화 알고리즘 출력 저장

6. SGX Enclave Versioning Support

- Versioning Support

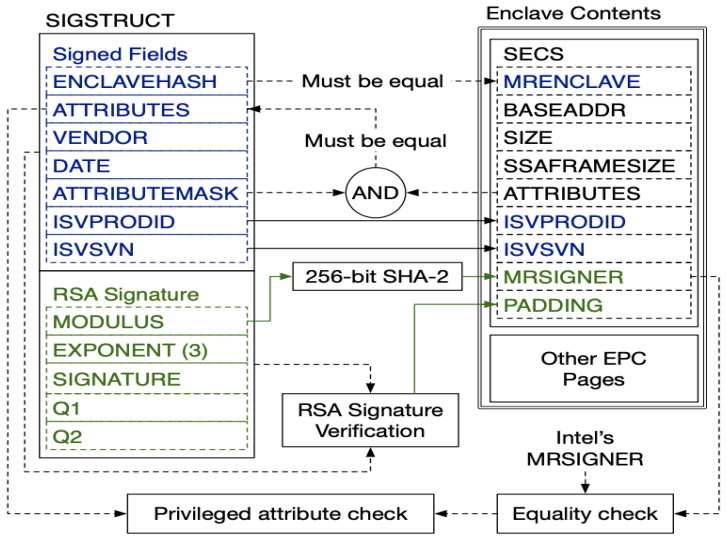
SGX는 같은 S/W에서 서로 다른 버전 간 Migration을 지원
Enclave 마다 작성자가 발급한 인증서 발급(SIGSTRUCT)
SIGSTRUCT 검토 후 Certificate-Based Identity 값 설정
EGETKEY로 KEY 입수 및 암호화 후 System S/W 양도



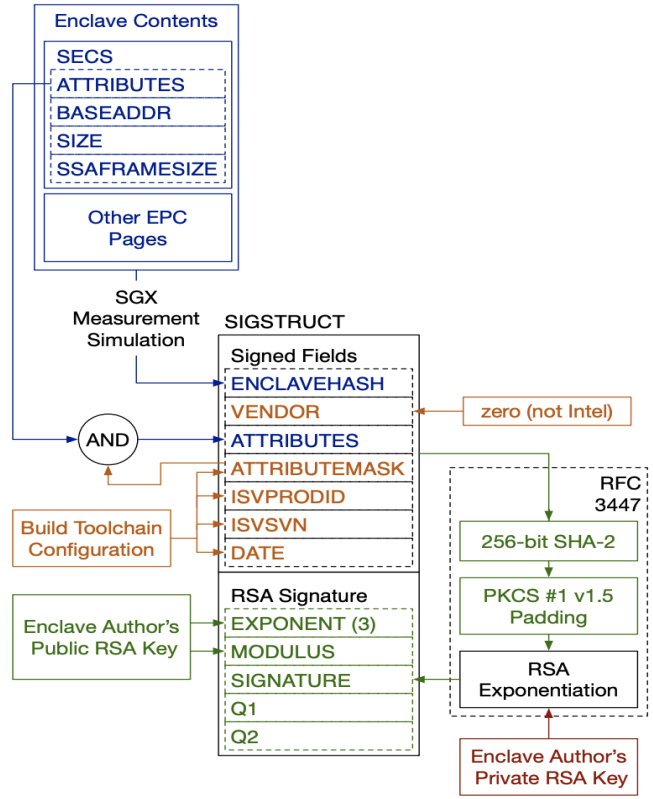
6. SGX Enclave Versioning Support | SIGSTRUCT

- SIGSTRUCT

SGX는 Enclave 마다 인증서를 갖도록 요구(EINIT 수행 시 검증)
SECS Metadata 및 Enclave building toolchain로 값 설정
Metadata가 저장 및 RSA 서명으로 신뢰성 보장
공개 지수 3, 3072bit RSA키 지원



Field	Bytes	Description
ENCLAVEHASH	32	Must equal the enclave's measurement (§ 5.6).
ISVPRODID	32	Differentiates modules signed by the same private key.
ISVSVN	32	Differentiates versions of the same module.
VENDOR	4	Differentiates Intel enclaves.
ATTRIBUTES	16	Constrains the enclave's attributes.
ATTRIBUTEMASK	16	Constrains the enclave's attributes.



