CSE 5095 & ECE 4451 & ECE 5451 – Spring 2017

Lecture 3a

- Slide deck originally based on some material by Chenglu and Masab Ahmad during ECE 6095 Spring 2017 on Secure Computation and Storage, a precursor to this course

Attacks Secure Processors

Marten van Dijk
Syed Kamran Haider, Chenglu Jin, Phuong Ha Nguyen

Department of Electrical & Computer Engineering
University of Connecticut

UCONN

Outline

Hardware attacks

- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Secure Processors/Coprocessors

- **Industry Solutions:**
- IBM 4765 Secure Coprocessor
- ORWL
- Microsoft NGSCB
- IBM SecureBlue++
- ARM TrustZone
- SGX
- **Boot Security:**
- The Trusted Platform Module (TPM)
- Intel's Trusted Execution Technology (TXT)
- **Academic Solutions:**
- XOM
- Aegis
- Bastion
- Ascend
- CHERI
- Sanctum

Physical Attacks

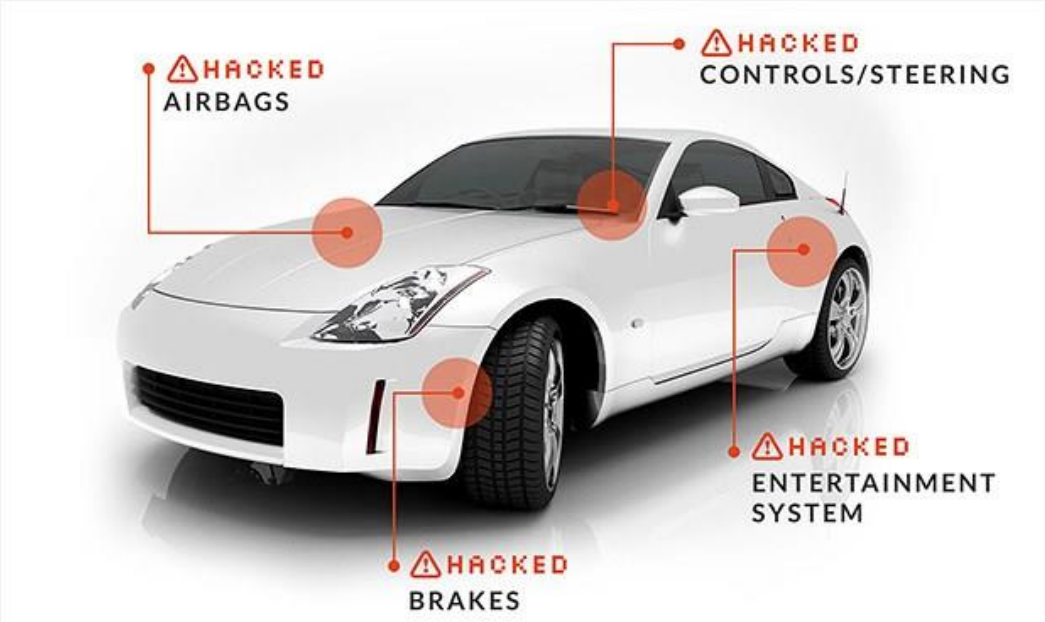
- Port Attacks
  - E.g. cold boot attack, where the attacker plugs in a USB flash drive into the victim's case and causes the computer to boot from the flash drive, whose malicious system software receives unrestricted access to the computer's peripherals.

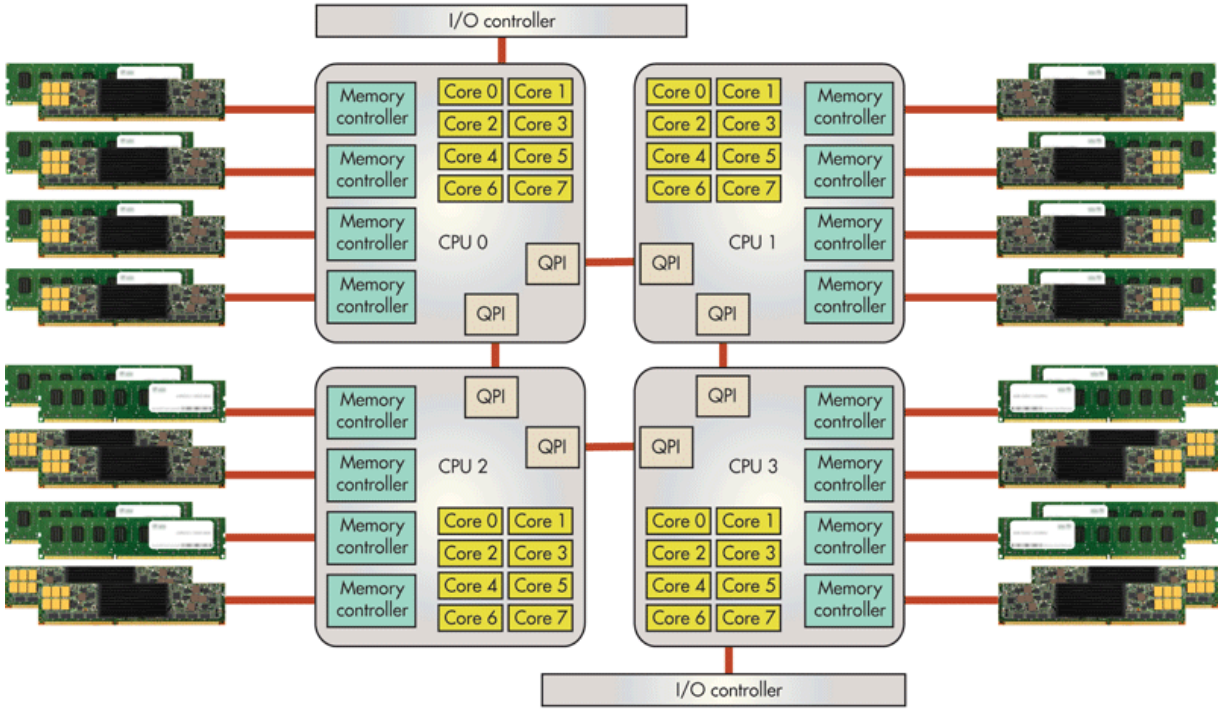
- Bus Tapping Attacks
  - Monitoring Attacks, Active Attacks, Replay Attacks

- Chip Attacks
  - Physically look/attack into the chip
  - Cutting/repairing silicon structures (security fuses, traces, etc.) → access to secrets restored

http://www.martybucella.com/

Bus Tapping Attacks





- An attacker taps a bus and snoops in on information
  - Can potentially insert information as well, causing un-required behavior

Intel Corporation. Intel R 64 and IA-32 Architectures Software Developer's Manual. Number 253669-033US. December 2009.

- An attacker can potentially extract information just by observing the DRAM address sequences

Chip Attacks

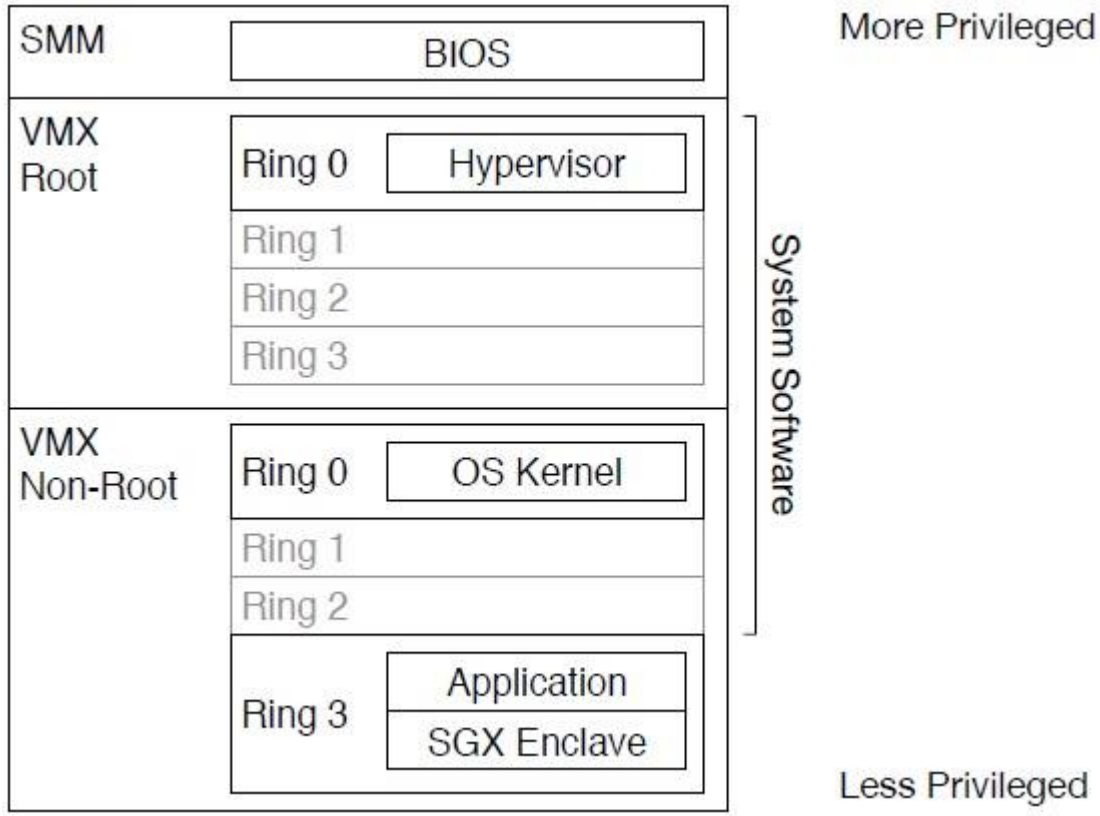
- Reduce temperature of the chip
  - Causes chips to go in “hibernation” mode with vulnerabilities
  - Security modules turned off
  - Potential DRAM writes possible
- Other possible attacks include tampering with hardware fuses and wires

