Introduction to Information Security
14-741/18-631 Fall 2021
Unit 3: Lecture 5:
Trusted Platform Module & Intel SGX & HW issues

Limin Jia liminjia@andrew

This Lecture's Agenda

- **Outline**
  - TCG
    - Trusted Computing Group
    - Trusted Platform Module
    - Examples: Bitlocker
  - Intel SGX
- **Objective**
  - Spark interest in a very current security topic
  - Provide an application example of some of the security materials we've seen earlier
  - Evidence the potential danger of trust assumptions on hardware

What is Trusted Computing?

- **A technology to ensure the system (computer) is trusted to run only authorized programs**
  - E.g., only "authentic" version of Windows can run
  - Trusted by whom?
    - Software vendor
    - User
- **Controversial topic**
  - **Advocates**
    - Make computers safer
    - Protection against viruses and malware
  - **Opponents**
    - Too much power and control into software vendors

Trusted vs Trustworthy

- A trusted system or component is one whose failure can break the security policy
- A trustworthy system or component is one that will not fail

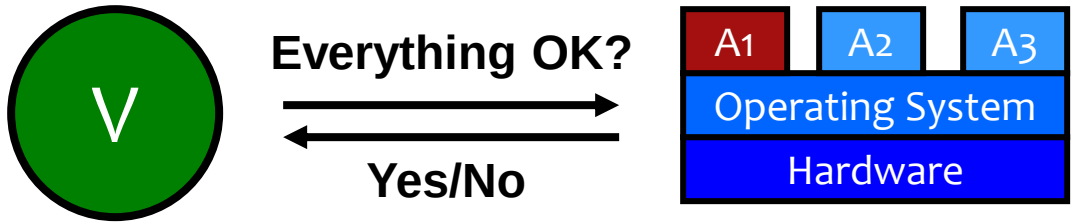
- Trusted Computing as used in the title of this lecture is really intended to mean "Towards Trustworthy Computing"

Why Trusted Computing?

- **Computers are everywhere**
  - Mobile devices, cars, …
- **Computing is distributed everywhere**
  - Cloud
- **How can we be sure that actions are carried out as specified?**
  - Malicious software and hardware

Externally Verifiable?

- **Desirable property: Remotely verify trustworthy device operation**



Outline

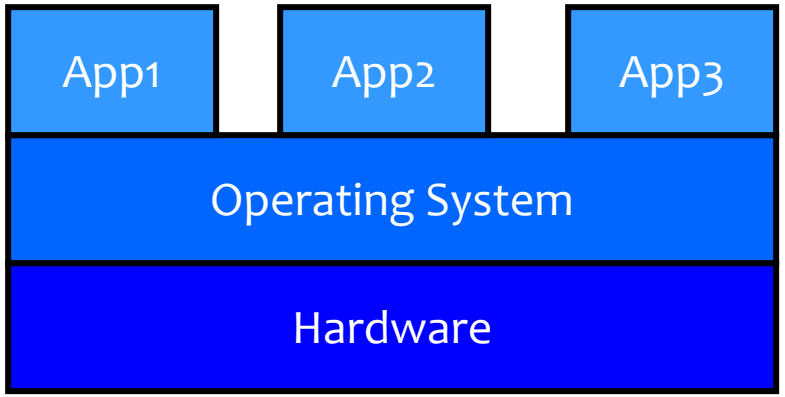
- **OS challenge: we study problem of how a remote entity can establish trust in a software system**
  - How can we (remotely) establish trust that correct OS and correct software is executing?
- **Review of some current approaches for building secure (trustworthy) systems**
- **Commodity TPM-based attestation**
  - Static Root of Trust (version 1.1b)

Adversary Model

- **Axiom: Every system has at least one more flaw ☺**
- **We assume remote adversary who can launch network-based attacks**
  - Adversary may control network communication
  - Adversary can compromise OS and/or applications
- **We trust local hardware**
  - local hardware attacks are even harder to defend against
- **Practical model, as remote attacks constitute majority of threats against commodity systems**

Security Properties to Consider

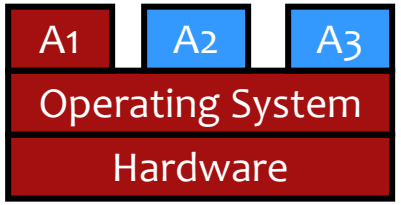
- **Trustworthy device operation**
  - How can we trust operations that our devices perform?
- **Questions to consider**
  - How can we trust App1?
  - What if App2 has a security vulnerability?
  - What if Operating System has a security vulnerability?



