Legal Disclaimers

• The comments and statements are mine and not necessarily Intel’s.

• Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

• No computer system can be absolutely secure.
Outline

• Problem Statement
• Attack Surface and Overview
• Programming environment
  – System programming view
  – Day in the life of an enclave
• SGX Access Control & Off Chip protections
• Attestation and Sealing
• Developing with SGX
• Summary
Outline

• Problem Statement
• Attack Surface and Overview
• Programming environment
  – System programming view
  – Day in the life of an enclave
• SGX Access Control & Off Chip protections
• Attestation and Sealing
• Developing with SGX
• Summary
The Basic Issue: Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps ...
The Basic Issue: Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps ...

... and apps from each other ...
The Basic Issue: Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps ... 

App

Malicious App

... and apps from each other ...

... UNTIL a malicious app exploits a flaw to gain full privileges and then tampers with the OS or other apps

Apps not protected from privileged code attacks
The Basic Issue: Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps ...

... and apps from each other ...

... UNTIL a malicious app exploits a flaw to gain full privileges and then tampers with the OS or other apps

Apps not protected from privileged code attacks
Reduced attack surface with SGX
Reduced attack surface with SGX

Application gains ability to defend its own secrets
- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets
Reduced attack surface with SGX

Application gains ability to defend its own secrets
- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets

Familiar development/debug
- Single application environment
- Build on existing ecosystem expertise
Reduced attack surface with SGX

Application gains ability to defend its own secrets
- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets

Familiar development/debug
- Single application environment
- Build on existing ecosystem expertise

Familiar deployment model
- Platform integration not a bottleneck to deployment of trusted apps

Scalable security within mainstream environment
SGX Programming Environment

Trusted execution environment embedded in a process
SGX Programming Environment

Trusted execution environment embedded in a process
SGX Programming Environment

Trusted execution environment embedded in a process

With its own code and data
Provide Confidentiality
Provide integrity
With controlled entry points
SGX Programming Environment

Trusted execution environment embedded in a process

With its own code and data
Provide Confidentiality
Provide integrity
With controlled entry points
Supporting multiple threads
SGX Programming Environment

Trusted execution environment embedded in a process

User Process

- OS
- Enclave
- App Data
- App Code

Enclave

- Enclave Code
- Enclave Data
- TCS (*n)

With its own code and data
Provide Confidentiality
Provide integrity
With controlled entry points
Supporting multiple threads
With full access to app memory
Life Cycle of An Enclave
Life Cycle of An Enclave

Virtual Addr Space

Physical Addr Space
Life Cycle of An Enclave

Virtual Addr Space

Physical Addr Space

ECREATE (Range)

System Memory

Enclave Page Cache

EPCM

Invalid

Invalid

Invalid

Invalid
Life Cycle of An Enclave

Virtual Addr Space

Physical Addr Space

ECREATE (Range)

System Memory

Enclave Page Cache

SECS

MRENCLAVE

EPCM

Invalid

Invalid

Invalid

Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

ECREATE (Range)

Physical Addr Space

Plaintext Code/Data

System Memory

Enclave Page Cache

SECS

Valid, SECS

Invalid

Invalid

Invalid

MRENCLAVE

Build

EPCM

8
Life Cycle of An Enclave

Virtual Addr Space

Physical Addr Space

Build

Update PTE

ECREATE (Range)
EADD (Copy Page)
Life Cycle of An Enclave

Virtual Addr Space

Code/Data

Code/Data

Physical Addr Space

System Memory

Plaintext Code/Data

Enclave Page Cache

Plaintext Code/Data

Plaintext Code/Data

SECS

MRENCLAVE

ECREATE (Range)
EADD (Copy Page)
Life Cycle of An Enclave

Virtual Addr Space

- Code/Data
- Code/Data

Physical Addr Space

System Memory

- Enclave Page Cache
  - Plaintext Code/Data
  - Plaintext Code/Data
  - SECS

ECREATE (Range)
EADD (Copy Page)
EEXTEND

MRENCLAVE

Build

EPCM
- Invalid
- Valid,ID, LA
- Valid,ID, LA
- Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

Code/Data

Code/Data

ECREATE (Range)
EADD (Copy Page)
EEXTEND

Physical Addr Space

System Memory

Enclave Page Cache

Plaintext Code/Data

Plaintext Code/Data

SECS

MRENCLAVE

EPCM

Invalid

Valid,ID, LA

Valid,ID, LA

Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space
- Code/Data
- Code/Data

Physical Addr Space
- System Memory
- Enclave Page Cache
  - Plaintext Code/Data
  - Plaintext Code/Data
  - SECS

