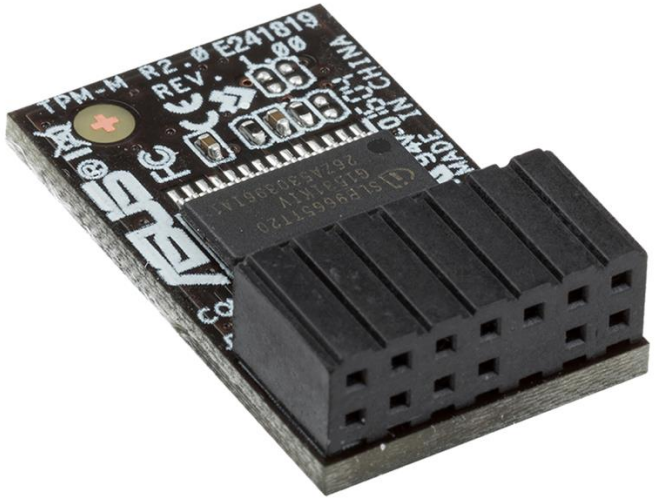
Trusted Computing

CS463/ECE424

University of Illinois

Trusted Platform Module (TPM)
Secure Enclaves and SGX



History of Tamper Resistance

- History of Tamper Resistance goes back centuries
- Examples:
  - Weight down **code books** on naval ships
  - The keys for wartime cipher machines have been printed in **water soluble ink**
  - IBM 3848 and the VISA security module (1980s)

Motivation of Secure Hardware:
Prevent **powerful** adversaries from getting secret (e.g., those that compromised OS)

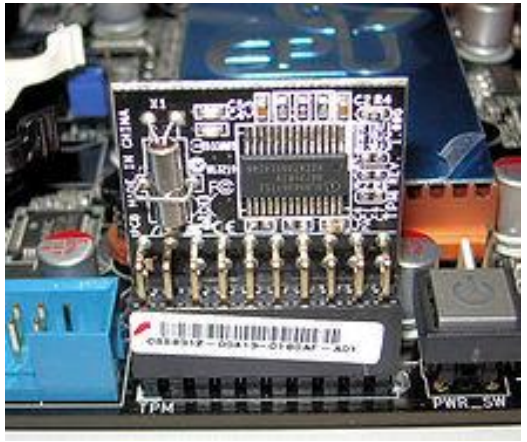
Trusted Computing

- Trusted Computing is the term developed by the Trusted Computing Group (TCG)
  - Founded in 1999 with a large number of companies

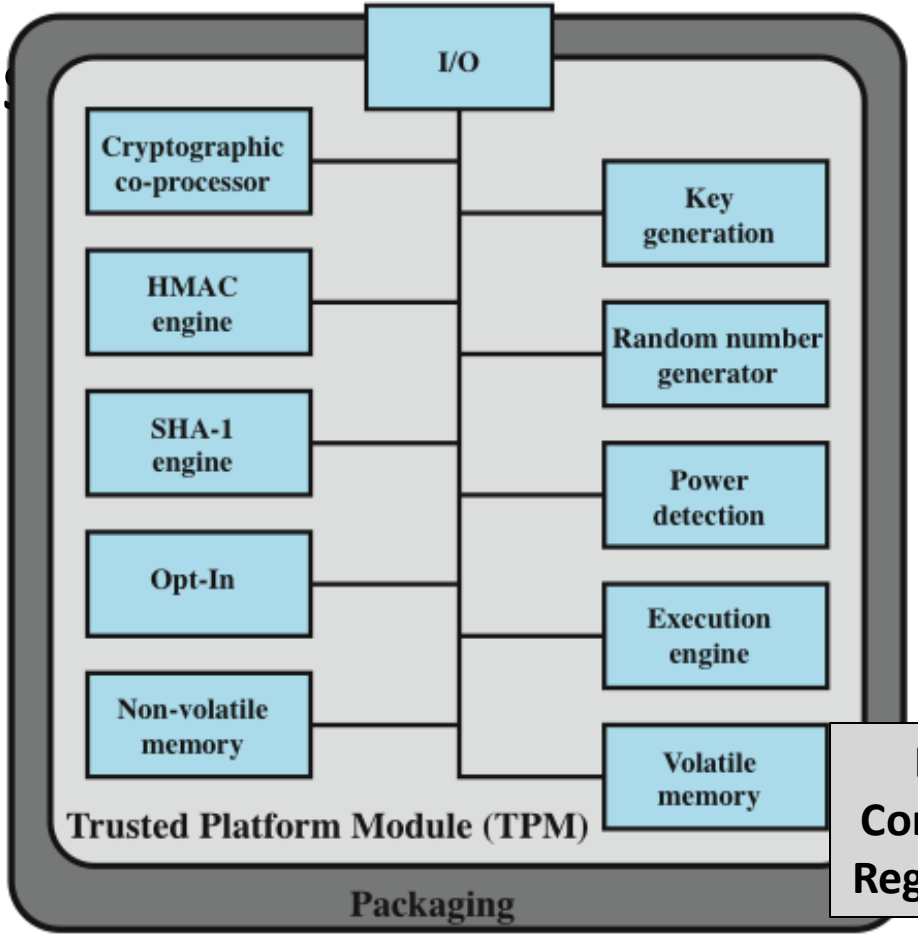
- Trusted Computing allows “a piece of data to dictate what Operating System and Application must be used to open it”

Trusted Platform Module (TPM)