Approaches to improve the trustworthiness of the execution environment

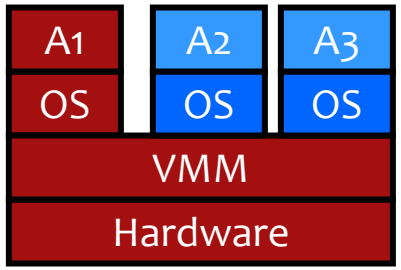
- **Isolation: Virtual-machine-based isolation**
- **Fix what's running**
  - Program code in ROM
  - Secure boot

- **Evaluation metric: size of Trusted Computing Base (TCB)**
- **We visualize components in TCB in red:**



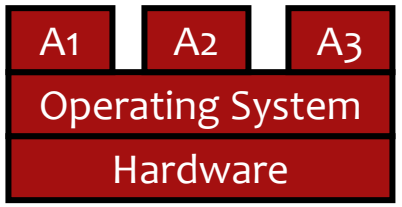
Virtual-machine-based Isolation

- **Approach: Isolate applications by executing them inside different Virtual Machines**
- **Advantages**
  - Smaller TCB
  - Isolation between applications
  - Eliminate worries about other applications
- **Disadvantages**
  - VMM is still large and part of TCB
  - Relatively complex, not well suited for average user
- **Verdict: Smaller TCB, step in right direction**



Program Code in ROM

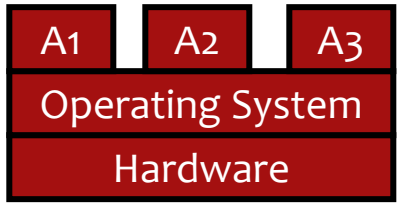
- **Approach: keep entire program in ROM**
- **Advantages**
  - Simplicity
  - Adversary cannot inject any additional software
- **Disadvantages**
  - Cannot update software (without exchanging ROM)
  - Adversary can still use control-flow attack
  - Entire system is in TCB, no isolation



- **Verdict**
  - Impractical for current systems, ability to update code for enhancing features or fixing bugs is critical

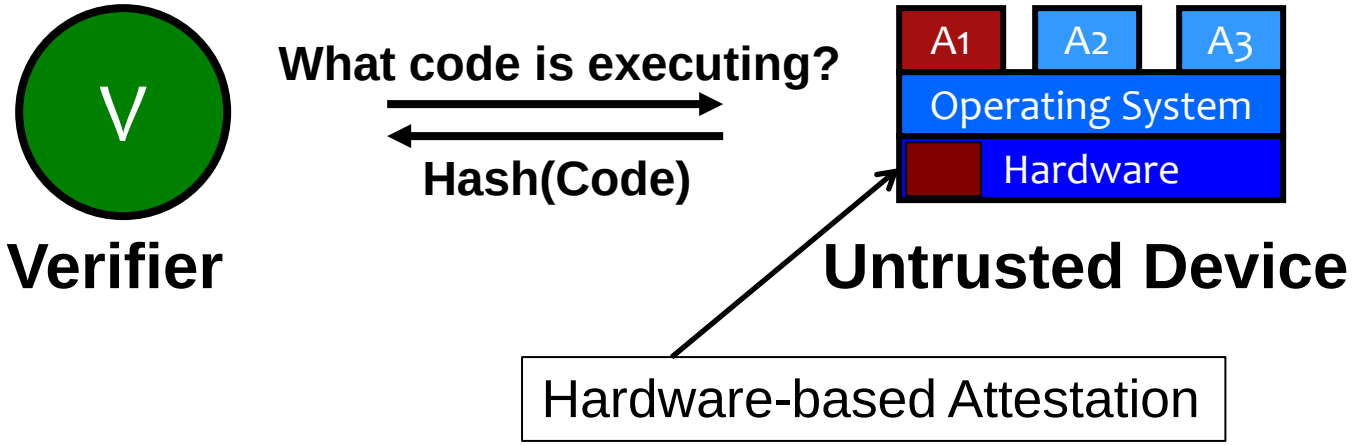
“Secure” Boot via certification chain

- **Each component of the boot process verifies following component to be loaded**
  - Example: digital signature on each boot component; boot loader contains public key and verifies digital signature on OS, etc.
- **Advantages**
  - Only approved software can be loaded (assuming no vulnerabilities)
- **Disadvantages**
  - Adversary only needs to compromise single component
  - Entire system is in TCB, no isolation
- **Verdict: Entire system is still part of TCB, Relatively weak security guarantee**



Remote Attestation

- **Attestation enables verifier to establish trust in untrusted device**
  - Attestation tells verifier what code is executing on device
  - If intended code is executing on untrusted device, verifier can trust its operation