6. SGX Enclave Versioning Support | Certificate-Based Enclave Identity

- Certificate-Based Enclave Identity

MODULUS, ISVPRODID, ISVSVN 3개의 필드로 결정

MODULUS -> SIGSTRUCT의 서명 RSA 키 계수
MRSIGNER -> MODULUS의 256 SHA-2 Hash 값
ISVPRODID -> Enclave 제품 ID
ISVSVN -> 보안 버전 넘버

동일 RSA 키로 인증서 발급 -> ISVPRODID 로 구분 (다른 소프트웨어 모듈)
동일 RSA 키 + 동일 ISVPRODID -> 동일 소프트웨어 모듈 (버전(ISVSVN) 다름)

높은 Version에서 낮은 Version으로 Migration 불가능

6. SGX Enclave Versioning Support | Key Derivation

- Enclave Key Derivation

EGETKEY -> 정해진 Metadata의 조합 + CR_SEAL_FUSES
동일한 값을 이용하면 CPU 전원 주기 전반에 같은 키 출력

KEYREQUEST Structure + SECS 이용하여 Key Derivation

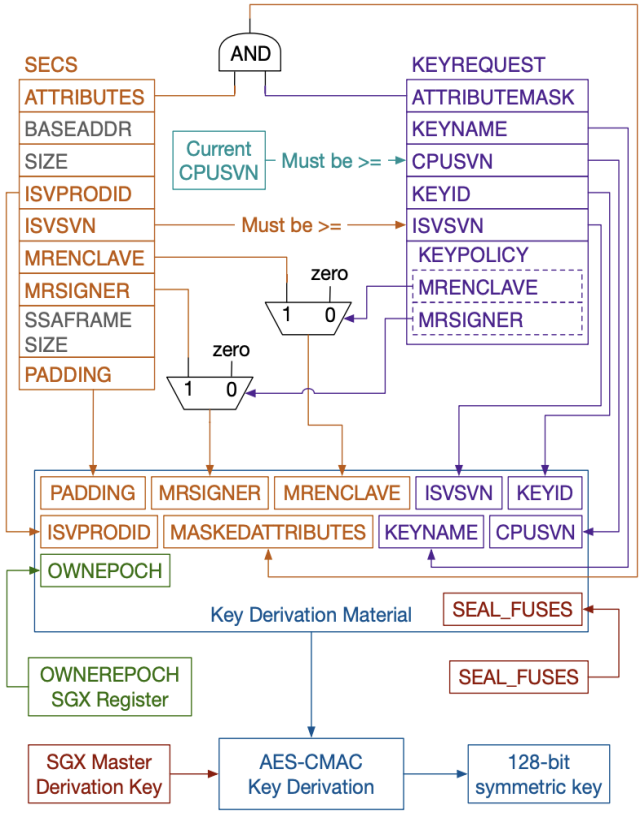
KEYREQUEST

KEYNAME -> 생성될 키 종류

KEYPLICY -> MRENCLAE, MRSIGER 사용

KEYID -> 무작위 번호 (키 마모 방지?)

Field	Bytes	Description
KEYNAME	2	The desired key type; secret migration uses Seal keys
KEYPOLICY	2	The identity information (MRENCLAVE and/or MRSIGNER)
ISVSVN	2	The enclave SVN used in derivation
CPUSVN	16	SGX implementation SVN used in derivation
ATTRIBUTEMASK	16	Selects enclave attributes
KEYID	32	Random bytes