Events:
- ECREATE (Range)
- EADD (Copy Page)
- EEXTEND
- EINIT

MRENCLAVE
- EPCM
  - Invalid
  - Valid,ID, LA
  - Valid,ID, LA
  - Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

- Code/Data
- Code/Data

ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER

Physical Addr Space

System Memory

- Enclave Page Cache
- Plaintext Code/Data
- Plaintext Code/Data
- SECS

MRENCLAVE

EPCM

- Invalid
- Valid,ID, LA
- Valid,ID, LA
- Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

- Code/Data
- Code/Data
- Code/Data
- Code/Data
- ECREATE (Range)
- EADD (Copy Page)
- EEXTEND
- EINIT
- EENTER

Physical Addr Space

- System Memory
- Enclave Page Cache
- Plaintext Code/Data
- Plaintext Code/Data
- SECS
- MRENCLAVE

EPCM
- Invalid
- Valid,ID, LA
- Valid,ID, LA
- Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

- Code/Data
- Code/Data

- ECREATE (Range)
- EADD (Copy Page)
- EEXTEND
- EINIT
- EENTER
- EEXIT

Physical Addr Space

- System Memory
- Enclave Page Cache
- Plaintext Code/Data
- Plaintext Code/Data
- SECS

MRENCLAVE

EPCM
- Invalid
- Valid, ID, LA
- Valid, ID, LA
- Valid, SECS

Build
Life Cycle of An Enclave

Virtual Addr Space
- Code/Data
- Code/Data

Physical Addr Space
- System Memory
- Enclave Page Cache
- Plaintext Code/Data
- Plaintext Code/Data
- SECS

ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER
EEXIT

Build

MRENCLAVE

EPCM
Invalid
Valid,ID, LA
Valid,ID, LA
Valid,SECS
Life Cycle of An Enclave

Virtual Addr Space

- ECREATE (Range)
- EADD (Copy Page)
- EEXTEND
- EINIT
- EENTER
- EEXIT
- EREMOVE

Physical Addr Space

- System Memory
- Enclave Page Cache

Build

MRENCLAVE

EPCM
- Invalid
- Invalid
- Invalid
- Invalid
SGX Access Control

Linear Address → Traditional IA Page Table Checks → Physical Address
SGX Access Control

Linear Address → Traditional IA Page Table Checks → Physical Address → Enclave Access?
SGX Access Control

Linear Address → Traditional IA Page Table Checks → Physical Address

Enclave Access?

No

Non-Enclave Access

Address in EPC?

Yes

Replace Address With Abort Page

No

Allow Memory Access
SGX Access Control

Linear Address

Traditional IA Page Table Checks

Enclave Access

Physical Address

Enclave Access?

Yes

Address in EPC?

Yes

Check EPCM

Checks Pass?

Yes

No

Non-Enclave Access

Address in EPC?

Yes

Replace Address With Abort Page

No

Allow Memory Access
SGX Access Control

Linear Address

Traditional IA Page Table Checks

Physical Address

Enclave Access?

Yes

Address in EPC?

Yes

Check EPCM

Pass?

Yes

Allow Memory Access

No

Replace Address With Abort Page

Non-Enclave Access

Address in EPC?

Yes

No

Signal Fault

Enclave Access

No

Checks Pass?

Yes

No
Protection vs. Memory Snooping Attacks

Non-Enclave Access
Protection vs. Memory Snooping Attacks

- Security perimeter is the CPU package boundary

Non-Enclave Access
- Security perimeter is the CPU package boundary
Protection vs. Memory Snooping Attacks

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package

Non-Enclave Access

AMEX: 3234-134584-26864
Protection vs. Memory Snooping Attacks

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package
- Data and code outside CPU package is encrypted and/or integrity checked
Protection vs. Memory Snooping Attacks

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

• Problem Statement
• Attack Surface and Overview
• Programming environment
  – System programming view
  – Day in the life of an enclave
• SGX Access Control & Off Chip protections
• Attestation and Sealing
• Developing with SGX
• Summary
The Challenge – Provisioning Secrets to the Enclave

• An enclave is in the clear before instantiation
  - Sections of code and data could be encrypted, but their decryption key can’t be pre-installed

• Secrets come from outside the enclave
  - Keys
  - Passwords
  - Sensitive data

• The enclave must be able to convince a 3rd party that it’s trustworthy and can be provisioned with the secrets

• Subsequent runs should be able to use the secrets that have already been provisioned
Trustworthiness