J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, Lest we remember: Cold boot attacks on encryption keys, in Proceedings of the 17th USENIX Security Symposium, San Jose, CA, 2008, pp. 45–60.

Privileged Software Attacks

- System Management Interrupts (SMIs) and Modules (SMMs)
  - Handled by a SM module that has high privileges
  - Handles keyboard presses and mouse taps
  - Exploited Multiple times
  - Compromised Intel TXT (see later slides)

- Heavily emphasized in Intel SGX
  - Hypervisor control needed – Isolation
  - OS also isolated from secure enclaves



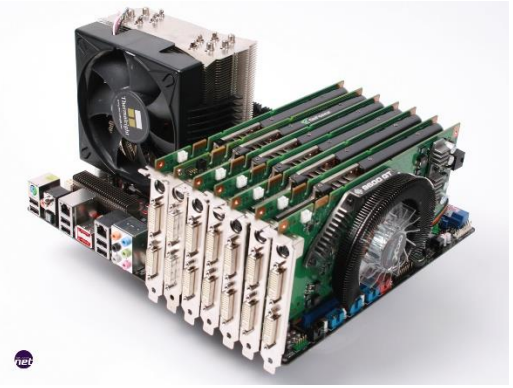
Software Attacks on Peripherals

- PCI (Peripheral Component Interconnect) Express Attacks
- DRAM Attacks
- The Performance Monitoring Side Channels
- Attacks on Boot Firmware and Intel ME
- Accounting for Software Attacks on Peripherals
- Malicious Peripherals

PCI Express Attacks

- PCI bus allows a device to do a direct memory access (DMA) to/from the DRAM
  - Attacker changes critical data
  - GPUs exposed this way

Single Computing Node



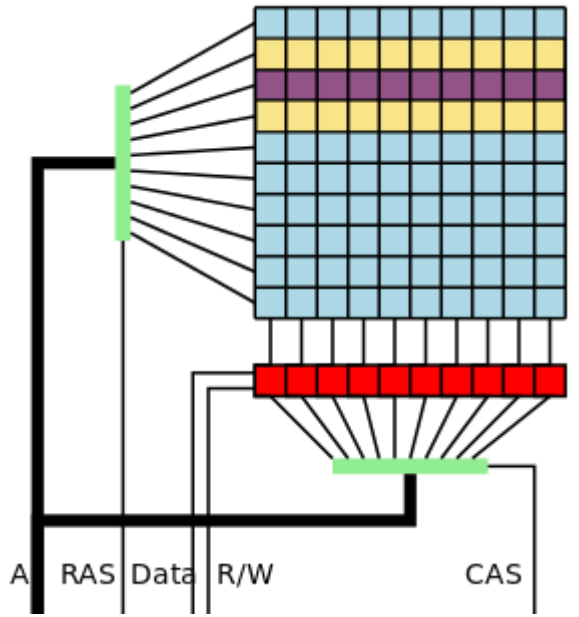
- An early implementation of Intel TXT gets compromised this way
  - The TXT reserved DRAM region is accessed by a compromised peripheral over DMA
  - It is fixed by adding security checks in DMA

Multi-Node Supercomputer connected over PCI buses

http://www.ibm.com/smarterplanet/us/en/ibmwatson/

DRAM Attacks

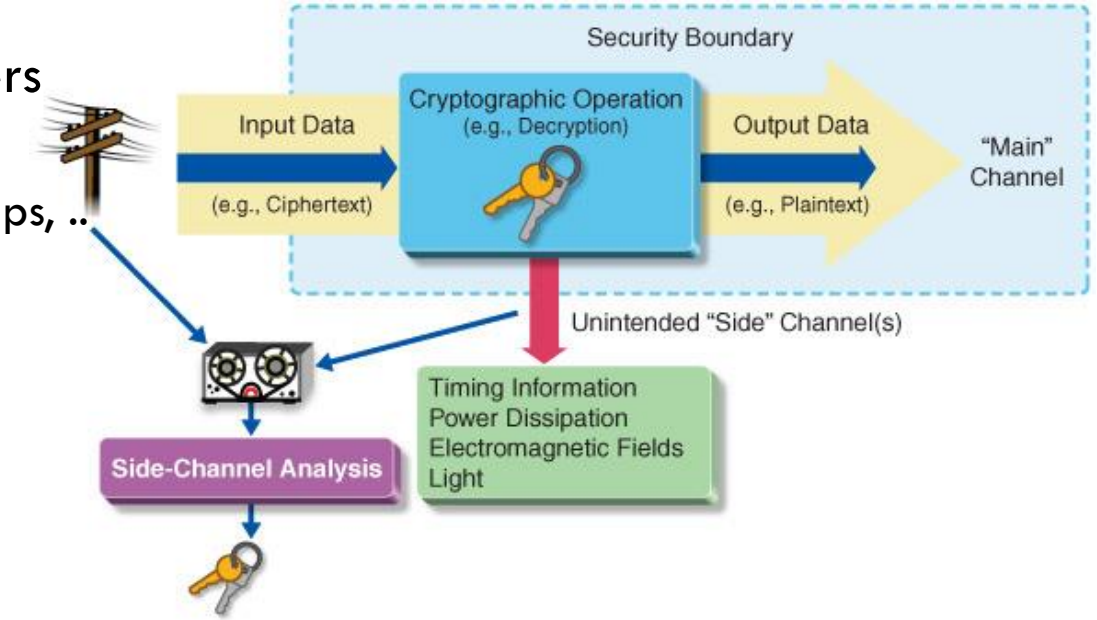
- Huge Class of attacks
  - **E.g. RowHammer** Attack
    - Bit flips on DRAM refresh
    - OR current leaks in memory can allow privileged access
    - Attacker then reads page table bits if leaked
    - Modification of page tables possible
- Isolation of page tables required in hardware
  - Hash checks

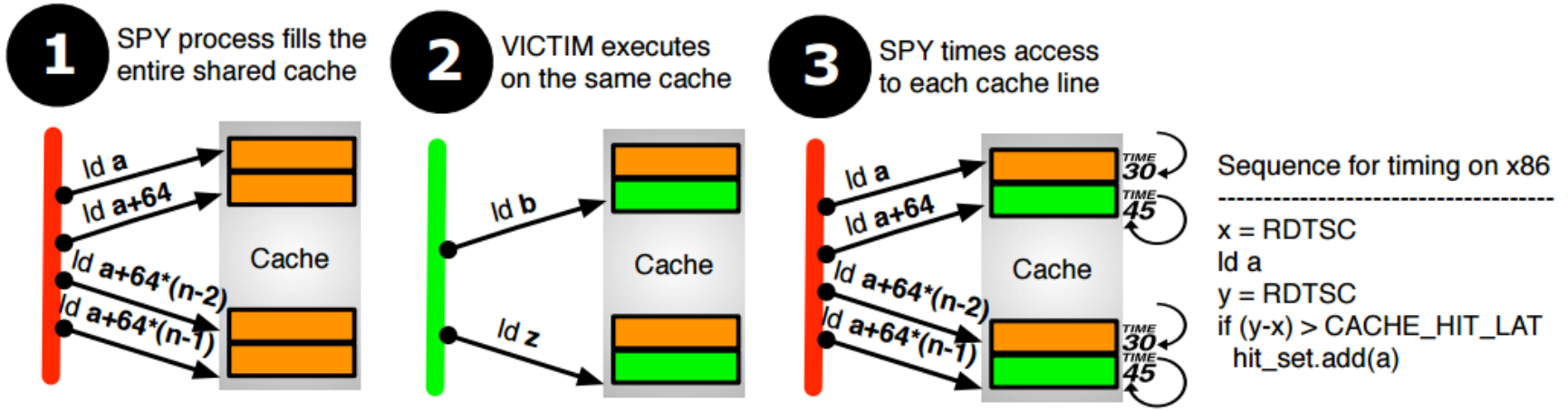


Bits leaked to adjacent cells

The Performance Monitoring Side Channels

- An attacker can gain access to performance counters
  - Read Model specific registers (MSRs)
  - Other Side channel attacks: Power analysis attacks, Heat maps, ..
- Isolation of performance counters required
  - The victim thread must not be able to access the counters