- Hardware module at heart of *hardware/ software* approach to Trusted Computing
- Uses a TPM chip
- Has three basic services:
  - Encryption, certification, authenticated boot

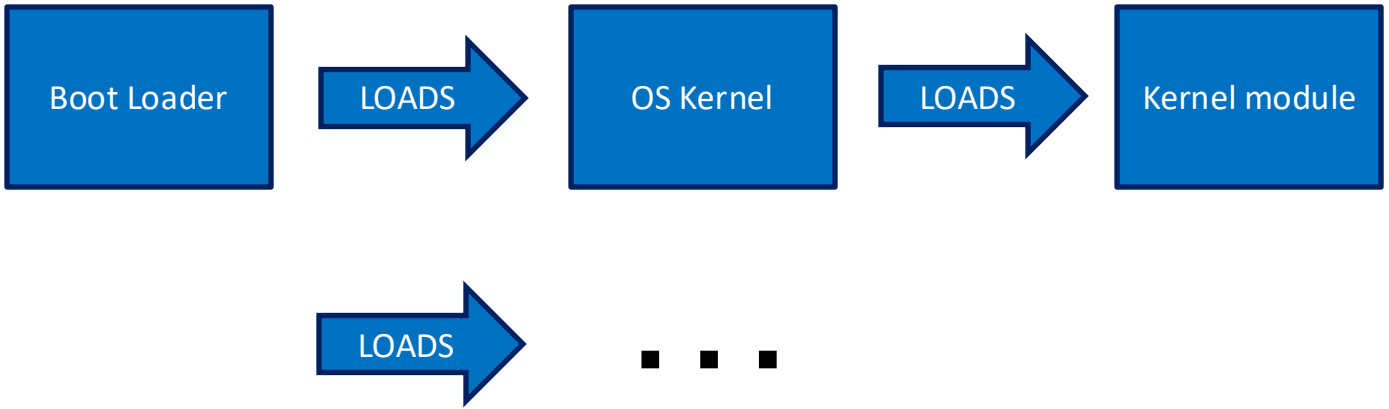


TPM Functions



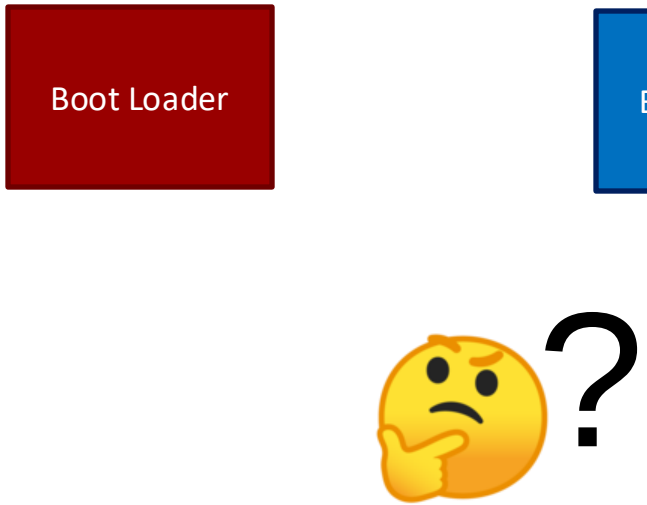
Platform Configuration Registers (PCR)

Secure/Authenticated Boot Service



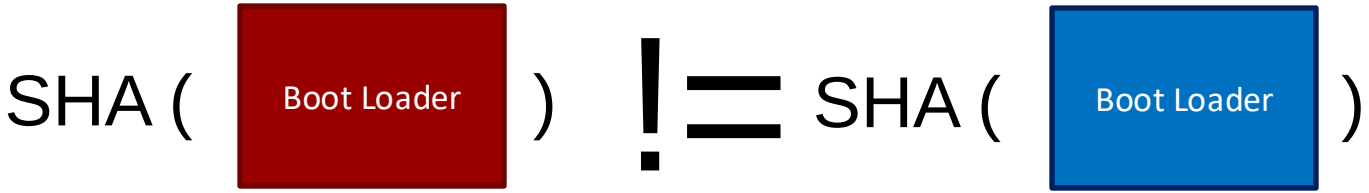
Secure/Authenticated Boot Service

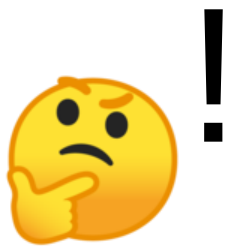
- Q: How can we verify that our boot loader is not tampered?



Secure/Authenticated Boot Service

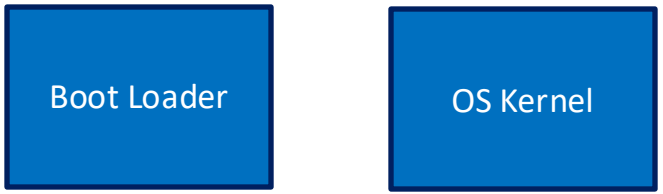
- A: Hashing!