7. SGX Software Attestation | Attestation

- Attestation

  Enclave가 정보를 주고 받기 위해 자신을 입증하는 과정

  같은 SGX 지원 CPU에서 호스팅 되는 다른 Enclave와 신뢰 관계 구축

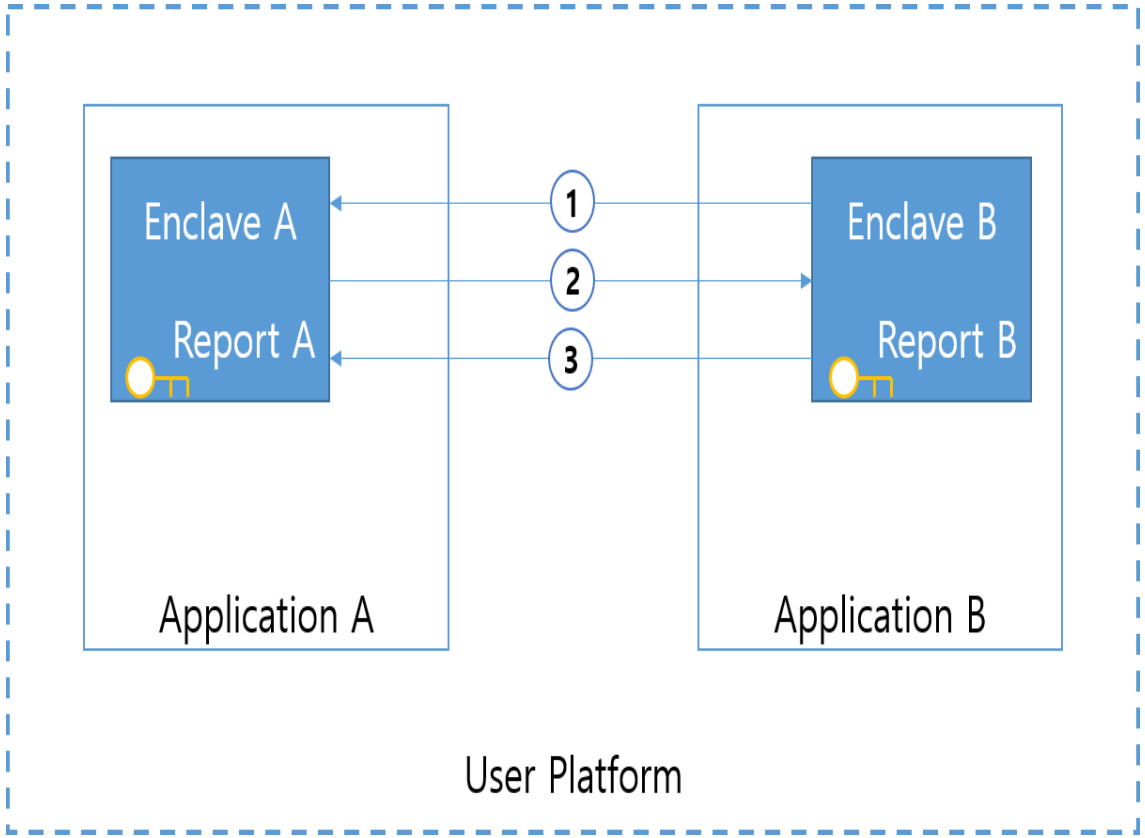
  Local Attestation

  Enclave가 원거리의 다른 제 3자와 서비스 하기 위해 신뢰 관계 구축

  Remote Attestation

  Quoting Enclave -> Remote Attestation 에서 증명 서명에 사용되는 intel이 발행한 특수 Enclave

7. SGX Software Attestation | Local Attestation



1. A, B 사이에 통신 채널 형성 후 B의 MRENCLAVE 및 Attributes 값 전송

2. A는 EREPORT 명령어를 이용해서 B에게 서명된 REPORT 전송

3. B는 EGETKEY 명령어를 사용하여 Report Key를 얻고 해당 키로 MAC 값을 확인하여 입증

7. SGX Software Attestation | Local Attestation

- EREPORT

Enclave SECS에서 필요한 정보 EREPORT 구조에 채움
(ATTRIBUTES, MRENCLAVE, Certificate-Based ID)
EGETKEY로 Report 키 생성 후 해당 키로 MAC 생성
Target Enclave의 MRENCLAVE, ATTRIBUTE 이용