Hardware-based Attestation

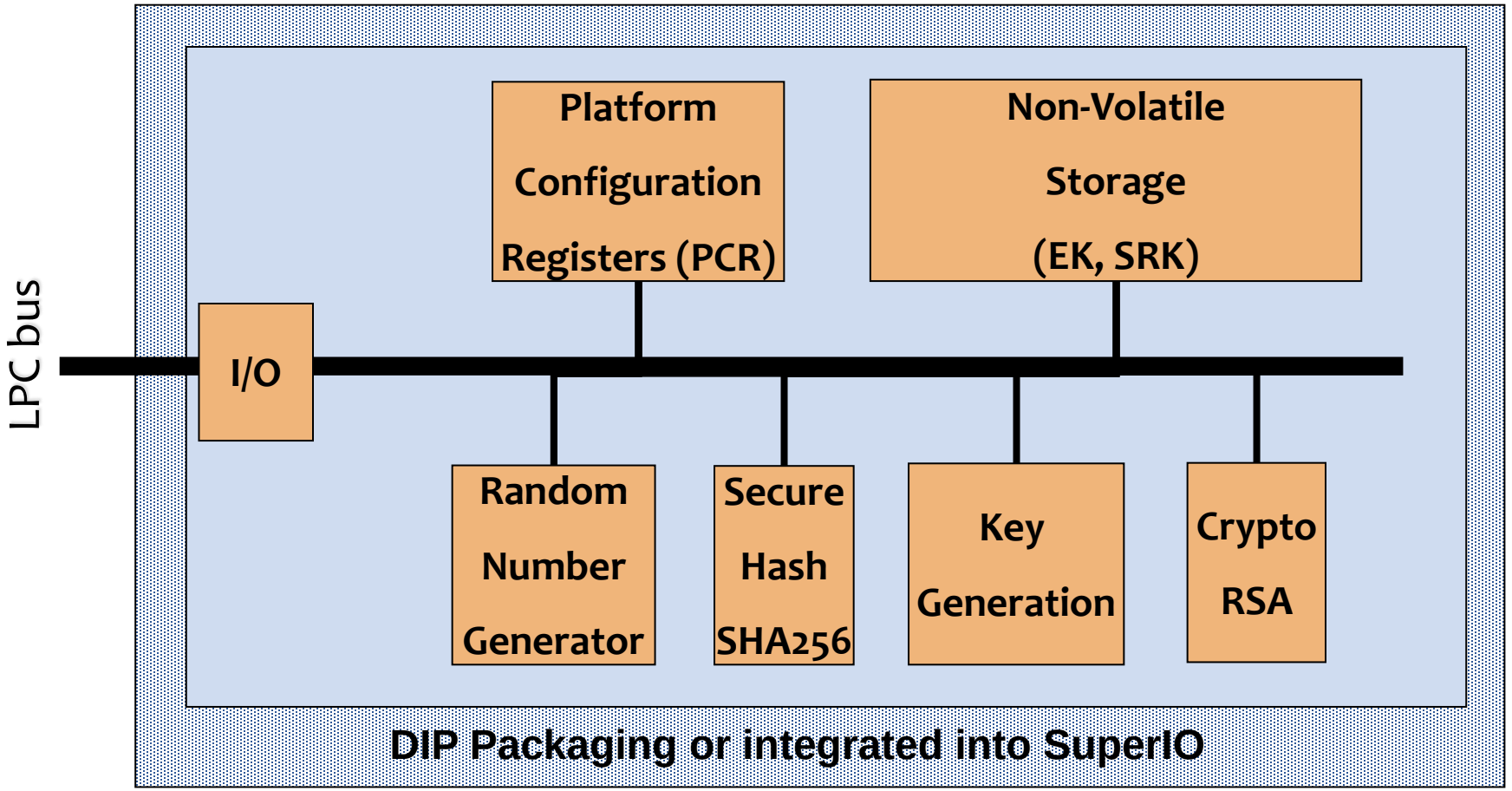
- **Leverages hardware support for attestation**
- **Why hardware?**
  - Can be made tamper-resistant easily
- **Trusted Computing Group (TCG) proposed Trusted Platform Module (TPM) chip**
  - Already included in many platforms
    - Most newer Intel/AMD chips have TPM
    - Windows 11 “demand” TPM 2.0 support

- **Modern microprocessors provide special instructions that interact with TPM chip**
  - AMD SVM: SKINIT instruction
  - Intel TXT/LT: GETSEC[SENTER] instruction

Trusted Computing Group (TCG)

- **Open organization to "develop, define, and promote open standards for hardware-enabled trusted computing and security technologies."**
  - Desktops, laptops, servers, cell phones, PDAs, …
  - Industry consortium by AMD, IBM, Intel, HP, Microsoft, …
- **These secure platform primitives include**
  - Platform integrity measurements
  - Measurement attestation
  - Sealed storage
- **Can enable**
  - Trusted boot (not secure boot)
  - Attestation, which lets a remote verifier check integrity of software
- **Goals:**
  - Ensure absence of malware
  - Detect spyware, viruses, worms, …