F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee. Last-Level Cache Side-Channel Attacks are Practical. In Proceedings of the 36th IEEE Symposium on Security and Privacy (S&P'15), 2015.

Attacks on Boot Firmware and Intel ME

- An attacker can use the highly privileged system management mode (SMM) to read/write device firmware
  - Such a mode can read/write anywhere and can access any peripheral for **debugging purposes**

- Intel management engine (ME) reads contents from the same flash as the above firmware, and has high privileges
  - A compromised ME would leak most of the powers that come with installing active probes on the DRAM bus, the PCI bus, and the System Management bus (SMBus), as well as power consumption meters

- Isolation and secure update of firmware required

Malicious BIOS

- A BIOS can become malicious in one of two ways:
  - Due to being maliciously written by the vendor, i.e. a backdoored BIOS, or
  - Due to somebody being able to later modify the original (benign) BIOS with a rogue one, either due to:
    - Lack of proper reflashing protection implemented by the original BIOS
    - In case of a BIOS that does apply proper reflashing protection: by exploiting subtle flaws in the original BIOS and getting code execution before the reflashing/update or SMM locks are applied
    - If we include physical attacks in our threat model: by an attacker who is able to connect an SPI programmer to the SPI chip and replace the firmware content stored there.
- Solutions: Digital signatures for updating BIOS, firmware measurement by TPM, Intel Boot Guard checks the signature of boot block to whitelist trusted firmware

Malicious Peripherals

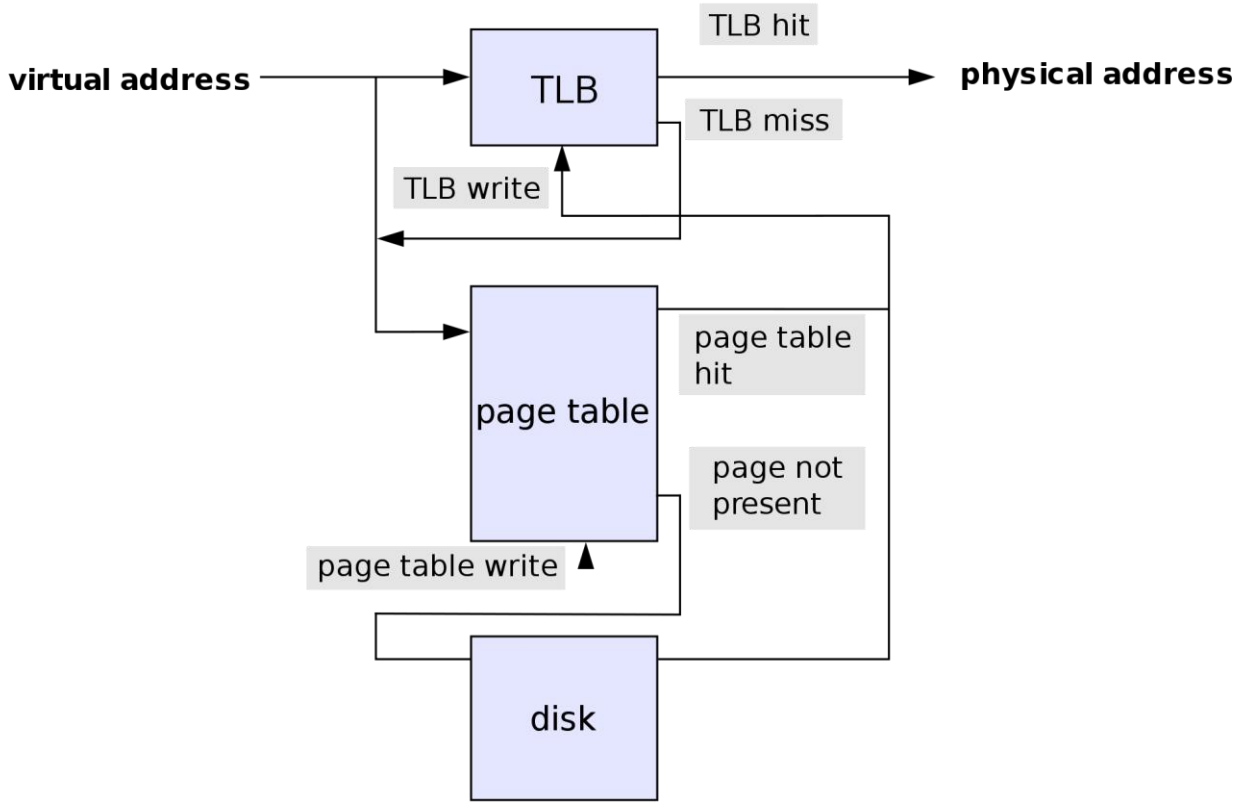
- Malicious USB devices can trigger the download of device drivers, record user's keystroke and impersonate a user.
- A malicious graphic system can record all the documents that the users have seen on the screen, even it is encrypted in the storage.
- A malicious disk controller can provide modified code for the OS kernel or hypervisor during the boot sequence, and thus compromise it. Possibly DMA attacks.
- A compromised audio card can control the microphone and listen to the users' conversation. It can also control the speaking to establish a convert channel to send out information from an "air-gapped" computer via an inaudible frequency.
- Using the ME and SMMs, an attacker can attack device peripherals
  - E.g. attack a wifi modem and transmit malicious stuff as a bot
  - However Intel's ME and SMM features are largely undocumented

Address Translation Attacks

- Passive Attacks
- Straightforward Active Attacks
- Active Attacks using Page Swapping
- Active Attacks based on TLBs

Passive Attacks

- Address translation used for page swapping
- An untrusted page table manager can swap pages using page faults and leak information
- Successful practical attacks on SGX!
  - Image inferred even though it was isolated by SGX
  - Intel's response (by Matt Hoekstra and Frank McKeen) puts blame on software developers
  - https://software.intel.com/en-us/blogs/2015/05/19/look-both-ways-and-watch-out-for-side-channels



Straightforward Active Attacks

- An attacker modifies page tables physically or via a vulnerability in a memory manager

- An isolated application then makes a secure access to memory, only to jump and execute to a wrong and malicious location