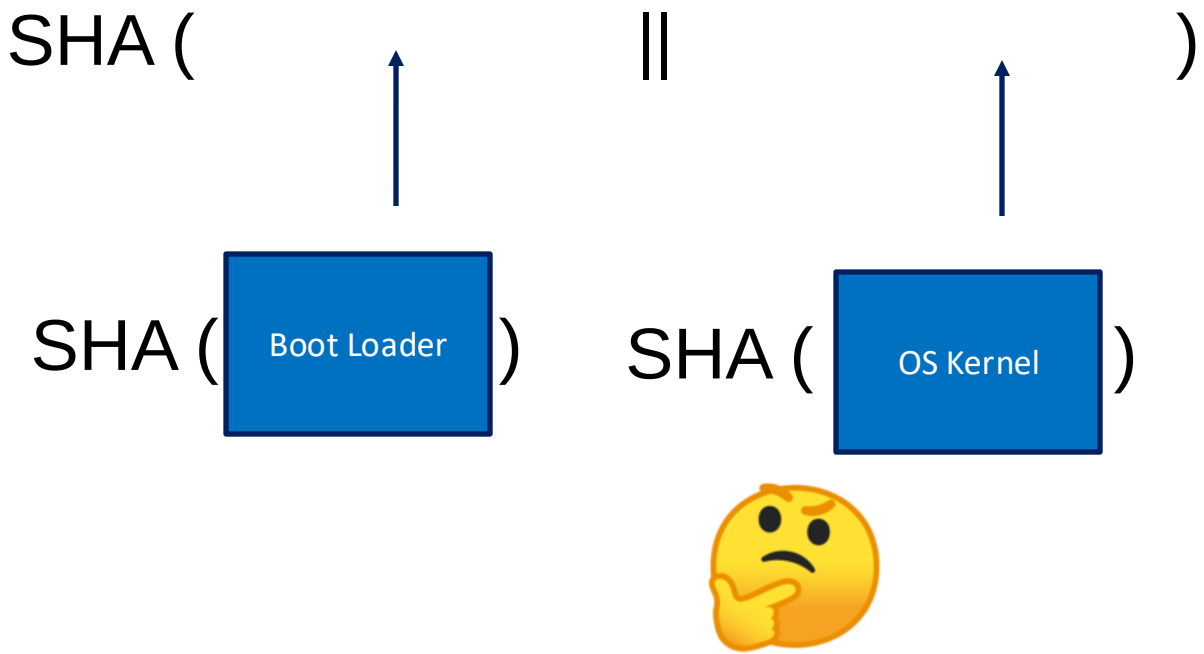


Secure/Authenticated Boot Service

- Q: How can we verify Boot Loader and OS Kernel are both not tampered?