Trusted Platform Module (TPM)



TPM Non-Volatile Memory

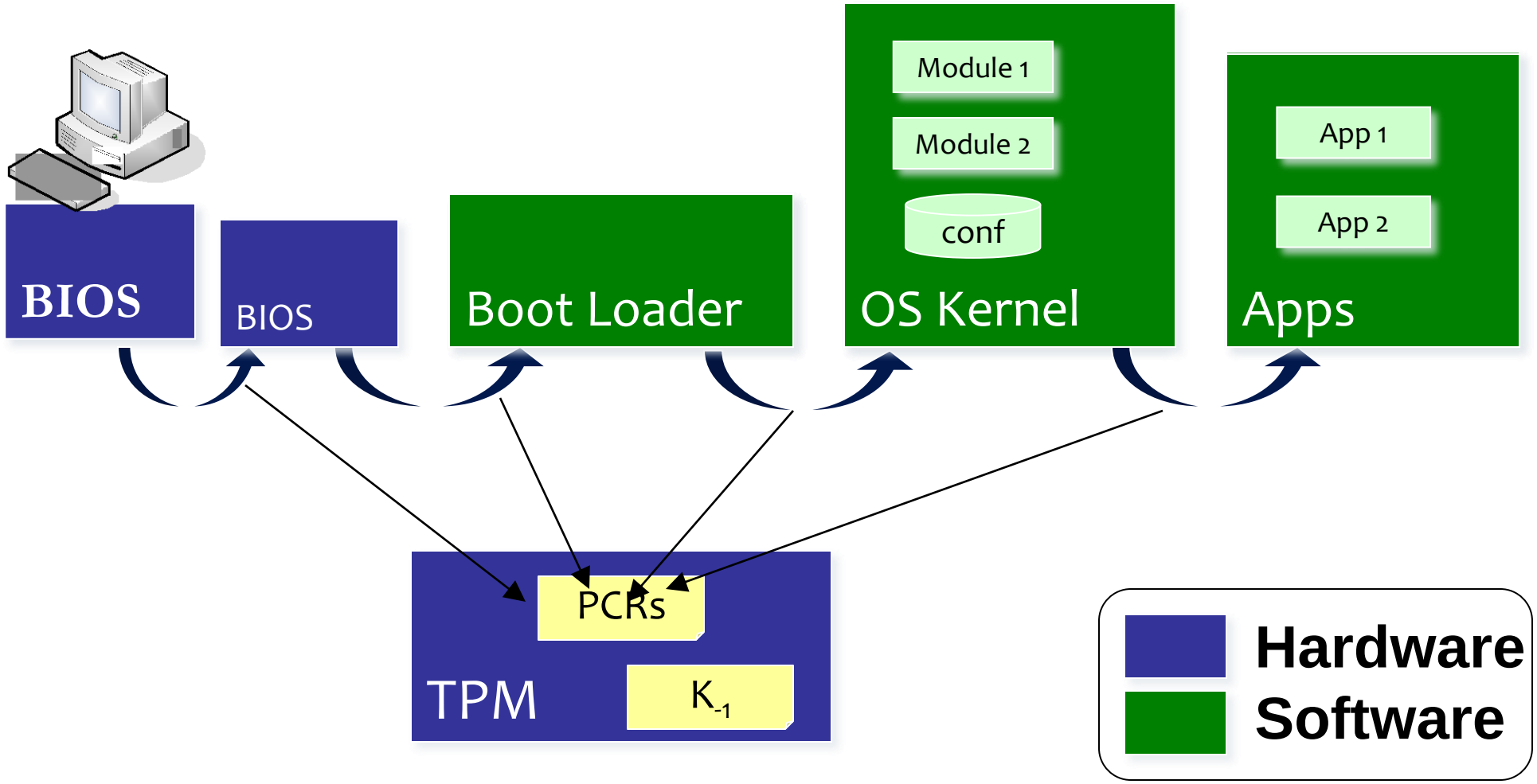
- **Endorsement key (EK) (2048-bit RSA)**
  - Public/private key pair
  - Unique to each chip, created at manufacturing time, cannot be changed
- **Manufacturer certificate, e.g., $\{PubEK\}_{K^{-1}_{IBM}}$**
- **On-chip storage for Storage Root Key K^{-1}_{SRK} (2048-bit RSA)**
  - Created using TPM_TakeOwnership(OwnerPassword, …)
  - Can be cleared later with TPM_ForceClear from BIOS
- **OwnerPassword (160 bits)**