Active Attacks using Page Swapping

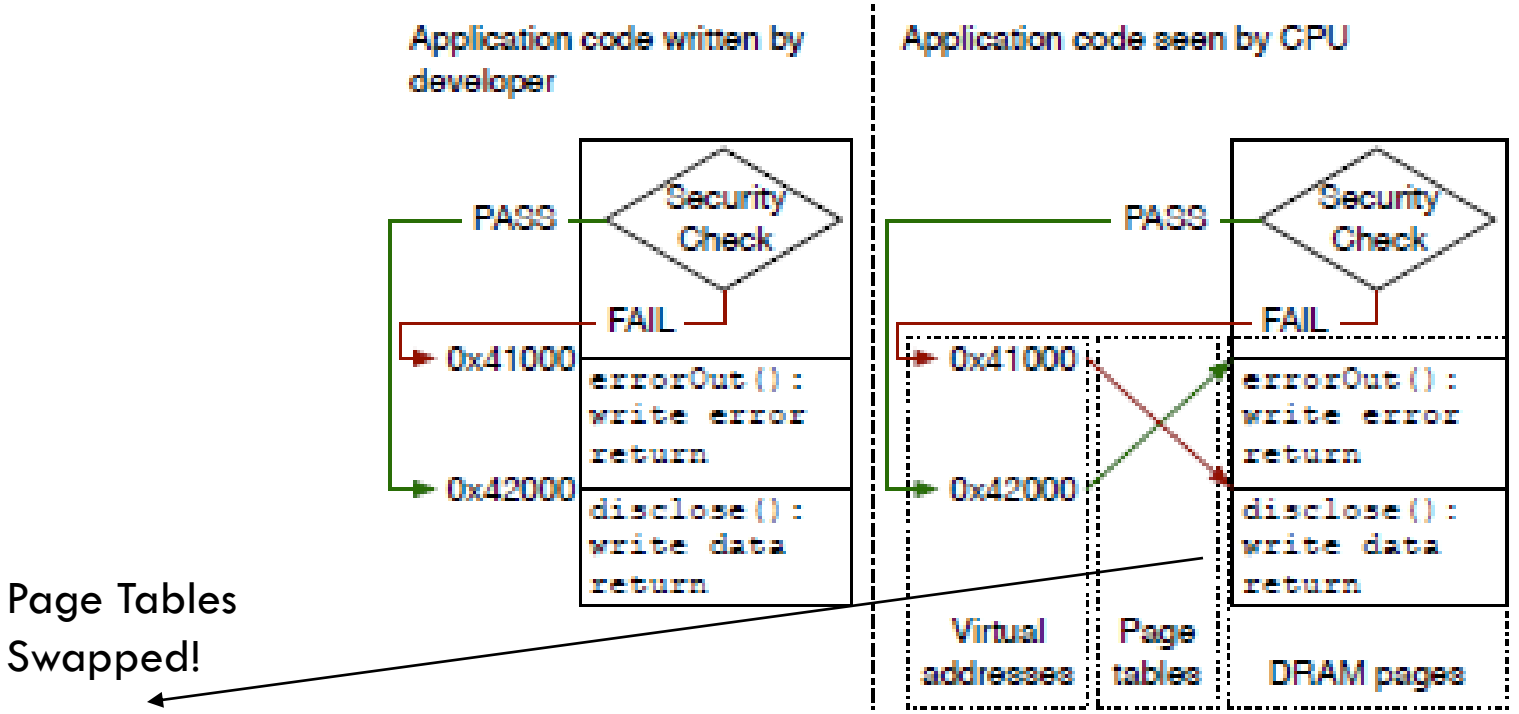
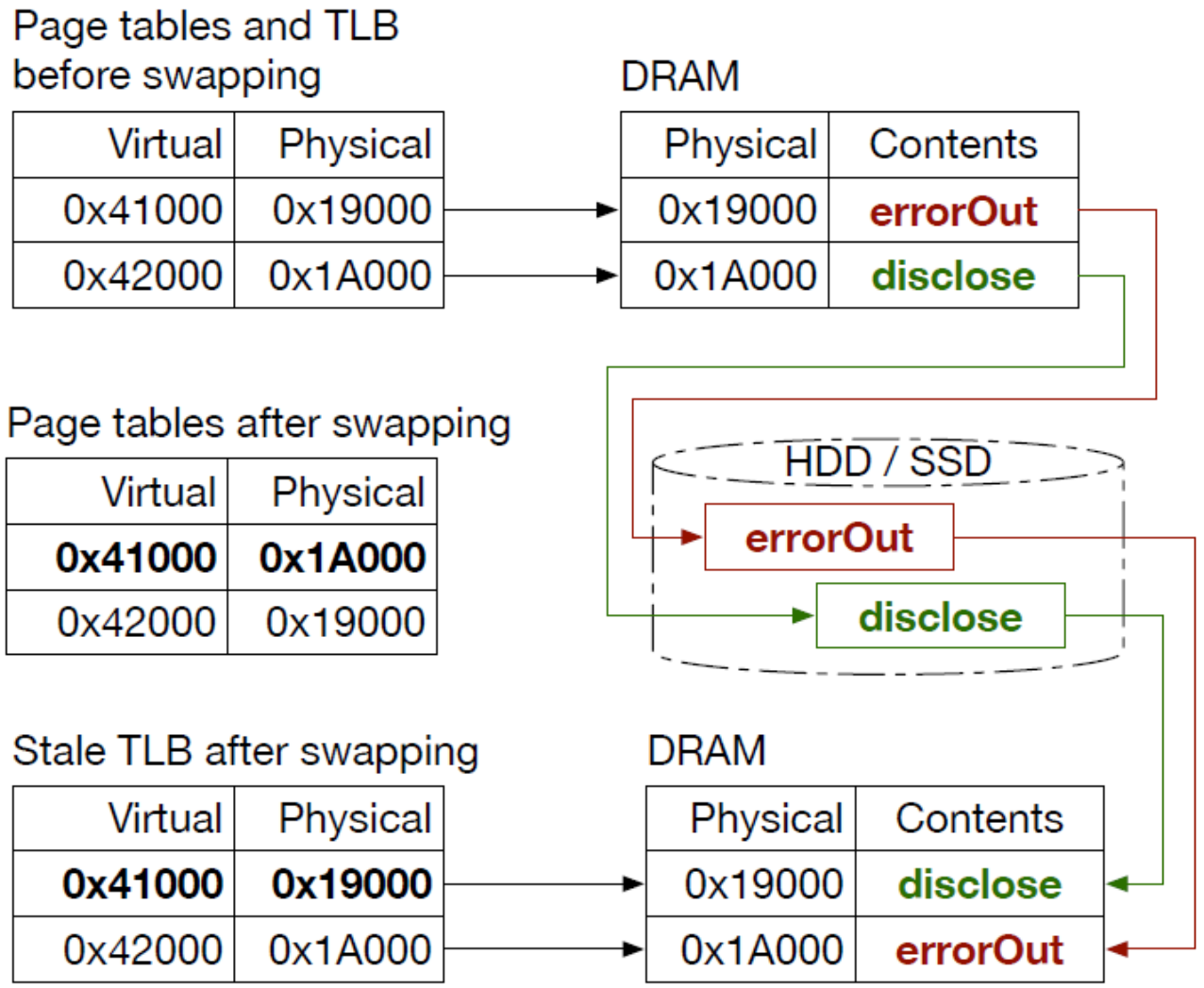


Figure 53: An example of an active memory mapping attack. The application's author intends to peform a security check, and only disclose a piece of sensitive information if the check passes. Malicious system software maps the virtual address of the procedure called when the security check fails to a DRAM page that contains the procedure that discloses the sensitive information, which is supposed to be called when the security check passes.

Active Attacks based on TLBs

- Page table is cached in TLBs (Translation Look-aside Buffers), and the system software is responsible for invalidate TLB entries when the page is evicted from memory
- A malicious system software can evict the two pages and swap them without invalidating the TLB entries. Then the attacker is able to access different regions.
  - Assuming that the TLB is not subject to security checks