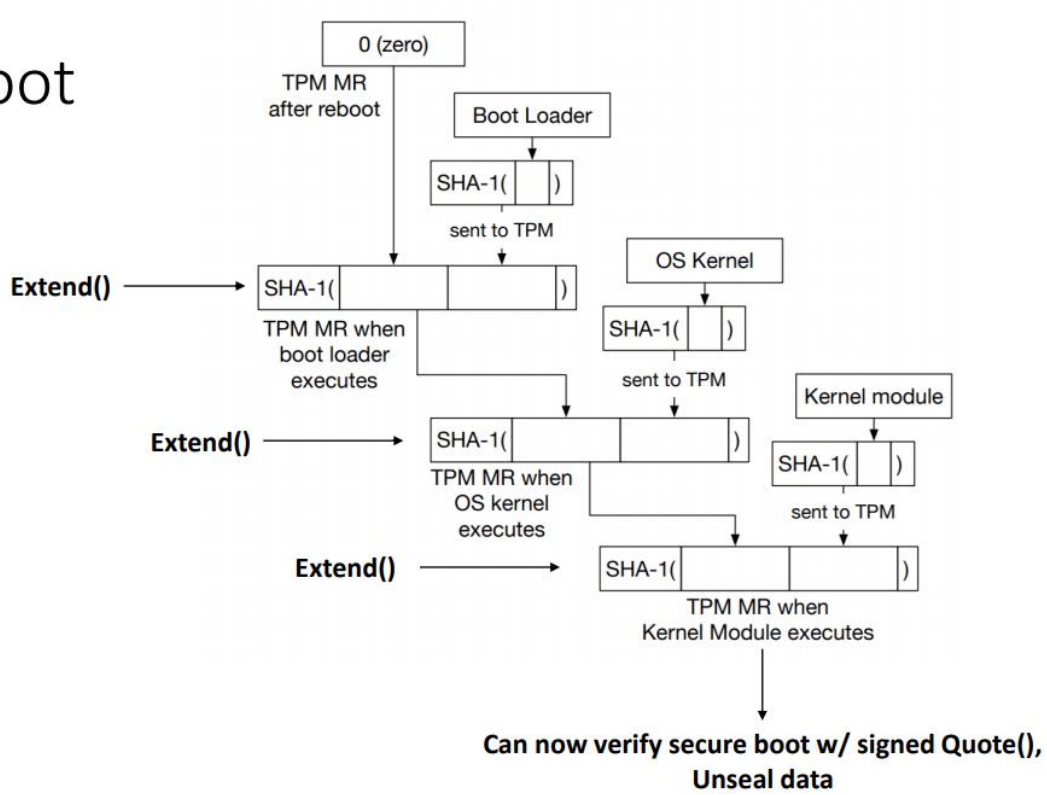


Secure/Authenticated Boot Service



Secure/Authenticated Boot Service

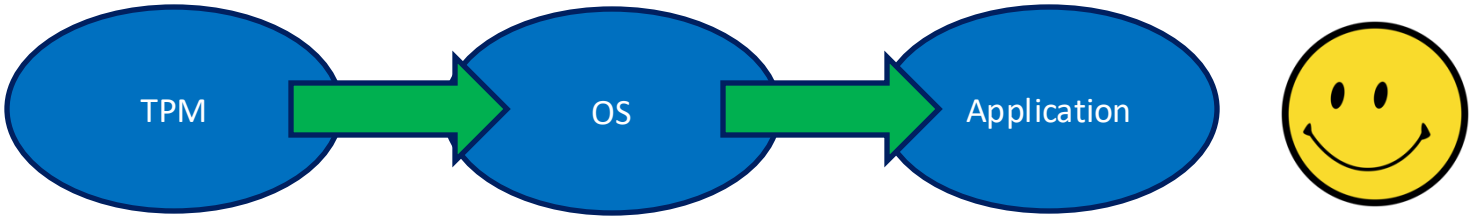
Secure boot



Certification Service

- Once a configuration is achieved and logged, the TPM can certify configuration to others (attestation)

- Challenge value in certificate assures timeliness
  - Use a random number as the challenge when requesting a certificate from TPM
- Provides a hierarchical certification approach



Encryption Service

- Encrypts data so that it can only be decrypted by a machine with a certain configuration
- TPM maintains a master secret key unique to machine
- Can extend scheme upward
  - Extend the trust to OS, and then application

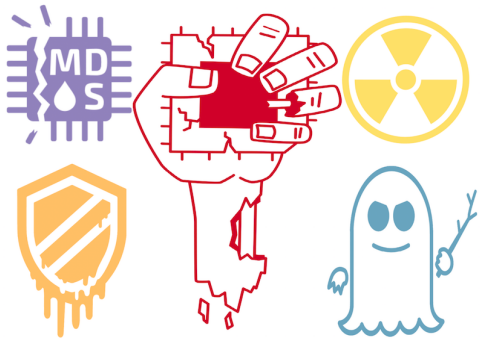
Example: Windows BitLocker Drive Encryption

- **Windows 10** uses TPM on multiple components including its **drive encryption**
- **BitLocker** relies on the TPM to allow the use of a key only when startup occurs in an **expected** way
- The hard drive remains confidential when:
  - Different OS booted from USB device
  - Hard disk is lost or stolen when powered off

Criticisms against TPM

The concept of Trusted Computing is developed and promoted by the Trusted Computing Group

- Root of trust
- Anti-competitive effect



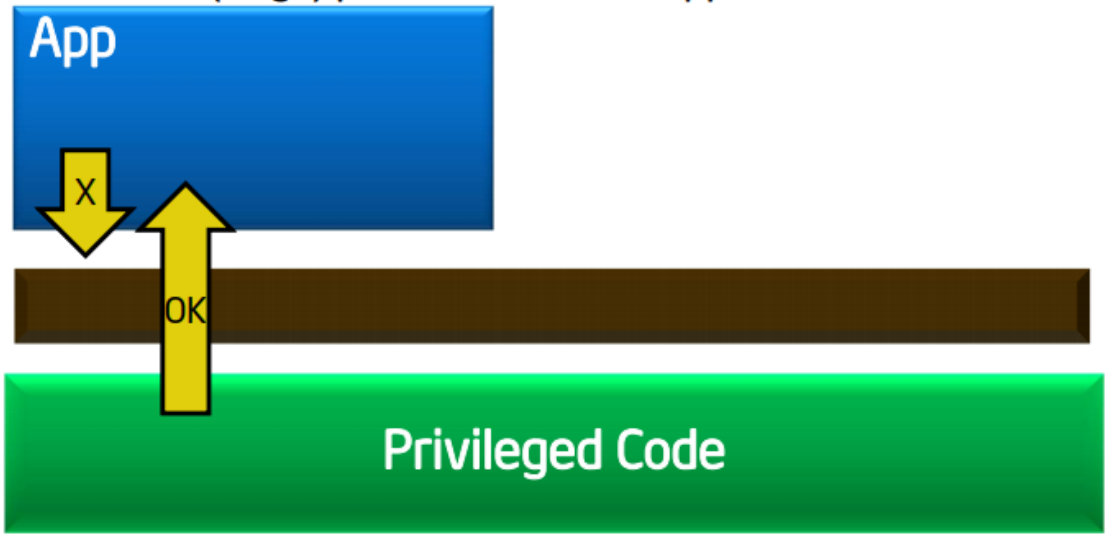
Secure Enclave and SGX

References/Credits:

