Overview of Intel® Software Guard Extensions Instructions and Data Structure

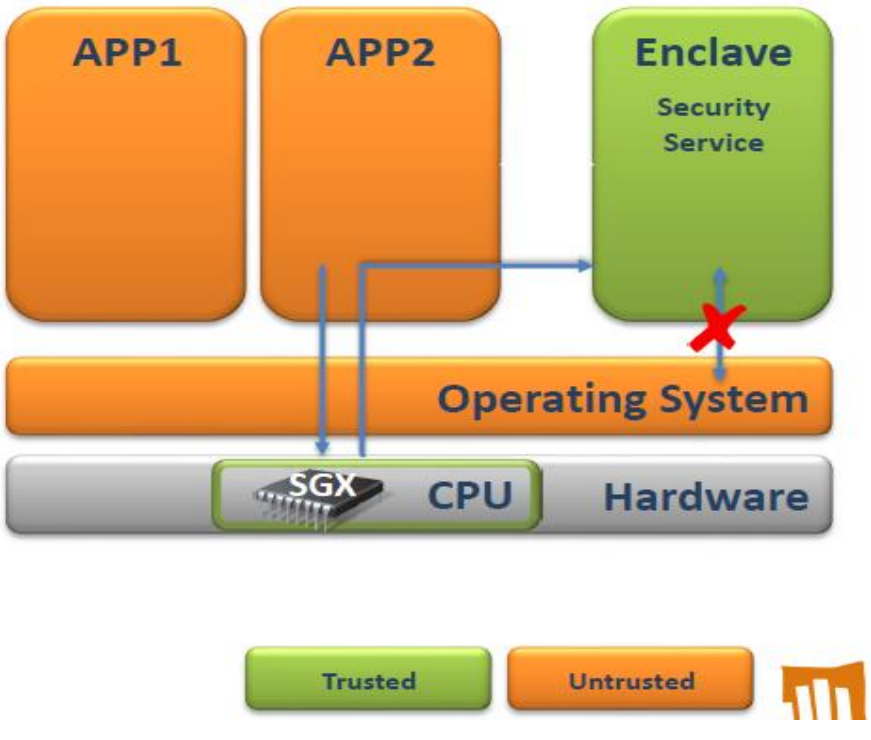
This post is intended to introduce developers to know about Intel SGX Instructions and Data Structure. There are 18 types of Instructions and 13 Data Structures available in SGX.

In this blog we are going to learn below topics

1. Intel Software Guard Extension.
2. Basics -Ring Separation.
3. SGX Instructions and Data Structures.
4. SGX Memory Access Control.
5. SGX Control Structure Access.

Intel® Software Guard Extensions (SGX):

- **Security critical code isolated in enclave**
- **Only CPU is trusted**
  - Transparent memory encryption
  - 18 new instructions
- **Enclaves cannot harm the system**
  - Only unprivileged code (CPU ring3)
  - Memory protection
- **Designed for Multi-Core systems**
  - Multi-threaded execution of enclaves
  - Parallel execution of enclaves and untrusted code
  - Enclaves are interruptible
- **Programming Reference available**



Basics -Ring Separation:

- **4 different privilege levels**
    1. **0 = Kernel**
    2. **1,2 = Device Driver**
    3. **3 = Applications**
- **Current Privilege Level maintained by the CPU in the Code Segment Register CS**

SGX Enclaves:

- **Enclaves are isolated memory regions of code and data**
- **One part of physical memory (RAM) is reserved for enclaves**
    - It is called **Enclave Page Cache (EPC)**
    - EPC memory is encrypted in the main memory (RAM)
    - Trusted hardware consists of the CPU-Die only
    - EPC is managed by OS/VMM

SGX Instructions and Data Structures:

- **18 Instruction**
    - 13 Supervisor Instructions
    - 5 User Instructions

- **13 Data Structures**
    - 8 data structures associated to a certain enclave
    - 3 data structures associated to certain memory page(s)
    - 2 data structures associated to overall resource management

13 Supervisor SGX Instructions:

Supervisor Instruction	**Description**
ENCLS[EADD]	Add a page
ENCLS[EBLOCK]	Block an EPC page
ENCLS[ECREATE]	Create an enclave
ENCLS[EDBGRD]	Read data by debugger
ENCLS[EDBGWR]	Write data by debugger
ENCLS[EEXTEND]	Extend EPC page measurement
ENCLS[EINIT]	Initialize an enclave
ENCLS[ELDB]	Load an EPC page as blocked
ENCLS[ELDU]	Load an EPC page as unblocked
ENCLS[EPA]	Add version array
ENCLS[EREMOVE]	Remove a page from EPC
ENCLS[ETRACK]	Activate EBLOCK checks
ENCLS[EWB]	Write back/invalidate an EPC page

5 User SGX Instructions:

User Instruction	Description
ENCLU[EENTER]	Enter an Enclave
ENCLU[EEXIT]	Exit an Enclave
ENCLU[EGETKEY]	Create a cryptographic key
ENCLU[EREPORT]	Create a cryptographic report
ENCLU[ERESUME]	Re-enter an Enclave

SGX Data Structures in Details:

1. **SGX Enclave Control Structure (SECS)**

   - Represents one enclave
   - Contains, for instance, Hash, ID, size etc.