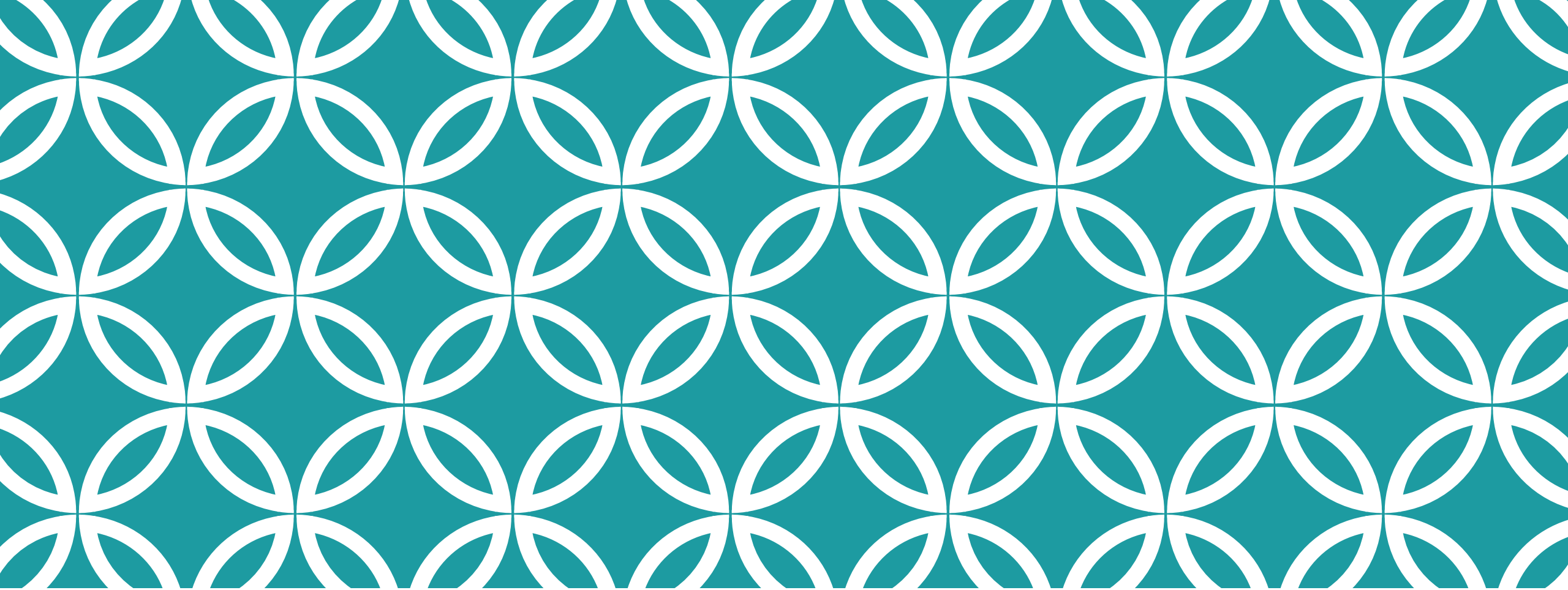
Outline

Hardware attacks

- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Secure Processors/Coprocessors

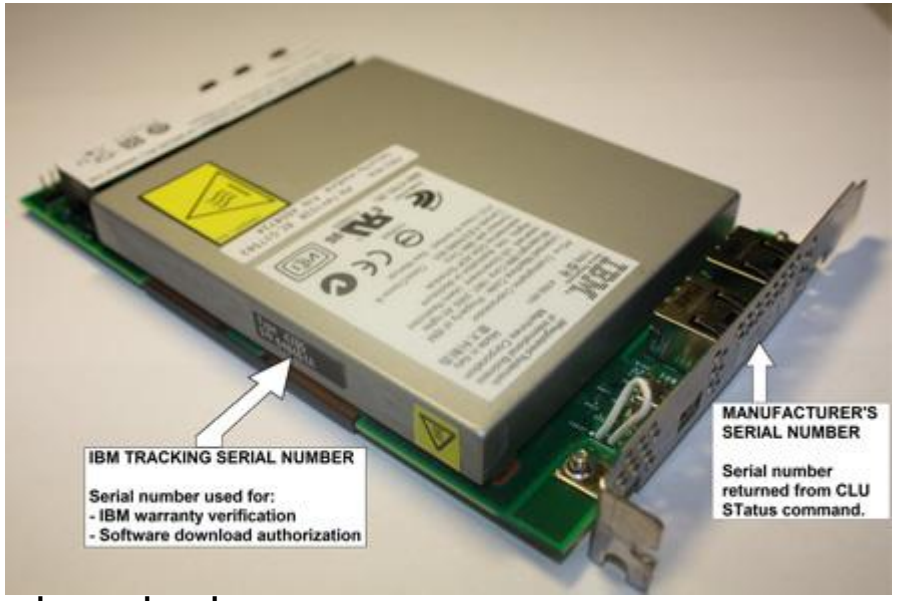
- **Industry Solutions:**
- IBM 4765 Secure Coprocessor
- ORWL
- Microsoft NGSCB
- IBM SecureBlue++
- ARM TrustZone
- SGX
- **Boot Security:**
- The Trusted Platform Module (TPM)
- Intel's Trusted Execution Technology (TXT)
- **Academic Solutions:**
- XOM
- Aegis
- Bastion
- Ascend
- CHERI
- Sanctum



Industry Secure Processors/Coprocessors/Computers

IBM 4765 Secure Coprocessor

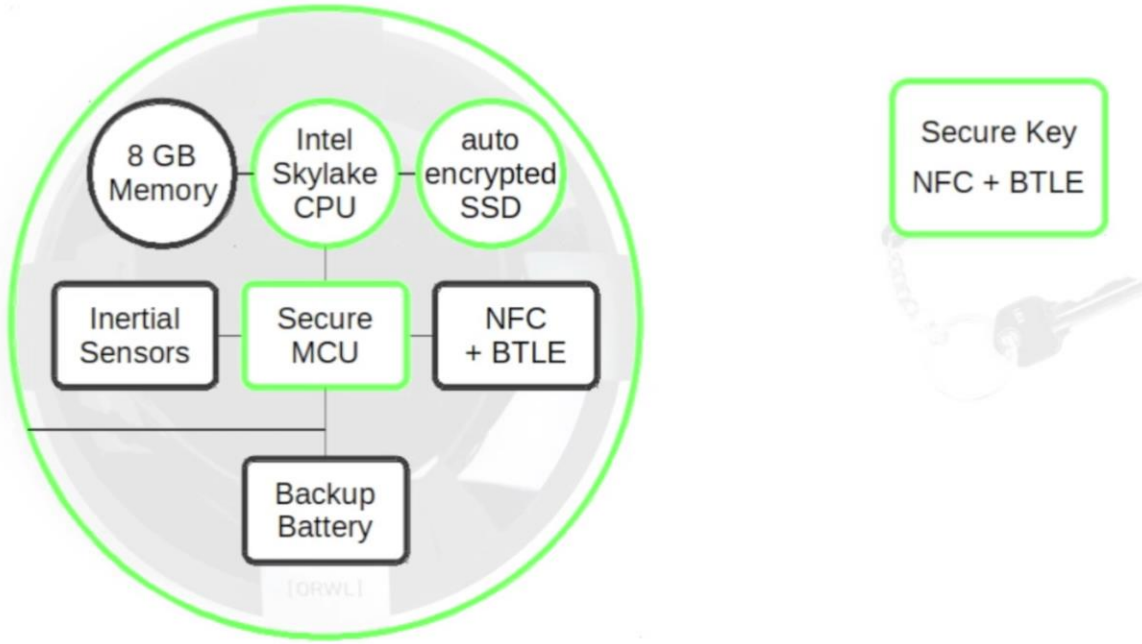
- Secure coprocessor contains an entire computer system: a CPU, a cryptographic accelerator, caches, DRAM, and an I/O controller within a tamper-resistant environment.
- Tamper-resistant environment contains a Faraday Cage and an array of sensors that can detect tampering attempts. The secret is destroyed once a sensor detects attack.
- 4765 meets the rigors of FIPS 140-2 Level 4
- Disadvantage: Too expensive ($14,408 in 2011)



https://www-03.ibm.com/security/cryptocards/pciecc/4765SerialNumbers.shtml
http://www-01.ibm.com/common/ssi/rep_ca/8/897/ENUS110-158/ENUS110-158.PDF