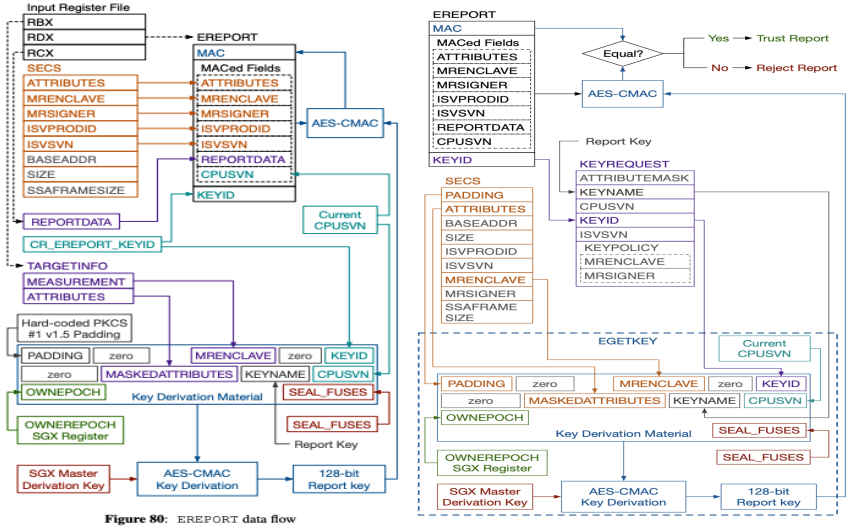
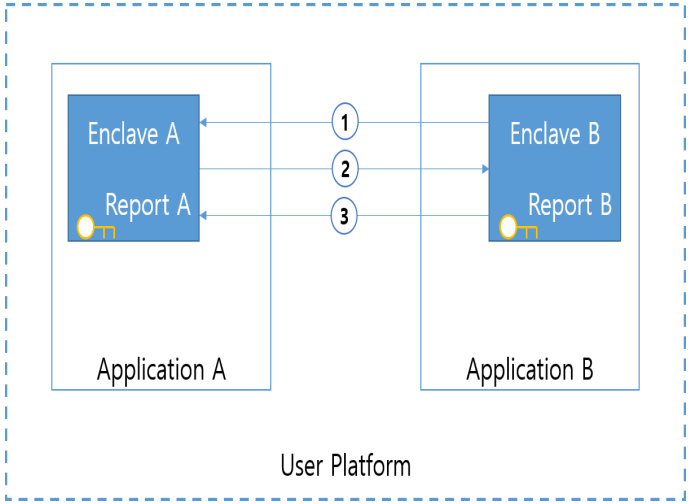


