Intel® Software Guard Extensions enables applications to execute code and protect secrets from within their own protected execution environment, giving developers direct control over their application security.

**What is Intel® SGX?**

- Reduces the trusted computing base to the smallest possible footprint
- Prevents SW attacks even when OS/drivers/BIOS/VMM/SMM are compromised
- Secrets remain protected even when attacker has full control of the platform
- Prevents memory bus snooping, memory tampering, and “cold boot” attacks against memory images in RAM
- Provides hardware based attestation capabilities to measure and verify valid code and data signatures
- In-band execution utilizing the full power of the Intel® Processor

**Example Usages**

- User authentication factors and their computation remain private
- Company digital assets are protected regardless of the OS security posture
- Ultra Premium video content securely streamed to the home
- Conversations are kept confidential when video conferencing or using VOIP
- Peace of mind while managing financial accounts when using a browser

**How does it work?**

1. App built with trusted and untrusted parts
2. App runs & creates the enclave which is placed in trusted memory
3. Trusted function is called, execution transitioned to the enclave
4. Enclave sees all process data in clear; external access to enclave data is denied
5. Trusted function returns; enclave data remains in trusted memory
6. Application continues normal execution

**What is it?**

- **Memory images in RAM**
- **User authentication factors and their computation remain private**
- **Company digital assets are protected regardless of the OS security posture**
- **Ultra Premium video content securely streamed to the home**
- **Peace of mind while managing financial accounts when using a browser**

**Partitioning**

Identify sensitive application data (secrets) and the operations that work on/with that data
- Ex. Key material, proprietary algorithms, biometric data, CSR generation, etc.
- Partition this functionality to an enclave
- Do not hard code secrets into the enclave

**Enclave Interface Definition**

Careful definition of the enclave interface is critical
- The enclave’s interface is its attack surface; it should be minimal and avoid data leakage
- Enclave Definition Language (EDL) is used to define an enclave’s Trusted and Untrusted interface functions
- Tools process the EDL to create proxy / bridge code to call into (ECALL) and return from (OCALL) an enclave

**Attestation & Sealing**

Support for enclave attestation to a 3rd party
- Can attest; Enclave SW, CPU Security level, Sealing identity, and more.
- Data can be sealed against an enclave using a hardware derived Seal Key
- The Seal Key is unique to the CPU and the specific enclave environment

** Provisioning Enclave Secrets**

Well designed enclaves never contain hard coded secrets, instead, they are provisioned or created after the enclave is loaded
- Enclave binaries are un-encrypted inside CPU package
- Data and code outside CPU package is encrypted and integrity checked
- External memory reads and bus snoops see only encrypted data

**Process View**

Protected execution environment embedded in a process

**Execution Flow**

1. App built with trusted and untrusted parts
2. App runs & creates the enclave which is placed in trusted memory
3. Trusted function is called, execution transitioned to the enclave
4. Enclave sees all process data in clear; external access to enclave data is denied
5. Trusted function returns; enclave data remains in trusted memory
6. Application continues normal execution

**Security Perimeter**

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package
- Data and code outside CPU package is encrypted and integrity checked
- External memory reads and bus snoops see only encrypted data

**App Development**

SGX Application

**Partitioning**

Identify sensitive application data (secrets) and the operations that work on/with that data
- Ex. Key material, proprietary algorithms, biometric data, CSR generation, etc.
- Partition this functionality to an enclave
- Do not hard code secrets into the enclave