Basic TPM Functions

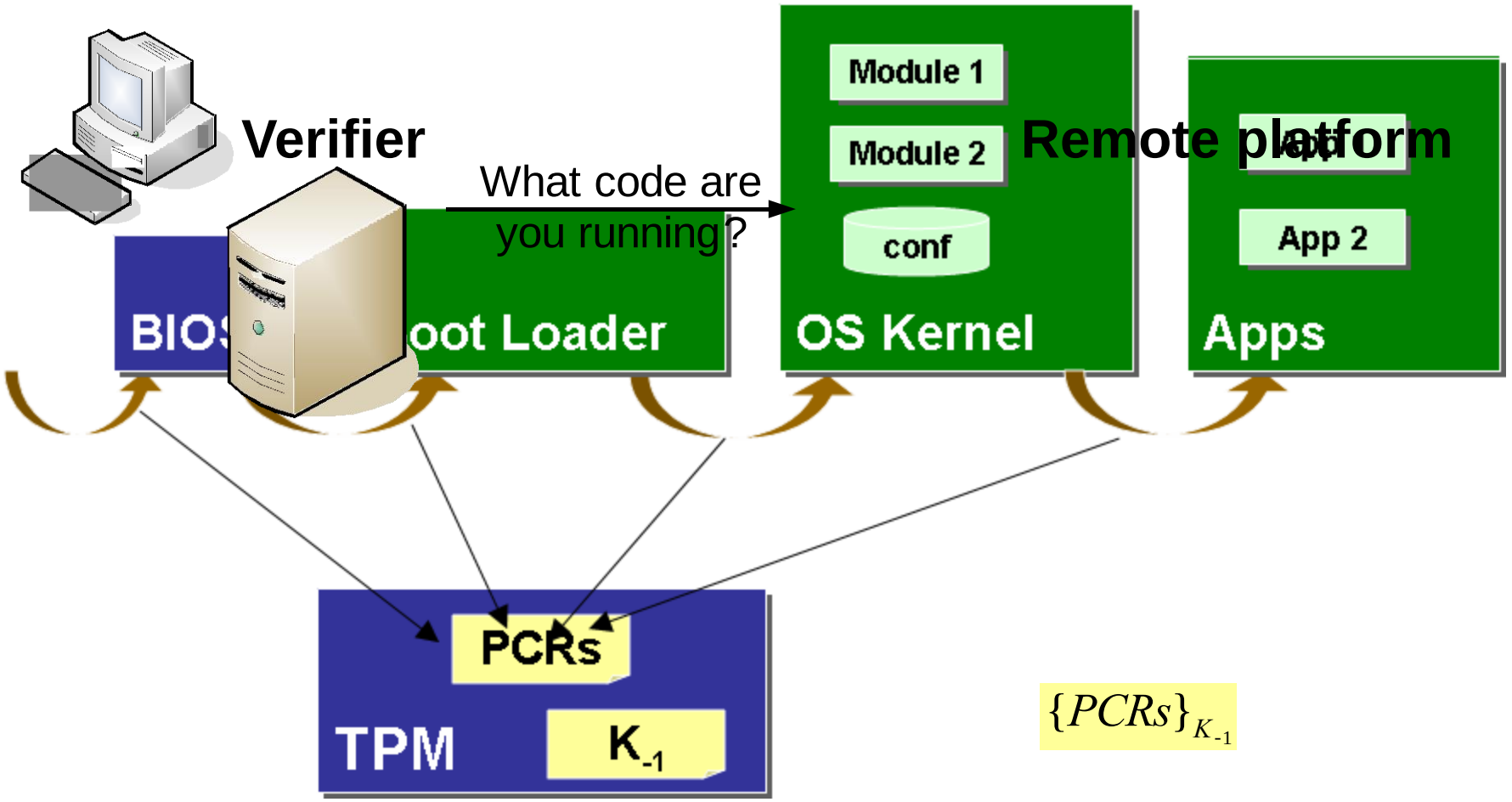
- **Integrity measurement: PCRs store integrity measurement chain**
  - PCRs initialized to default value at boot time
  - TPM_Extend(n, data)
    $PCR[n]_{new} = SHA256(PCR[n]_{old}||SHA256(data))$
- **Remote attestation (PCRs + AIK)**
  - Attestation Identity Keys (AIKs) for signing PCRs
  - Attest to value of integrity measurements to remote party
- **Sealed storage (PCRs + SRK)**
  - Protected storage + unlock state under a particular integrity measurement (data portability concern)

Basic TCG-Style Attestation

BIOS calls TPM_Startup (ST_CLEAR) initialize PCRs to 0



Basic TCG-Style Attestation



Problem?

- What would go wrong if TPM_Startup (ST_CLEAR) could be called at any time after boot?