ORWL

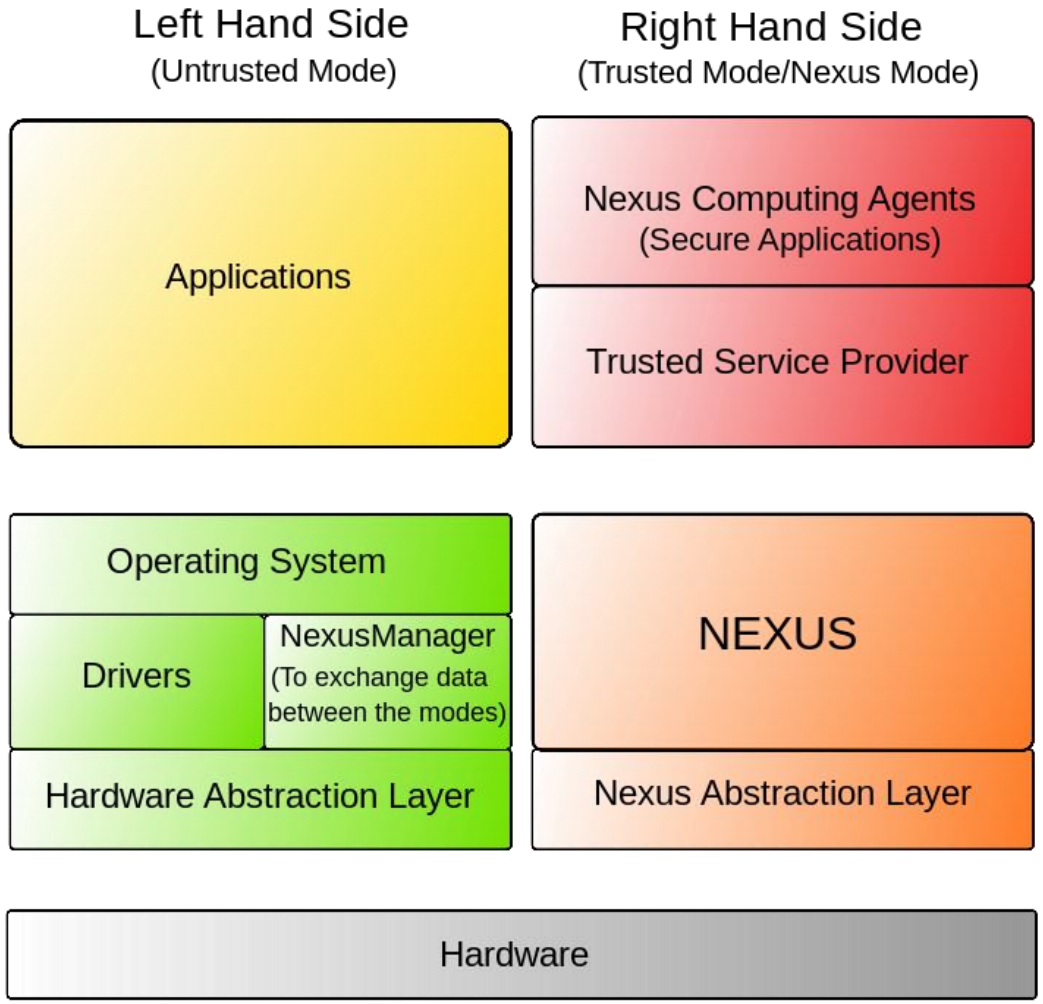
- It is marketed as The First Open Source, Physically Secure Computer.
- They are actively pursuing getting ORWL certified as a cryptographic module under a US National Institute of Standards and Technology (NIST) Federal Information Processing Standard known as FIPS 140-2.
- Intel Skylake CPU which contains SGX is used to prevent software attacks.
- The outer shell of ORWL and the sensors are used to prevent against physical attacks (like IBM 4765).



https://www.crowdsupply.com/design-shift/orwl

Microsoft NGSCB

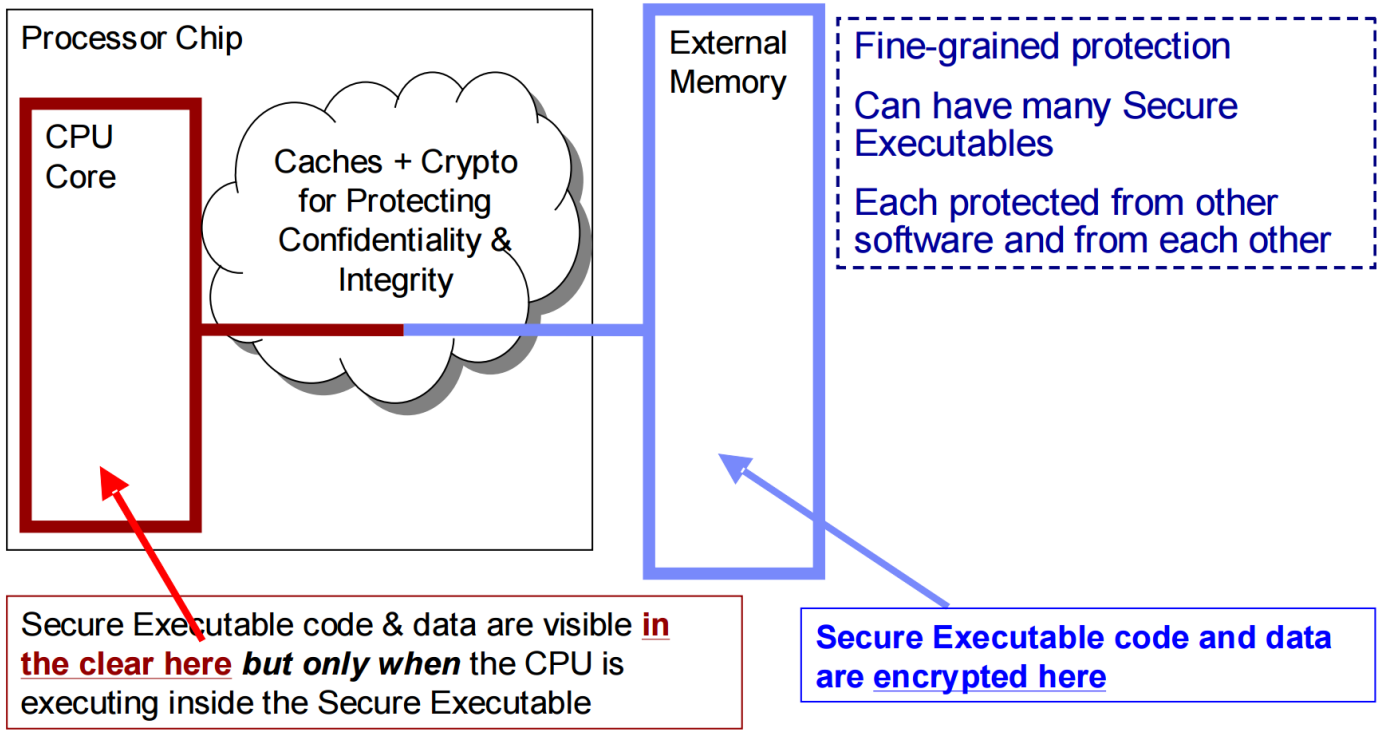
- NGSCB = Next-Generation Secure Computing Base
- NGSCB produces a parallel operation environment hosted by a new kernel called the "Nexus" that existed alongside Windows and provides new applications with features such as hardware-based process isolation, data encryption based on integrity measurements, authentication of a local or remote machine or software configuration, and encrypted paths for user authentication and graphics output.
- Project was never finished to completion.



https://en.wikipedia.org/wiki/Next-Generation_Secure_Computing_Base

IBM SecureBlue++

- Provides fine-grained crypto protection to protect information in one program from other software (including privileged software like OS, device drivers or malware that obtains root privileges)
- Protects confidentially & integrity of information so other software cannot read it or undetectably tamper with it.
- Looks like XOM, see later slides



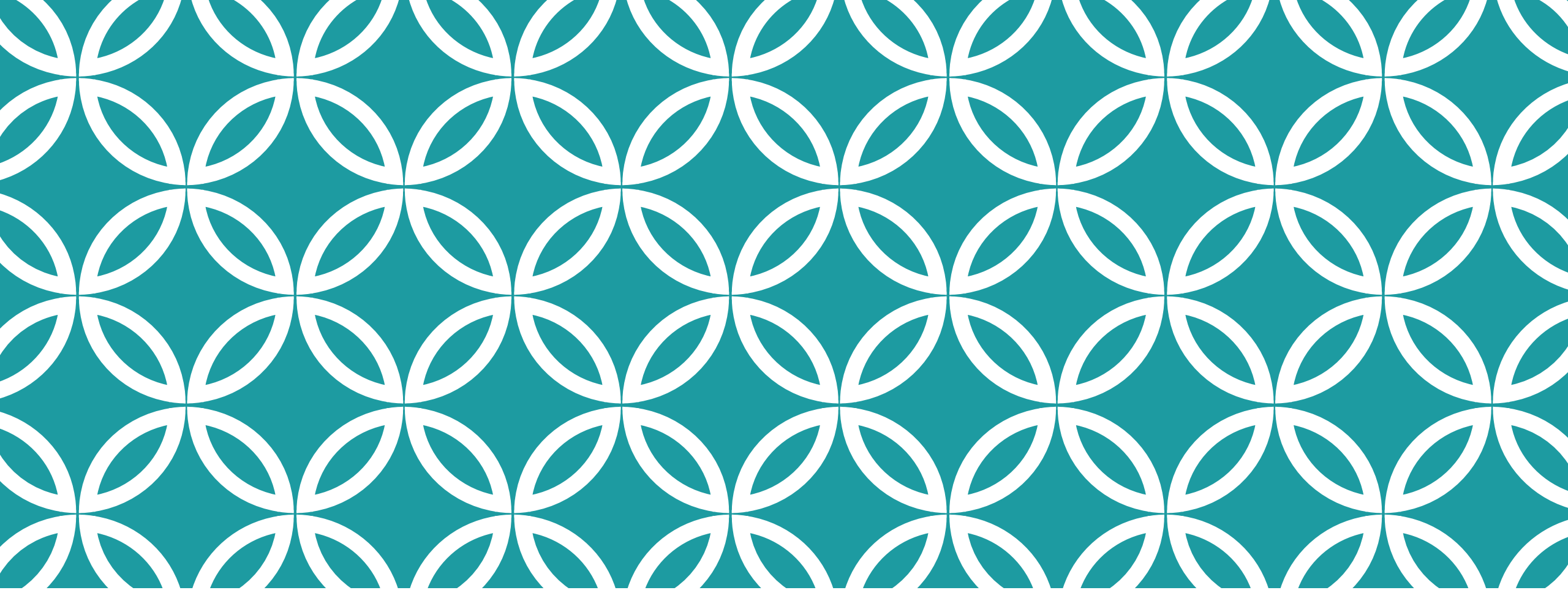
http://domino.research.ibm.com/library/cyberdig.nsf/papers/E605BDC5439097F085257A13004D25CA/$File/rc25287.pdf