- A service provider should vet the enclave’s Trusted Computing Base (TCB) before it should trust it and provide secrets to it
  - The enclave’s software
  - The CPU’s hardware & firmware

- Intel® SGX provides the means for an enclave to securely prove to a 3rd party:
  - What software is running inside the enclave
  - Which execution environment the enclave is running at
  - Which Sealing Identity will be used by the enclave
  - What’s the CPU’s security level
Attestation – Software TCB

• When building an enclave, Intel® SGX generates a cryptographic log of all the build activities
  - Content: Code, Data, Stack, Heap
  - Location of each page within the enclave
  - Security flags being used

• MRENCLAVE (“Enclave Identity”) is a 256-bit digest of the log
  - Represents the enclave’s software TCB

• A software TCB verifier should:
  - Securely obtain the enclave’s software TCB
  - Securely obtain the expected enclave’s software TCB
  - Compare the two values
Local Attestation

• “Local attestation”: The process by which one enclave attests its TCB to another enclave on the same platform

• Using Intel® SGX’s EREPORT and EGETKEY instructions
  - EREPORT generates a cryptographic REPORT that binds MRENCLAVE to the target enclave’s REPORT KEY
  - EGETKEY provides the REPORT KEY to verify the REPORT

<table>
<thead>
<tr>
<th>TCB component</th>
<th>Attestation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Firmware &amp; hardware</td>
<td>Symmetric - CPU REPORT KEY</td>
</tr>
<tr>
<td>Software</td>
<td>MRENCLAVE</td>
</tr>
</tbody>
</table>
Remote Attestation

• “Remote attestation”: The process by which one enclave attests its TCB to another entity outside of the platform

• Intel® SGX Extends Local attestation by allowing a Quoting Enclave (QE) to use Intel® EPID to create a QUOTE out of a REPORT
  - Intel® EPID is a group signature scheme

<table>
<thead>
<tr>
<th>TCB component</th>
<th>Attestation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Firmware &amp; hardware</td>
<td>Asymmetric - Intel® EPID</td>
</tr>
<tr>
<td>Software</td>
<td>MRENCLAVE</td>
</tr>
</tbody>
</table>
Local Attestation - Flow

Processor

Client Application

Client Application
Local Attestation - Flow

1. Verifying enclave sends its MRENCLAVE to reporting enclave
Local Attestation - Flow

1. Verifying enclave sends its MRENCLAVE to reporting enclave
Local Attestation - Flow

1. Verifying enclave sends its MRENCLAVE to reporting enclave
2. Reporting enclave creates a cryptographic REPORT that includes its MRENCLAVE
Local Attestation - Flow

1. Verifying enclave sends its MRENCLAVE to reporting enclave
2. Reporting enclave creates a cryptographic REPORT that includes its MRENCLAVE
3. Verifying enclave obtains its REPORT key and verifies the authenticity of the REPORT
Local Attestation - Flow

1. Verifying enclave sends its MRENCLAVE to reporting enclave
2. Reporting enclave creates a cryptographic REPORT that includes its MRENCLAVE
3. Verifying enclave obtains its REPORT key and verifies the authenticity of the REPORT
Remote Attestation - Flow

1. Verifying enclave becomes the Quoting Enclave.
2. After verifying the REPORT the, QE signs the REPORT with the EPID private key and converts it into a QUOTE.
3. Remote platform verifies the QUOTE with the EPID public key and verifies MRENCLAVE against the expected value.
1. Verifying enclave becomes the Quoting Enclave.
2. After verifying the REPORT the, QE signs the REPORT with the EPID private key and converts it into a QUOTE.
3. Remote platform verifies the QUOTE with the EPID public key and verifies MRENCLAVE against the expected value.
1. Verifying enclave becomes the Quoting Enclave.
2. After verifying the REPORT the, QE signs the REPORT with the EPID private key and converts it into a QUOTE
3. Remote platform verifies the QUOTE with the EPID public key and verifies MRENCLAVE against the expected value
Remote Attestation - Flow

1. Verifying enclave becomes the Quoting Enclave.
2. After verifying the REPORT the, QE signs the REPORT with the EPIP private key and converts it into a QUOTE.
3. Remote platform verifies the QUOTE with the EPIP public key and verifies MRENCLAVE against the expected value.
1. Verifying enclave becomes the Quoting Enclave.
2. After verifying the REPORT the, QE signs the REPORT with the EPID private key and converts it into a QUOTE
3. Remote platform verifies the QUOTE with the EPID public key and verifies MRENCLAVE against the expected value
Sealing Authority

• Every enclave has an Enclave Certificate (SIGSTRUCT) which is signed by a Sealing Authority
  - Typically the enclave writer
  - SIGSTRUCT includes:
    - Enclave’s Identity (represented by MRENCLAVE)
    - Sealing Authority’s public key (represented by MRSIGNER)

• *EINIT* verifies the signature over SIGSTRUCT prior to enclave initialization
Sealing

“Sealing”: Cryptographically protecting data when it leaves the enclave.