Sealed Storage

- **TPM_seal: encrypt files based on PCR values (into a blob)**
- **TPM_unseal: decrypt blob**
  - Only succeed if the PCR values at the time of unseal is the same as the PCR values in the blob
- **PCR values ensure that data can only be decrypted by "authorized" programs**
  - If OS kernel (application) is changed, PCR values will not match

BitLocker

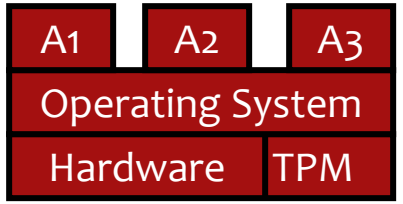
- **A full disk encryption application included with Windows Vista and later**
- **Volume Master Key (VMK) encrypts disk volume key**
- **VMK is sealed (encrypted) under TPM SRK using PCR values**
  - BIOS, extensions, and optional ROM (PCR 0 and 2)
  - Master boot record (MBR) (PCR 4)
  - NTFS Boot Sector and block (PCR 8 and 9)
  - NTFS Boot Manager (PCR 10), and –
  - BitLocker Access Control (PCR 11)

BitLocker—System Updates

- **Measurement may change for legitimate reasons**
  - BIOS updates
  - OS updates
- **Have to suspend BitLocker protection before update**
- **Or use recovery key after updates**
- **Otherwise, suffer loss of data**