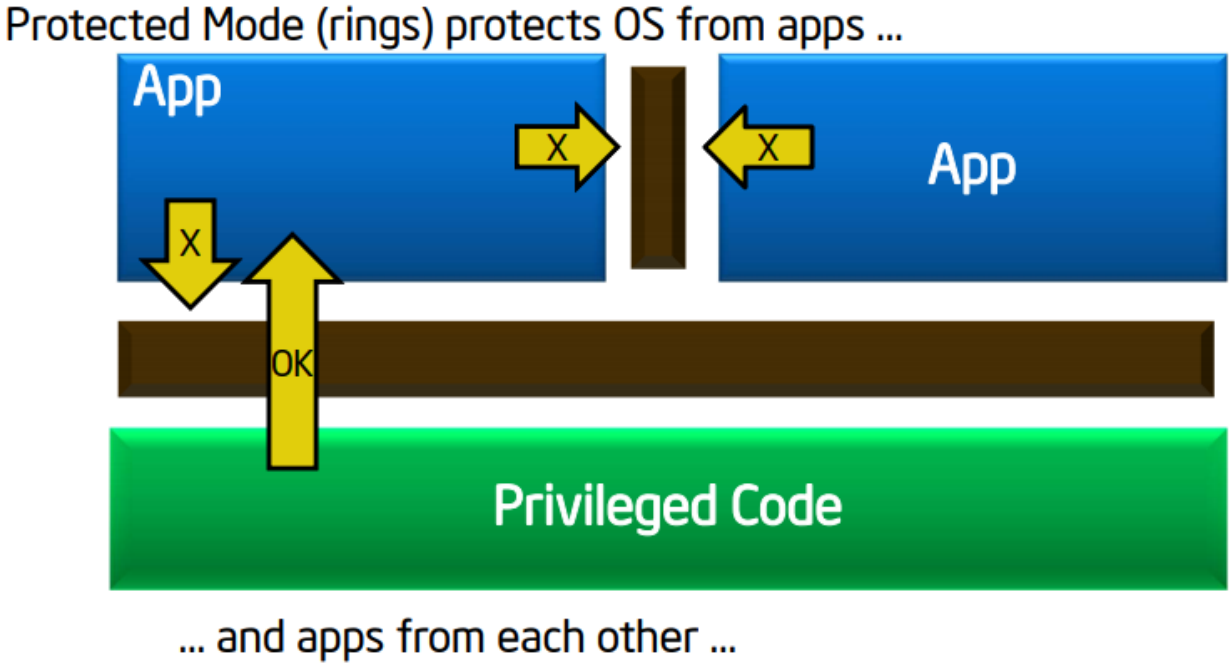
1. Intel SGX (Reference Number: 332680-002) presented at ISCA 2015
2. Intel's SGX In-depth Architecture by Syed Kamran Haider with Hamza Omar, Masab Ahmad, Chenglu Jin, and Marten van Dijk

Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps …

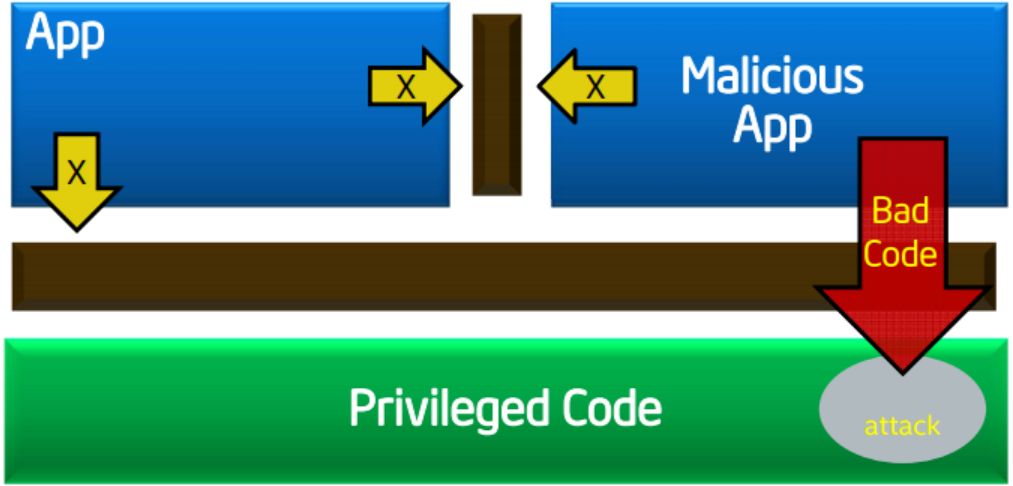


Why Aren’t Compute Devices Trustworthy?



Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps …



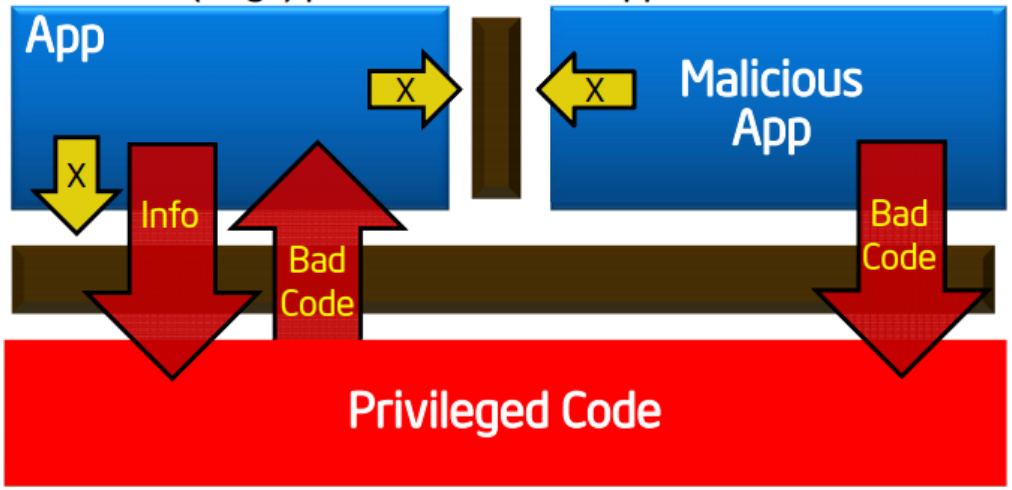
… and apps from each other …

… 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

Why Aren’t Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps …



… and apps from each other …

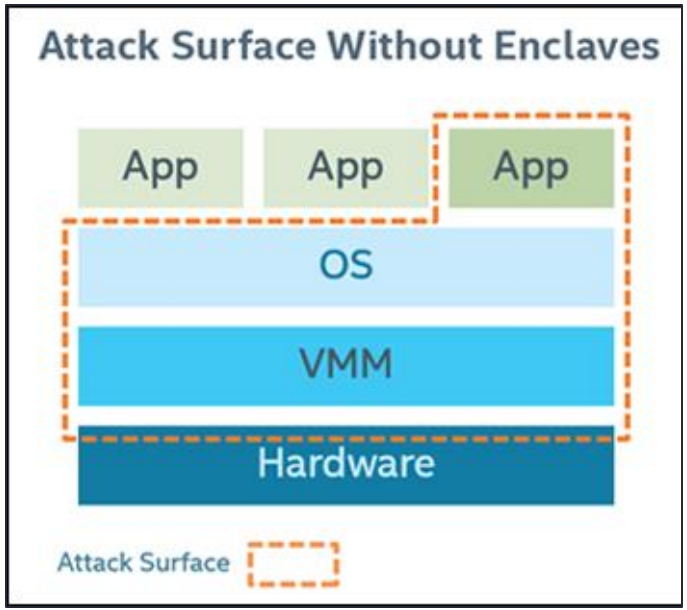
… UNTIL a malicious app exploits a flaw to gain full privileges and then tampers with the OS or other apps

Apps not protected from privileged code attacks

Reduced Attack Surface with SGX

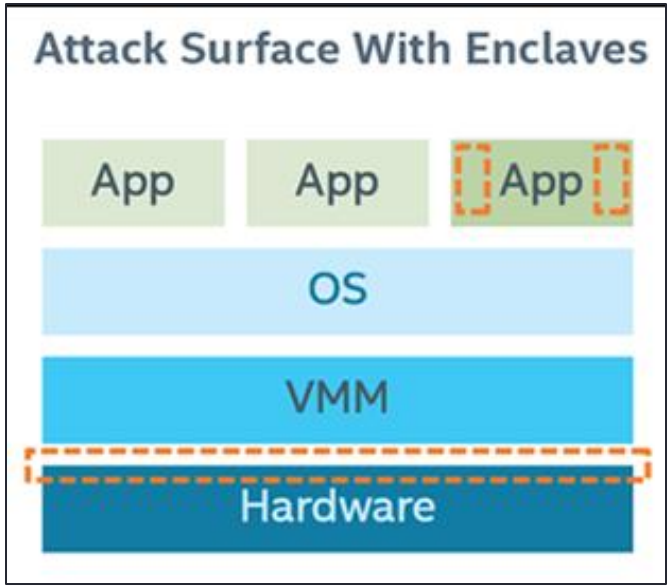
- Complexity of modern systems force programmers to inspect a large code base for vulnerability detections

- Application code
- OS code
- Virtual Machine Manager Code



Reduced attack surface with SGX

- Applications gain ability to defend their 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