2. **Thread Control Structure (TCS)**

   - Each executing thread in the enclave is associated with a Thread Control Structure
   - Contains, for instance, Entry point, pointer to SSA

3. **State Save Area (SSA)**

   - When an AEX occurs while running in an enclave, the architectural state is saved in the thread's SSA

4. **Page Information (PAGEINFO)**

   - PAGEINFO is an architectural data structure that is used as a parameter to the EPC-management instructions
     - Linear Address
     - Effective address of the page (aka virtual address)
     - SECINFO
     - SECS

5. **Security Information (SECINFO)**

   - The SECINFO data structure holds meta-data about an enclave page
     - Read/Write/Execute
     - Page type (SECS, TCS, normal page or VA)

6. **Paging Crypto MetaData (PCMD)**

- The PCMD structure is used to keep track of crypto meta-data associated with a paged-out page. Combined with PAGEINFO, it provides enough information for the processor to verify, decrypt, and reload a paged-out EPC page.
- EWB writes out (the reserved field and) MAC values.
- ELDB/U reads the fields and checks the MAC.
- Contains Enclave ID, SECINFO and MAC

7. **Version Array (VA)**

- In order to securely store the versions of evicted EPC pages, SGX defines a special EPC page type called a Version Array (VA).
    - Each VA page contains 512 slots, each of which can contain an 8-byte version number for a page evicted from the EPC.
- When an EPC page is evicted, software chooses an empty slot in a VA page; this slot receives the unique version number of the page being evicted.
- When the EPC page is reloaded, a VA slot must hold the version of the page. If the page is successfully reloaded, the version in the VA slot is cleared.
- VA pages can be evicted, just like any other EPC page.
    - When evicting a VA page, a version slot in some other VA page must be used to receive the version for the VA being evicted.

8. **Enclave Page Cache Map (EPCM)**

- EPCM is a secure structure used by the processor to track the contents of the EPC. The EPCM holds exactly one entry for each page that is currently loaded into the EPC. EPCM is not accessible by software, and the layout of EPCM fields are implementation specific.
- Contains, for instance, RWX, page type, linear address, state etc.

9. **Enclave Signature Structure (SIGSTRUCT)**

- SIGSTRUCT contains information about the enclave from the enclave signer
- SIGSTRUCT includes ENCLAVEHASH as SHA256
- SIGSTRUCT includes four 3072-bit integers (MODULUS, SIGNATURE, Q1, Q2)

10. **EINT Token Structure (EINITTOKEN)**

- The EINIT token is used by EINIT to verify that the enclave is permitted to launch
- Contains, for instance, attributes, hash and signer of the enclave
- Authenticated with a cryptographic MAC on EINITTOKEN using Launch key

11. Report (REPORT)

- The REPORT structure is the output of the EREPORT instruction
  - Attributes of the enclave
  - Hash of the enclave
  - Signer of the enclave
  - A set of data used for communication between the enclave and the target enclave
  - A CMAC on the report using report key

12. Report Target Info (TARGETINFO)

- This structure is an input parameter to the EREPORT instruction. It is used to identify the enclave which will be able to cryptographically verify the REPORT structure returned by EREPORT
- Contains attributes and hash of target enclave

13. Key Request (KEYREQUEST)

- This structure is an input parameter to the EGETKEY instruction.
- It is used for selecting the appropriate key and any additional parameters required in the derivation of that key.

SGX Memory Access Control:

Access control in two direction

1. **From enclaves to "outside"**

   - Isolating malicious enclaves
   - Enclaves needs some means to communicate with the outside world, e.g., their "host applications"

2. **From "outside" to enclaves**

   - Enclave memory must be protected from
   - Applications
   - Privileged software (OS/ Virtual Machine Monitor (also known as Hypervisor))
   - Other enclaves

SGX MAC from enclaves to "outside":

From enclaves to "outside"

- All memory access has to conform to segmentation and paging policies by the OS/VMM
- Enclaves cannot manipulate those policies, only unprivileged instructions inside an enclave
- Code fetches from inside an enclave to a linear address outside that enclave will results in a General Protection Fault (0)exception

SGX MAC "outside" to enclaves:

From "outside" to enclaves

- Non-enclave accesses to EPC memory results in abort page semantics
- Direct jumps from outside to any linear address that maps to an enclave do not enable enclave mode and result in a about page semantics and undefined behavior
- Hardware detects and prevents enclave accesses using logical-to-linear address translations which are different than the original direct EA used to allocate the page. Detection of modified translation results in General Protection Fault (0)

SGX Control Structure Access:

No direct access to SGX control structures

- Every EPC page has a type: SECS, TCS, normal page or VA
- Non are accessible from outside an enclave
- Only "normal pages" are accessible from inside an enclave
- SECS, TCS and VA are initialized and manipulated by the hardware