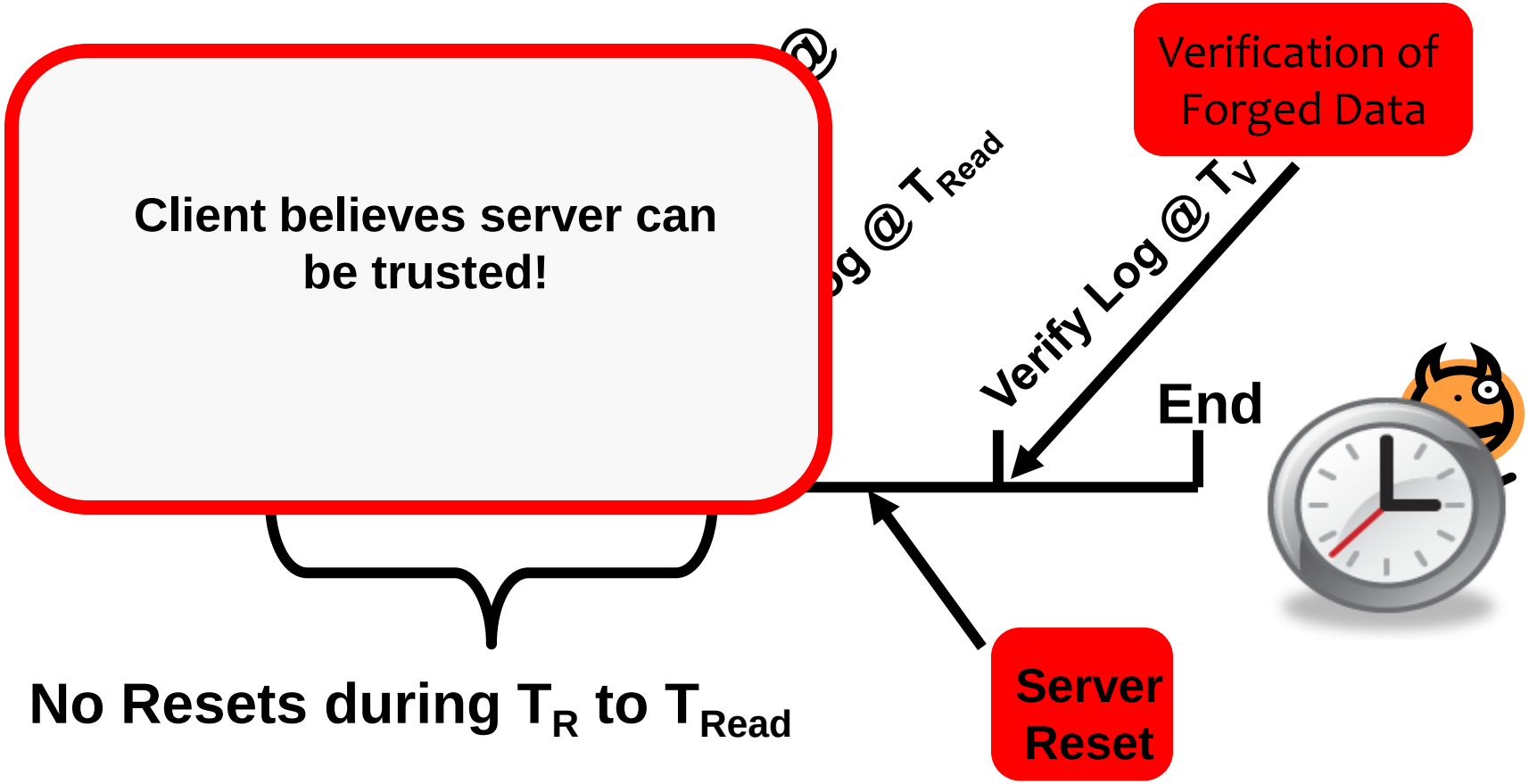
Shortcomings of TCG-style Attestation

- **Integrity measurements are done at load-time not at run-time**
  - Time-of-check-time-of-use (TOCTOU) problem
  - Cannot detect any dynamic attacks!
- **Coarse-grained, measures entire system**
  - Accumulates hundreds of integrity measurements
  - Every host is different, different firmware versions, different drivers, different patches, different apps, different spyware, …
  - What does a PCR mean in this context?
  - TCB includes entire system!
- **No guarantee of execution**
- **Requires special hardware: TPM chip**



Time of Check, Time of Use (TOCTOU)

- Reset attack possible after read of log



Policy Issues

- **Can TPMs be used for malicious purposes?**
  - Could software vendor control all applications that are executed?
  - Could content provider have total control over how we use data? Fair use?
- **TPMs can enhance security of computer systems**
  - Should government require use of TPMs?
  - Other issues?

TCG Controversy

- TCG is considered very controversial because it potentially allows content providers to control clients (DRM enforcement)
- This takes away the freedom of the user to use the system as it sees fit (it can be used to lock-out GPL software)
- A privacy concern is that TCG can be used to track users
- Are these concerns valid?

DRM Example

- **Downloading a music file**
- **Remote attestation**
  - Refuse to play except on specific music player
    - Windows Media Player
  - Sealed storage prevent opening file from another player

More Benign Examples

- **Prevent cheating in online games**
  - Players modify game in order to cheat
  - Remote attestation can verify all players connected to game server are running an unmodified copy
- **Virus and spyware**
  - Users identify applications modified by third parties that add spyware to software
    - Malicious version of Outlook that contains spyware

TCG Published Best Practices Document

- **Design, Implementation, and Usage Principles for TPM-Based Platforms**
  - May 2005
  - Clearly states that
    - "Use of coercion to effectively force the use of the TPM capabilities is not an appropriate use of TCG technology"
  - Compliance is voluntary
  - Can government enforce compliance?

Arguments for TCG

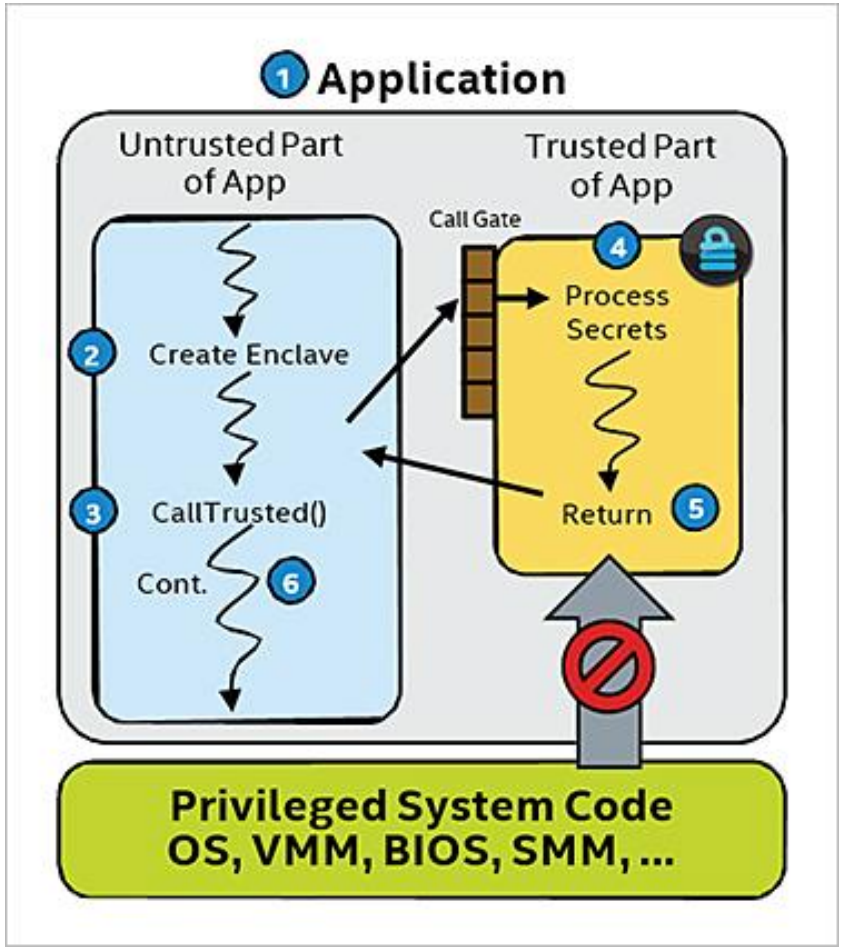
- TCG designers were aware of such concerns and TPM focus is that machine owner is in full control (individual or enterprise). Neither service nor content provider control TPM
- TPM does not lock-out software, it merely measures it (if enabled)
- It does allow a service/content provider to not service the machine if attestation statement does not meet its requirements
- Is this very different from current mechanism where each browser sends browser name, OS, version to web server?