Figure 80: EREPORT data flow

7. SGX Software Attestation | Local Attestation



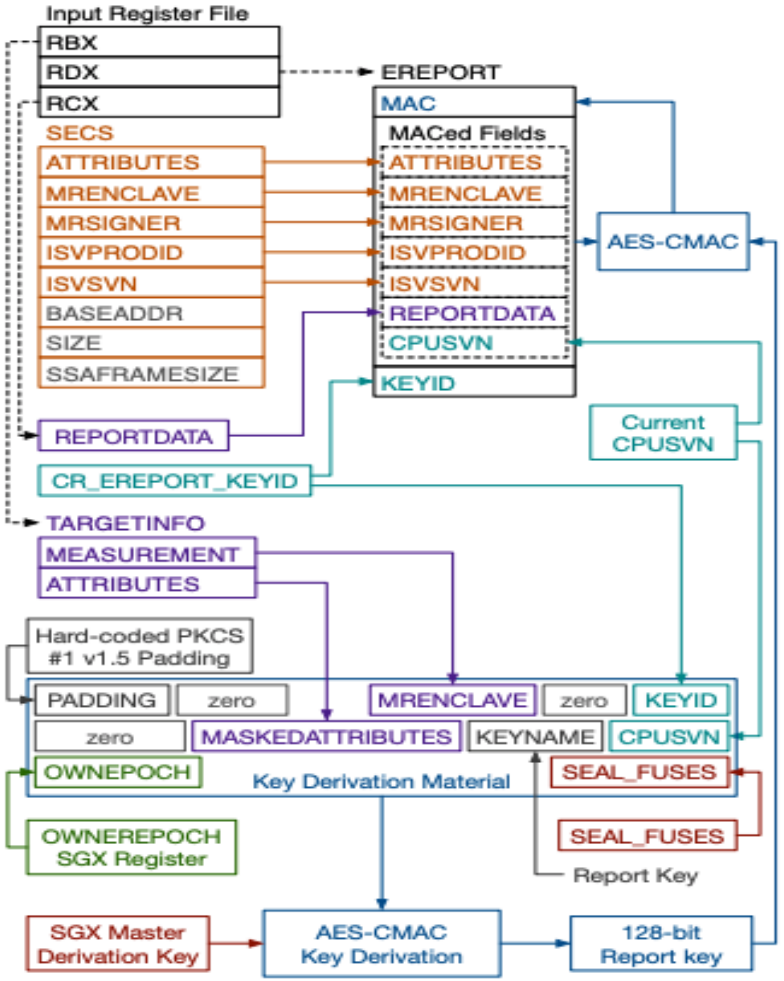
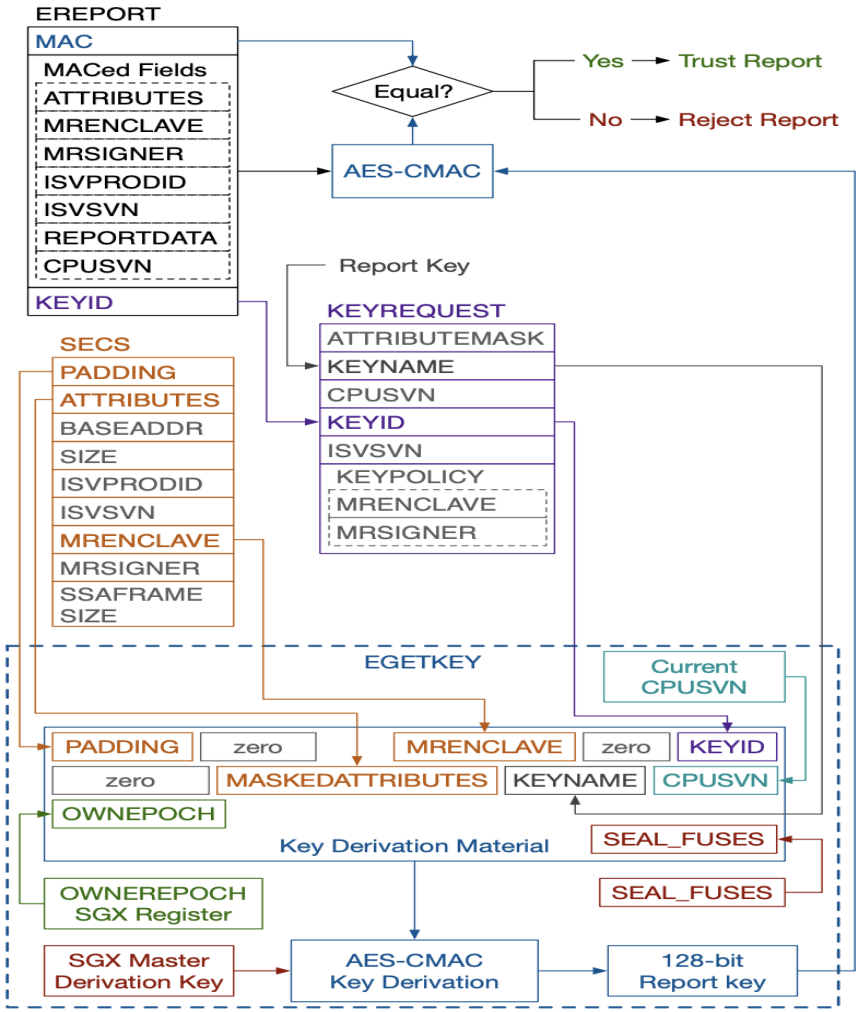


Figure 80: EREPORT data flow



7. SGX Software Attestation | Remote Attestation

- Provisioning Secret

  EGETKEY 과정에서 사용하는 SGX Master Derivation Key를 출력할 때 사용
  Intel Key Generation Facility 에서 생성되서 Intel Provisioning Service DB 저장
  Provisioning Key