ARM TrustZone

- It conceptually partitions a system's resources between a secure world, which hosts a secure container, and a normal world, which runs an untrusted software stack.
- The secure container must also implement a monitor that performs the context switches needed to transition an execution core between the two worlds.
- Code in secure world can compromise any level in the normal world's software stack, but the code in normal world can only access the secure world via an instruction that jumps into the monitor.
- Drawbacks:
  - Cache lines are shared among two worlds, which makes cache side channel possible.
  - Unfortunately, hardware manufacturers that license the TrustZone IP cores are reluctant to disclose all the details of their designs
  - No countermeasures for physical attacks. ARM recommends to store all the code for secure world in on-chip SRAM.
  - Documentation does not describe any software attestation implementation (one of the next lectures).

Software Guard Extension (SGX)

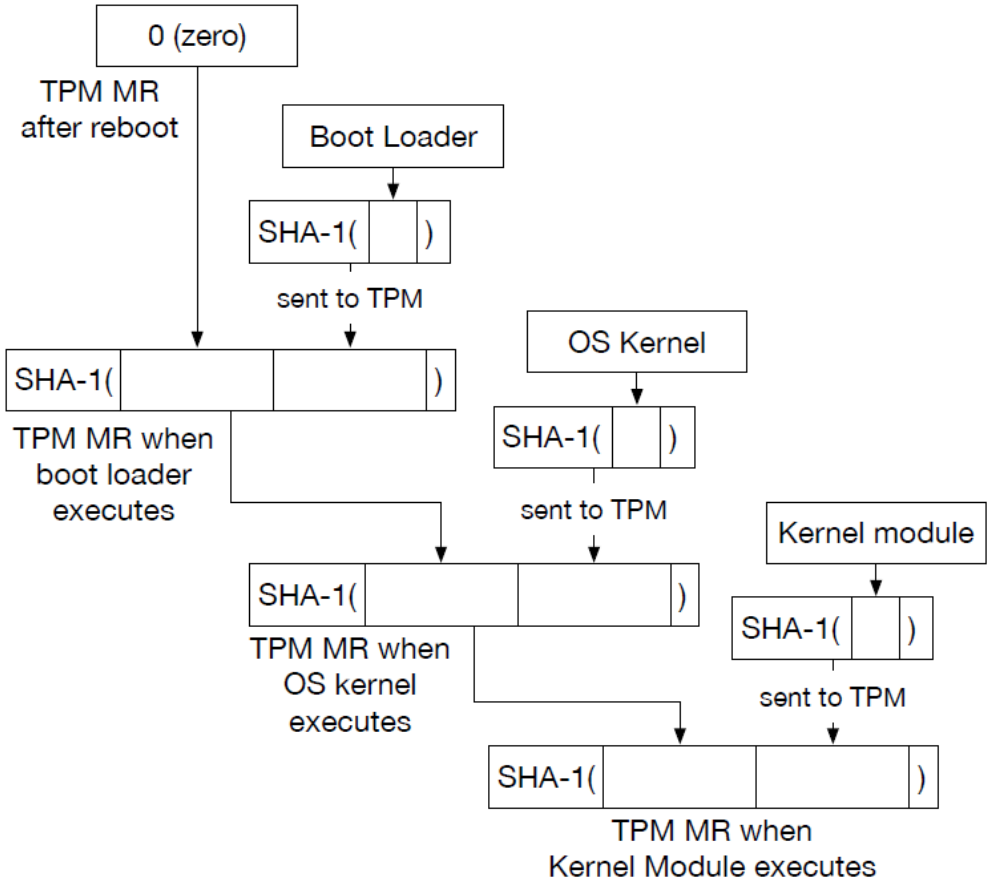
- SGX implements secure containers for applications without making any modifications to the processor's critical execution path.
- SGX does not trust any layer in the computer's software stack (firmware, hypervisor, OS).
- Intel's documentation states that SGX guarantees DRAM confidentiality, authentication, and freshness by virtue of a Memory Encryption Engine (MEE).
- SGX does not protect against cache timing attacks.
- Next couple of lectures



Boot Security

Trusted Platform Module (TPM)

- The Trusted Platform Module (TPM) introduced the software attestation model.
- The TPM design does not require any hardware modifications to the CPU, and instead relies on an auxiliary tamper-resistant chip.
- The TPM chip is only used to store the attestation key and to perform software attestation.
- The TPM's design relies on the software running on the CPU to report its own cryptographic hash. The TPM chip resets the measurements stored in Platform Configuration Registers (PCRs) when the computer is rebooted.
- The TPM expects the software at each boot stage to cryptographically hash the software at the next stage, and send the hash to the TPM.



Breaking a TPM

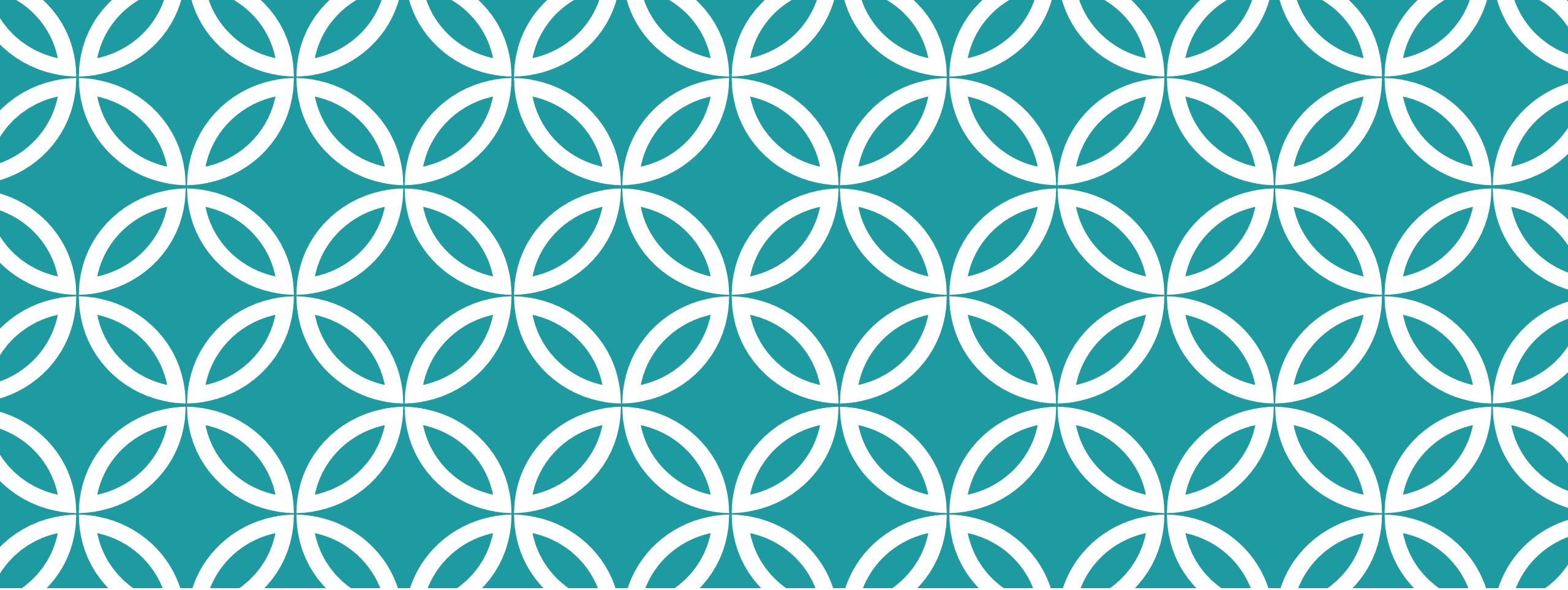
- A TPM-based system is vulnerable to an attacker who has physical access to the machine, as the TPM chip does not provide any isolation for the software on the computer.
- The TPM chip receives the software measurements from the CPU, so TPM-based systems are vulnerable to attackers who can tap the communication bus between the CPU and the TPM.
- It is difficult to maintain a long chain of trust
- The need to anchor the chain at some trusted piece of code, somewhere at the very beginning of the platform life cycle. This piece is usually referred to as the Core Root of Trust for Measurement (CRTM). Therefore CRTM must be stored in a ROM.
- Up until recently the CRTM was implemented by the BIOS using the normal SPI flash memory. The same flash memory the attacker can usually modify after successfully attacked the BIOS.