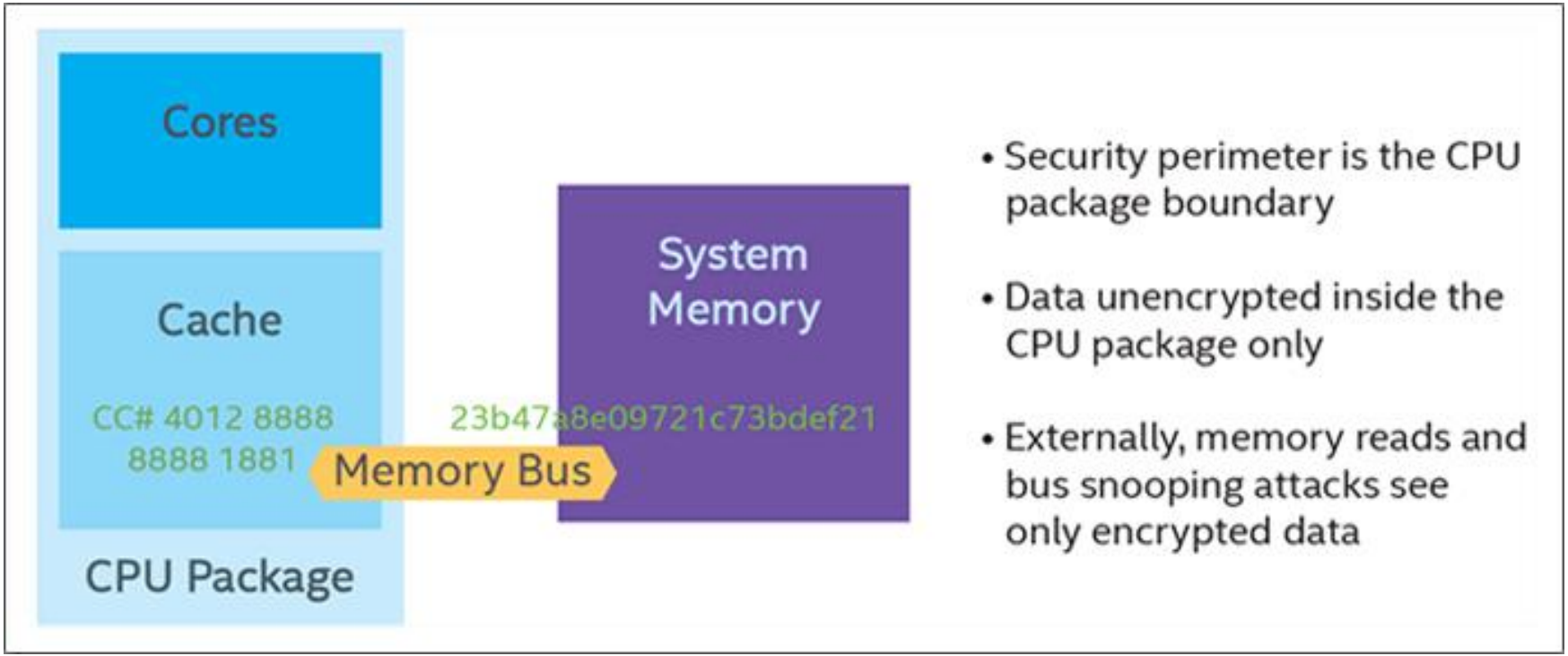


Intel becomes the root of trust in this security model.

SGX: Software Guard Extensions

- **Intel Software Guard Extensions (SGX)**
  - Built into Intel CPUs
  - The built-in CPU instructions allow user-level as well as OS code to define private regions of memory, called *enclaves*
  - Contents in enclaves are encrypted and unable to be either read or written by any process outside the enclave (including privileged processes).

SGX Overview



https://software.intel.com/content/www/us/en/develop/articles/intel-software-guard-extensions-tutorial-part-1-foundation.html

Isolation and Attestation

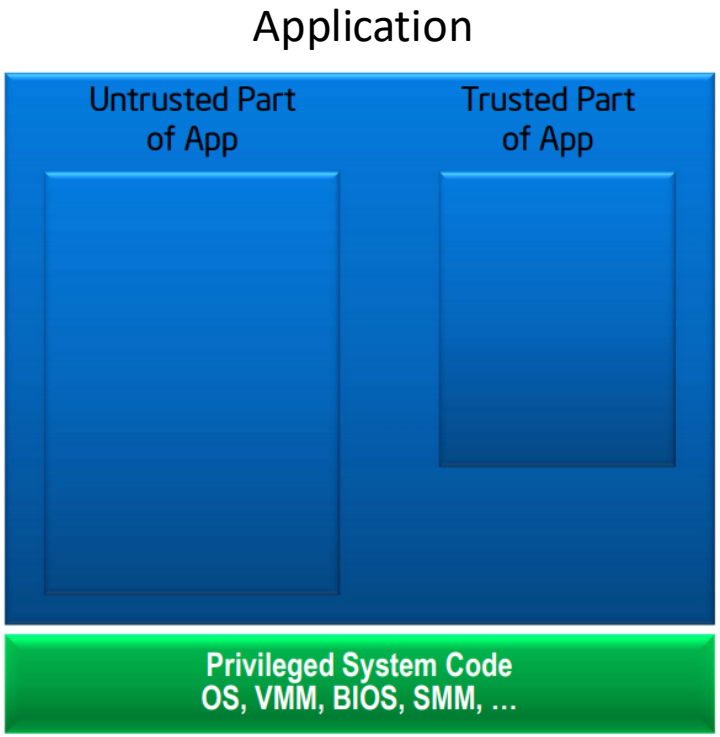
SGX enabled processors offer two crucial properties.

- **Isolation:** Each enclave's environment is isolated from the untrusted software outside the **enclave**, as well as from other enclaves.
- **Attestation:** A software attestation scheme that allows a remote party to authenticate the software running inside an **enclave**.

We will focus on the isolation property for the remainder of the lecture.

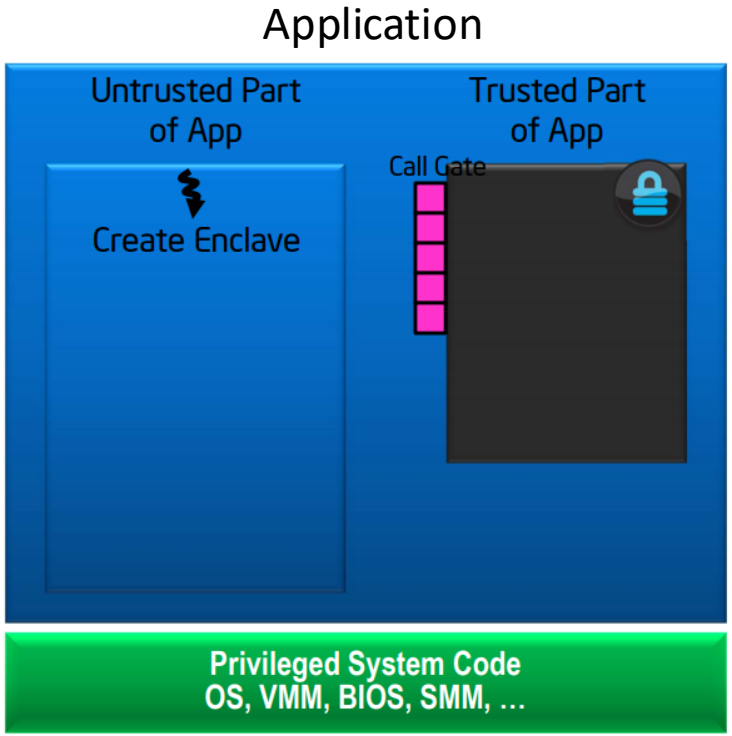
How Secure Enclaves Work

- Application is built with **trusted** and **untrusted** parts
- **Trusted** and **untrusted** parts are explicitly separated by app developers