Intel SGX

Intel SGX

- **Intel Software Guard Extensions (SGX): extensions to Intel processes with additional instructions**
  https://software.intel.com/en-us/sgx/details
- **Goal:**
  - **Enclave:** protect the secrecy and integrity of program and data
  - **Remote attestation:** prove program is running in enclave
- **Applications:**
  - Keep secret key in enclave
  - Running client's program in enclave in the cloud

Enclave



- Application is divided into trusted and untrusted parts
- Untrusted part create enclave and execute trusted parts in enclave
- Enclave has restricted entry and exit points
- Data written to memory is encrypted
- Only program in enclave can access data in the enclave

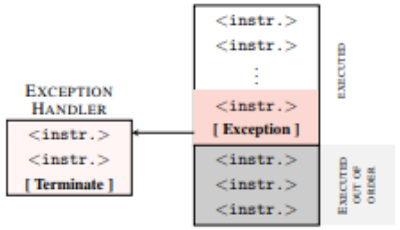
Hardware issues

Meltdown

- **Meltdown uses out-of-order execution to read arbitrary-kernel memory (attack on Intel chips)**
  - Speculative execution of following instructions

```
1 raise_exception();
2 // the line below is never reached
3 access(probe_array[data * 4096]);
```

Listing 1: A toy example to illustrate side-effects of out-of-order execution.



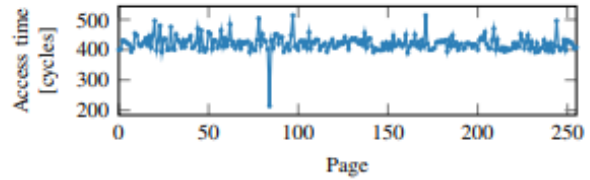


Figure 4: Even if a memory location is only accessed during out-of-order execution, it remains cached. Iterating over the 256 pages of `probe_array` shows one cache hit, exactly on the page that was accessed during the out-of-order execution.

https://meltdownattack.com/meltdown.pdf

Meltdown Steps

1. Load inaccessible memory location into register
2. Access cache line
3. Flush+Reload to determine accessed cache line

```
1 ; rcx = kernel address
2 ; rbx = probe array
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax]
```

Listing 2: The core instruction sequence of Meltdown. An inaccessible kernel address is moved to a register, raising an exception. The subsequent instructions are already executed out of order before the exception is raised, leaking the content of the kernel address through the indirect memory access.

https://meltdownattack.com/meltdown.pdf

Meltdown Why Does This Work?

- The key is that Intel (but not ARM or AMD) don't squash under-privileged TLB (translation look-ahead buffer) hits. The load executes, and only actually faults when the faulting load tries to retire.

Spectre Attack

- **Similar to Meltdown, but also for AMD and ARM processors**
  - Utilizes branch prediction to achieve speculative execution
  - No privilege escalation vulnerability on Intel CPUs
- **Mis-train CPU to get branch prediction wrong**

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```

Listing 1: Conditional Branch Example

https://spectreattack.com/spectre.pdf

Spectre Attack

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```

Listing 1: Conditional Branch Example

- Value of x maliciously chosen so array1[x] is secret byte in victim's memory (also user mode program)
- Make sure array1_size isn't cached so speculatively try to read from array2
- Check cache and win!

https://spectreattack.com/spectre.pdf

Take Away Slide

- **TPM aims to confirm integrity of the code running**
  - Avoiding Trojans, malicious software
  - Making sure a remote party is behaving properly
  - Initially used for integrity of core hardware (BIOS, bootloaders)
  - Extended to applications
- **Muddy waters from a policy standpoint**
  - Can result in vendor lock-in?
    - Apple example
  - Entertainment industry **very** interested in using this for DRM
- **TPM itself doesn't do any enforcement**
- **Hardware errors can cause serious security problems**
  - Meltdown, Spectre