Trusted Execution Technology (TXT)

- Main goal is to eliminate BIOS from TCB.
- Uses the TPM's software attestation model and auxiliary tamper-resistant chip, but reduces the software inside the secure container to a virtual machine (guest operating system and application) hosted by the CPU's hardware virtualization features.
- Isolates the software inside the container from untrusted software by ensuring that the container has exclusive control over the entire computer while it is active.
- TPM measures boot process to make up the platform's Static Root of Trust Measurement (SRTM). When a TXT VM is initialized, it updates TPM registers that make up the Dynamic Root of Trust Measurement (DRTM).
- While the TPM's SRTM registers only reset at the start of a boot cycle, the DRTM registers are reset by the SINIT ACM, every time a TXT VM is launched.

Broken promise of TXT

- System Management Mode code (SMM), more precisely MSI handler, is able to compromise TXT-loaded hypervisor or OS.
- SMM is provided by the BIOS, and so if the BIOS gets compromised it can load arbitrary SMM
- Solution from Intel: a special additional hypervisor, dedicated to sandboxing of SMM, called SMM Transfer Monitor, or STM.
- Why STM may not be a good solution:
  - STM is provided by BIOS vendor. Although a backdoored STM provided by BIOS vendor can be detected by the hypervisor loaded by TXT, there is no known good STM out there to be compared with.



Academic Secure Processors

XOM Architecure

- The execute-only memory (XOM) architecture introduced the approach of executing sensitive code and data in isolated containers managed by untrusted host software.
- XOM outlined the mechanisms needed to isolate a container's data from its untrusted software environment.
- XOM supports multiple containers by tagging every cache line with the identifier of the container owning it, and ensures isolation by disallowing memory accesses to cache lines that don't match the current container's identifier.
- XOM also introduced the integration of encryption and HMAC functionality in the processor's memory controller to protect container memory from physical attacks on DRAM.
- XOM does not guarantees DRAM freshness.

Aegis

- The Aegis secure processor relies on a security kernel in the operating system to isolate containers, and includes the kernel's cryptographic hash in the measurement reported by the software attestation signature.
- The Aegis memory controller encrypts the cache lines in one memory range, and HMACs the cache lines in one other memory range.
- Aegis was the first secure processor not vulnerable to physical replay attacks, as it uses a Merkle tree construction to guarantee DRAM freshness.
- A future lecture

Bastion

- The Bastion architecture introduced the use of a trusted hypervisor to provide secure containers to applications running inside unmodified, untrusted operating systems. Bastion's hypervisor ensures that the operating system does not interfere with the secure containers.
- Each Bastion container has a Security Segment that lists the virtual addresses and permissions of all the container's pages, and the hypervisor maintains a Module State Table that stores an inverted page map, associating each physical memory page to its container and virtual address.
- The processor's hardware page walker is modified to invoke the hypervisor on every TLB miss, before updating the TLB with the address translation result. The hypervisor checks that the address translation is done correctly.

Ascend

- The Ascend secure processors introduced practical implementations of Oblivious RAM techniques in the CPU's memory controller.
- These processors are resilient to attackers who can probe the DRAM address bus and attempt to learn a container's private information from its DRAM memory access pattern.
- A future lecture

CHERI

- CHERI extends a conventional RISC Instruction Set Architecture, compiler, and operating system to support fine-grained, capability-based memory protection to mitigate memory-related vulnerabilities in C-language TCBs.

Sanctum

- Sanctum introduced a straightforward software/hardware co-design that yields the same resilience against software attacks as SGX, and adds protection against memory access pattern leaks, such as page fault monitoring attacks and cache timing attacks.
- Sanctum partitions a computer's DRAM into equally-sized continuous DRAM regions, and each DRAM region uses distinct sets in the shared last-level cache (LLC). Each DRAM region is allocated to exactly one container, so containers are isolated in both DRAM and the LLC. Containers are isolated in the other caches by flushing on context switches.
- Sanctum relies on a trusted security monitor, which is the first piece of firmware executed by the processor. The monitor is measured by bootstrap code in the processor's ROM, and its cryptographic hash is included in the software attestation measurement.
- A future lecture

Remote Adversary

Physically Present Adversary

Attack	TrustZone	TPM	TPM+TXT	SGX	XOM	Aegis	Bastion	Ascend, Phantom	Sanctum
Malicious containers (direct probing)	N/A (secure world is trusted)	N/A (The whole computer is one container)	N/A (Does not allow concurrent containers)	Access checks on TLB misses	Identifier tag checks	Security kernel separates containers	Access checks on each memory access	OS separates containers	Access checks on TLB misses
Malicious OS (direct probing)	Access checks on TLB misses	N/A (OS measured and trusted)	Host OS preempted during late launch	Access checks on TLB misses	OS has its own identifier	Security kernel measured and isolated	Memory encryption and HMAC	X	Access checks on TLB misses
Malicious hypervisor (direct probing)	Access checks on TLB misses	N/A (Hypervisor measured and trusted)	Hypervisor preempted during late launch	Access checks on TLB misses	N/A (No hypervisor support)	N/A (No hypervisor support)	Hypervisor measured and trusted	N/A (No hypervisor support)	Access checks on TLB misses
Malicious firmware	N/A (firmware is a part of the secure world)	CPU microcode measures PEI firmware	SINIT ACM signed by Intel key and measured	SMM handler is subject to TLB access checks	N/A (Firmware is not active after booting)	N/A (Firmware is not active after booting)	Hypervisor measured after boot	N/A (Firmware is not active after booting)	Firmware is measured and trusted
Malicious containers (cache timing)	N/A (secure world is trusted)	N/A (Does not allow concurrent containers)	N/A (Does not allow concurrent containers)	X	X	X	X	X	Each enclave its gets own cache partition
Malicious OS (page fault recording)	Secure world has own page tables	N/A (OS measured and trusted)	Host OS preempted during late launch	X	N/A (Paging not supported)	X	X	X	Per-enclave page tables
Malicious OS (cache timing)	X	N/A (OS measured and trusted)	Host OS preempted during late launch	X	X	X	X	X	Non-enclave software uses a separate cache partition
DMA from malicious peripheral	On-chip bus bounces secure world accesses	X	IOMMU bounces DMA into TXT memory range	IOMMU bounces DMA into PRM	Equivalent to physical DRAM access	Equivalent to physical DRAM access	Equivalent to physical DRAM access	Equivalent to physical DRAM access	MC bounces DMA outside allowed range
Physical DRAM read	Secure world limited to on-chip SRAM	X	X	Undocumented memory encryption engine	DRAM encryption	DRAM encryption	DRAM encryption	DRAM encryption	X
Physical DRAM write	Secure world limited to on-chip SRAM	X	X	Undocumented memory encryption engine	HMAC of address and data	HMAC of address, data, timestamp	Merkle tree over DRAM	HMAC of address, data, timestamp	X
Physical DRAM rollback write	Secure world limited to on-chip SRAM	X	X	Undocumented memory encryption engine	X	Merkle tree over HMAC timestamps	Merkle tree over DRAM	Merkle tree over HMAC timestamps	X
Physical DRAM address reads	Secure world in on-chip SRAM	X	X	X	X	X	X	ORAM	X
Hardware TCB size	CPU chip package	Motherboard (CPU, TPM, DRAM, buses)	Motherboard (CPU, TPM, DRAM, buses)	CPU chip package	CPU chip package	CPU chip package	CPU chip package	CPU chip package	CPU chip package
Software TCB size	Secure world (firmware, OS, application)	All software on the computer	SINIT ACM + VM (OS, application)	Application module + privileged containers	Application module + hypervisor	Application module + security kernel	Application module + hypervisor	Application process + trusted OS	Application module + security monitor