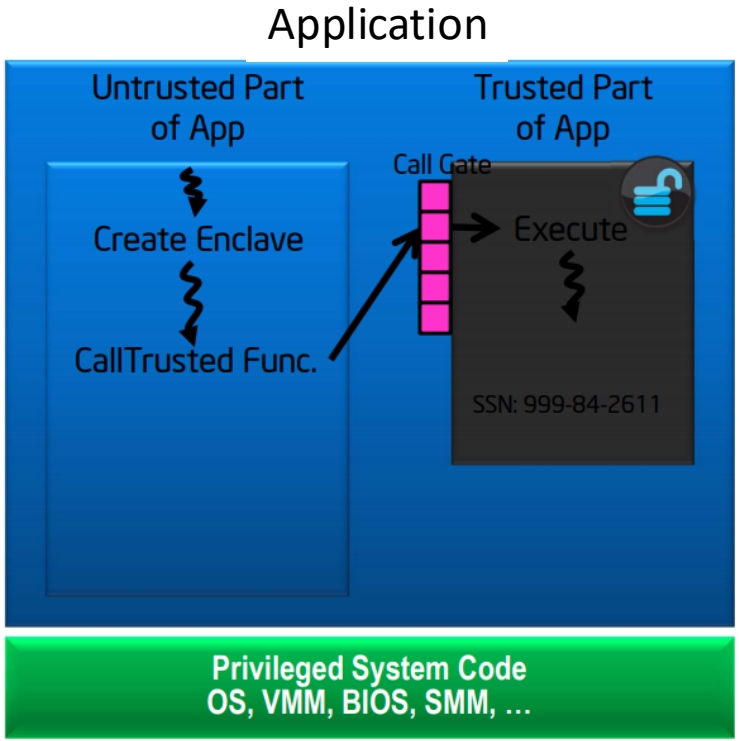
How Secure Enclaves Work

- App runs & creates **enclave** which is placed in **trusted memory**
- The memory region (enclave page cache, or EPC) is encrypted
- When processor fetches the data, it needs to decrypt it



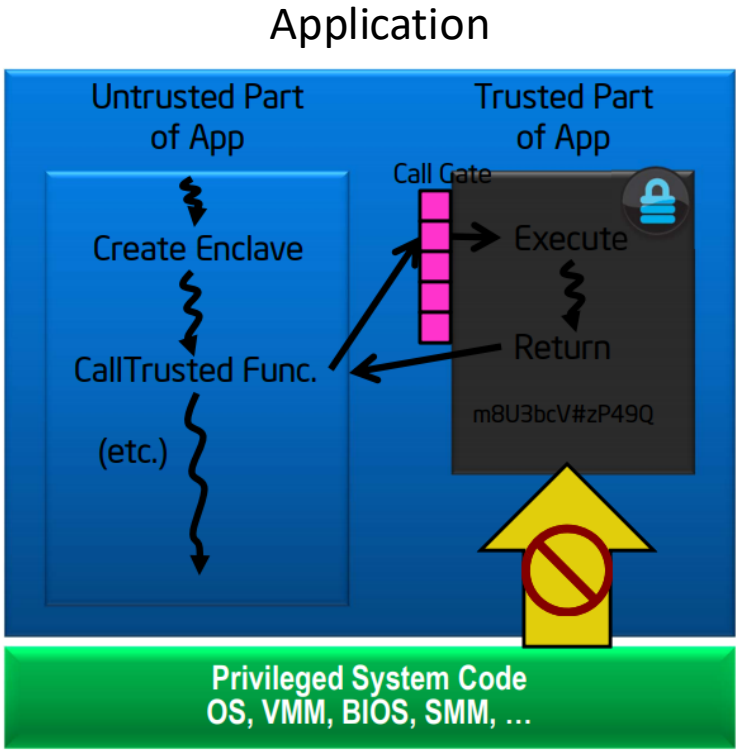
How Secure Enclaves Work

- Trusted function (ECALL) is called
- Code running inside enclave sees data in clear
- **External access to data is denied**



How Secure Enclaves Work

- Function returns (OCALL)
- Enclave data **remains** in trusted memory



Enclaves and Objects

- An Enclave is like a class.
- It can maintain its own state like Objects do with private variables.
- ECALLs is like a method, and OCALL is like a return.
- Enclave’s property of isolation is like Object Oriented Programming’s principle of encapsulation.

Intel SGX SDK for Linux

- Intel provides a Software Development Kit (SDK) for C/C++ programmers: https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions/get-started.html

"It is a collection of APIs, libraries, documentation, sample source code, and tools that allows developers to create and debug Intel SGX enabled applications."

Disclaimer

- SGX allows a subset of C/C++ library functions to be used inside the Enclave
- List of allowed/disallowed library functions are defined at the Intel SGX Developer Reference

https://download.01.org/intel-sgx/linux-2.4/docs/Intel_SGX_Developer_Reference_Linux_2.4_Open_Source.pdf

Limitations

SGX does not defend against software side-channel adversary!

Software Side-Channel Adversary: An adversary who can gather statistics from the CPU regarding execution and may be able to use them to deduce characteristics of the software being executed (side-channel analysis)

1. Gather power statistics
2. Gather performance statistics including platform cache misses
3. Gather branch statistics via timing
4. Gather information on pages accessed via page tables

Discussion Questions

- Should we accept Intel as a root of trust?

- What are some use cases for Trusted Computing in addition to disk encryption (e.g., Bitlocker)?