Enclaves use EGETKEY to retrieve an enclave, platform persistent key and encrypts the data.

EGETKEY uses a combination of enclave attributes and platform unique key to generate keys:
- Enclave Sealing Authority
- Enclave Product ID
- Enclave Product Security Version Number (SVN)
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between envelope and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls EREPORT to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls EGETKEY to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Example: Secure Transaction

1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls *EREPORT* to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls *EGETKEY* to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
1. Enclave built & measured against ISV’s signed certificate
2. Enclave calls *EREPORT* to obtain a REPORT that includes enclave specific data (ephemeral key)
3. REPORT & user data sent to Quoting Enclave who signs the REPORT with an EPID private key
4. QUOTE sent to server & verified
5. Ephemeral key used to create a trusted channel between enclave and remote server
6. Secret provisioned to enclave
7. Enclave calls *EGETKEY* to obtain the SEAL KEY
8. Secret is encrypted using SEAL KEY & stored for future use
Intel® SGX Software Development

Application

App Code
Processing Component

Processing Component

Intel SGX enabled CPU
Intel® SGX Software Development

- Software Developer decides which components should execute within an enclave

Trusted

- Windows DLL / Enclave
- Processing Component

Application

- App Code
- Processing Component

Intel SGX enabled CPU
Intel® SGX Software Development

- Software Developer decides which components should execute within an enclave
- Development Environment allows the Developer to quickly develop enclave enabled binaries

Windows DLL / Enclave

Processing Component

App Code

Processing Component

Intel SGX enabled CPU
Intel® SGX Software Development

- Software Developer decides which components should execute within an enclave.
- Development Environment allows the Developer to quickly develop enclave enabled binaries.
- Including support for common software libraries, exporting interfaces, and support for provisioning.

Intel SGX enabled CPU
SGX Technical Summary

• Provides any application the ability to keep a secret
  • Provide capability using new processor instructions
  • Application can support multiple enclaves

• Provides integrity and confidentiality
  • Resists hardware attacks
  • Prevent software access, including privileged software and SMM

• Applications run within OS environment
  • Low learning curve for application developers
  • Open to all developers

• Resources managed by system software
Links

Joint research poster session:  
http://sigops.org/sosp/sosp13/

Public Cloud Paper using SGX2:  

Programming Reference for SGX1 & SGX2:  
http://www.intel.com/software/isa

HASP Workshop:  
https://sites.google.com/site/haspworkshop2013/workshop-program
Thank You
Backup
SGX Paging Introduction

Requirement:

• Remove an EPC page and place into unprotected memory. Later restore it.

• Page must maintain same security properties (confidentiality, anti-replay, and integrity) when restored

Instructions:

• EWB: Evict EPC page to main memory with cryptographic protections

• ELDB/ELDU: Load page from main memory to EPC with cryptographic protections

• EPA: Allocate an EPC page for holding versions

• EBLOCK: Declare an EPC page ready for eviction

• ETRACK: Ensure address translations have been cleared
Page-out Example

EPC

SECS
Enclave Page
VA Page
Enclave Page
Enclave Page

EWB

System Memory
Page-out Example

EWB Parameters:
- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location
**Page-out Example**

**EWB Parameters:**
- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location

**EWB Operation**
- Remove page from the EPC
- Populate version slot
- Write encrypted version to outside
- Write meta-data, PCMD
Page-out Example

**EWB Parameters:**
- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location

**EWB Operation**
- Remove page from the EPC
- Populate version slot
- Write encrypted version to outside
- Write meta-data, PCMD

All pages, including SECS and Version Array can be paged out.
Page-in Example

EPC

ELD

System Memory

SECS

Enclave Page

Free Enclave Page

Encrypted Page

VER

MD
Page-in Example

ELD Parameters:
- Encrypted page
- Free EPC page
- SECS (for an enclave page)
- Populated version slot
Page-in Example

ELD Parameters:
- Encrypted page
- Free EPC page
- SECS (for an enclave page)
- Populated version slot

ELD Operation:
- Verify and decrypt the page using version
- Populate the EPC slot
- Make back-pointer connection (if applicable)
- Free-up version slot