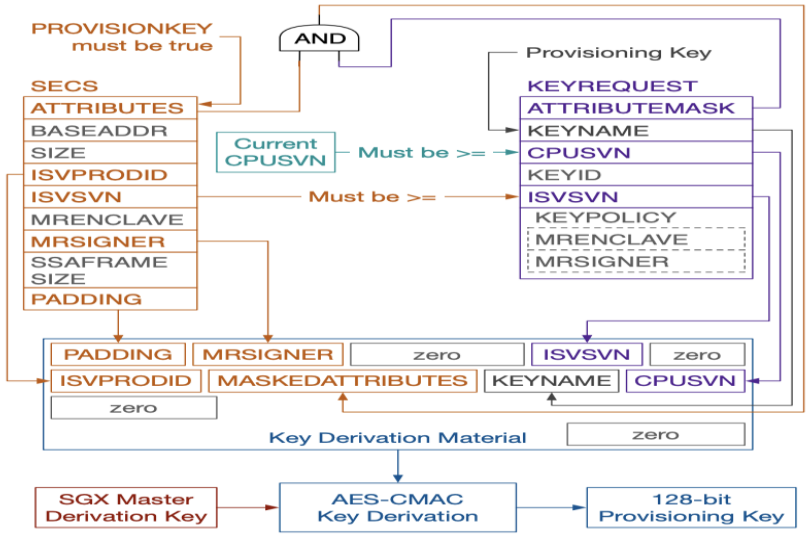
- Seal Secret

  EGETKEY 유도 과정에서 사용되는 CR_SEAL_FUSES
  칩 내부에서 생성되는 비밀로 Intel은 알지 못함
  Seal Key, Report Key, Provisioning Seal Key

7. SGX Software Attestation | Provisioning Key & Seal

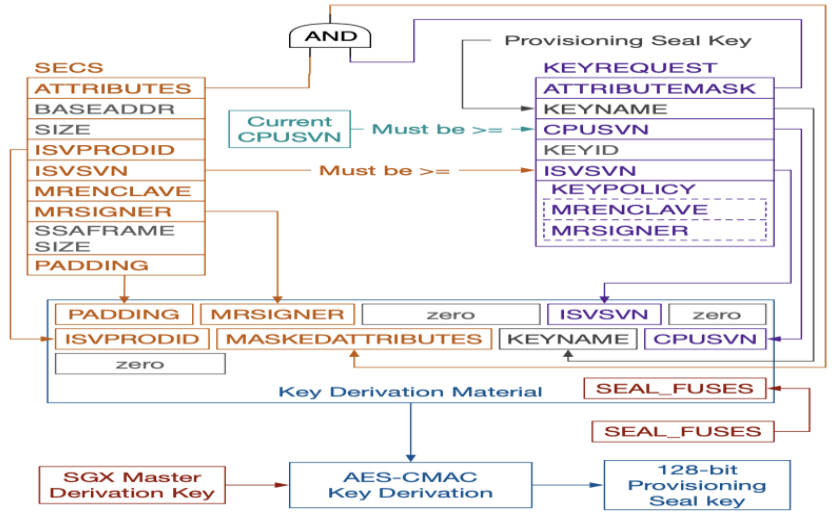
- Provisioning Key

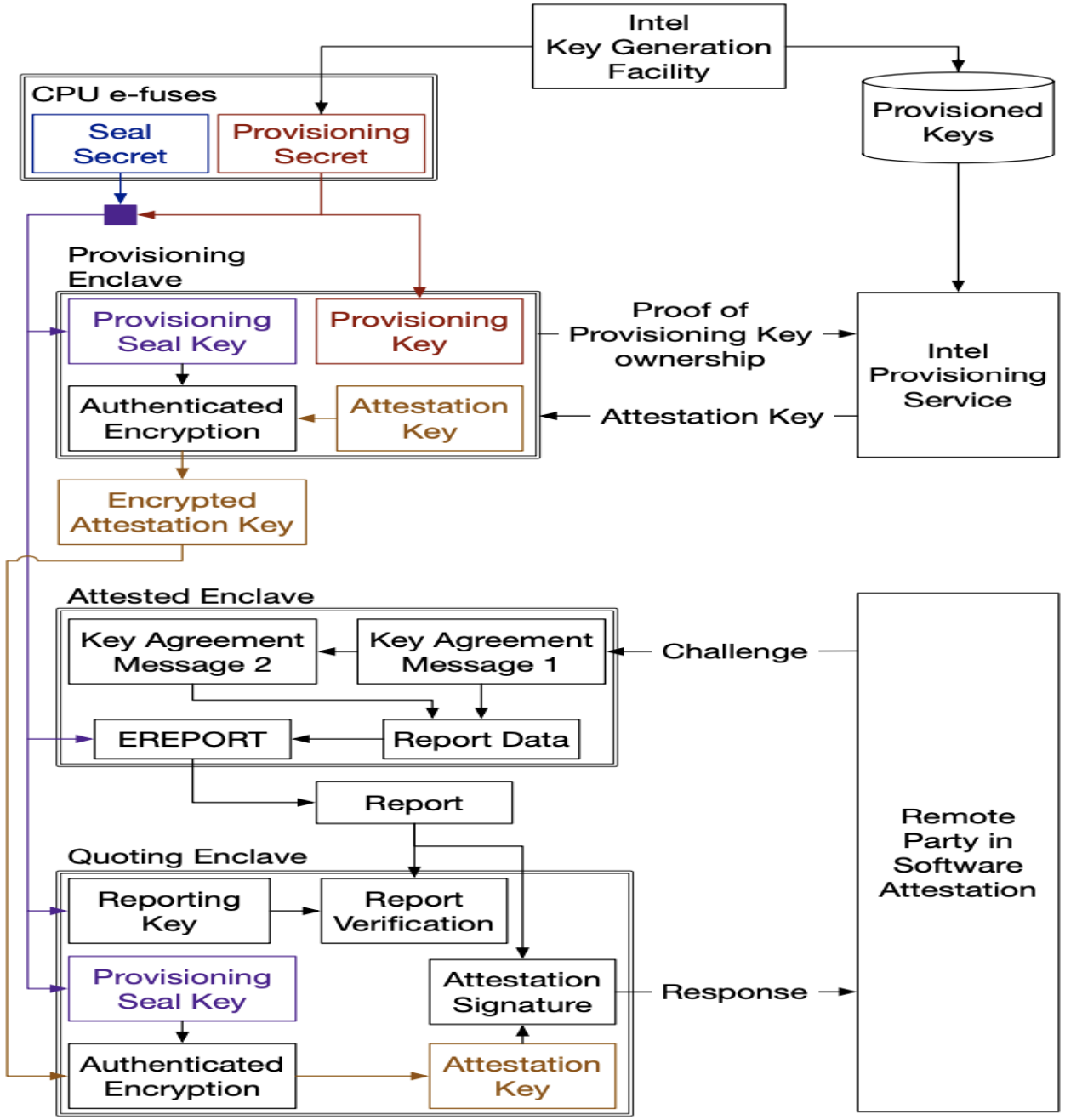
  Seal Secret, OWNERPOCH 값 포함 X
  Certificate-Based ID 및 CPUSVN 이용해서 키 유도
  Intel Provisioning Service와 인증할 때 사용



- Provisioning Seal Key

  Certificate-Based ID 및 CPUSVN 이용해서 키 유도
  Seal Secret, OWNERPOCH 값 포함
  Intel Service와 인증키 암호화 할 때 사용




Intel
Key Generation
Facility
Provisioned
Keys
CPU e-fuses
Seal
Secret
Provisioning
Secret
Provisioning
Enclave
Provisioning
Seal Key
Provisioning
Key
Proof of
Provisioning Key
ownership
Intel
Provisioning
Service
Authenticated
Encryption
Attestation
Key
Attestation Key
Encrypted
Attestation Key
Attested Enclave
Key Agreement
Message 2
Key Agreement
Message 1
Challenge
EREPORT
Report Data
Report
Quoting Enclave
Reporting
Key
Report
Verification
Remote
Party in
Software
Attestation
Provisioning
Seal Key
Attestation
Signature
Response
Authenticated
Encryption
Attestation
Key

Q & A