SGX Security Background

- Masab Ahmad
masab.ahmad@uconn.edu
- Department of Electrical and Computer Engineering
University of Connecticut

Security Background Outline

- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Security Background Outline

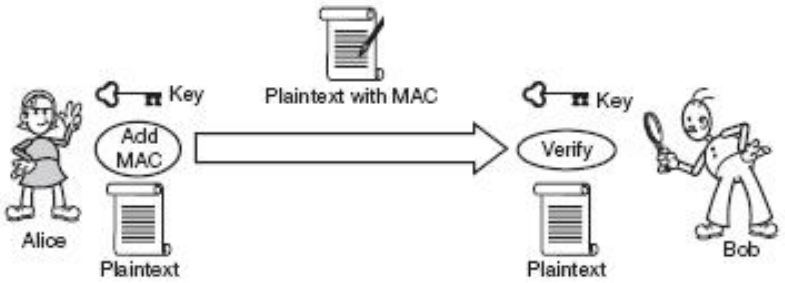
- **Cryptographic Primitives**
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Cryptographic Primitives

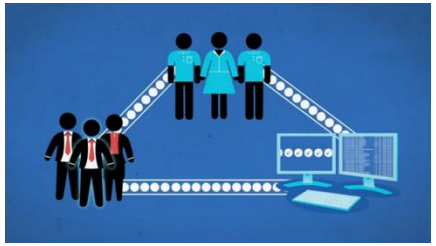
- Cryptographic Keys

- Privacy – No one should see/infer my data – *Encryption*

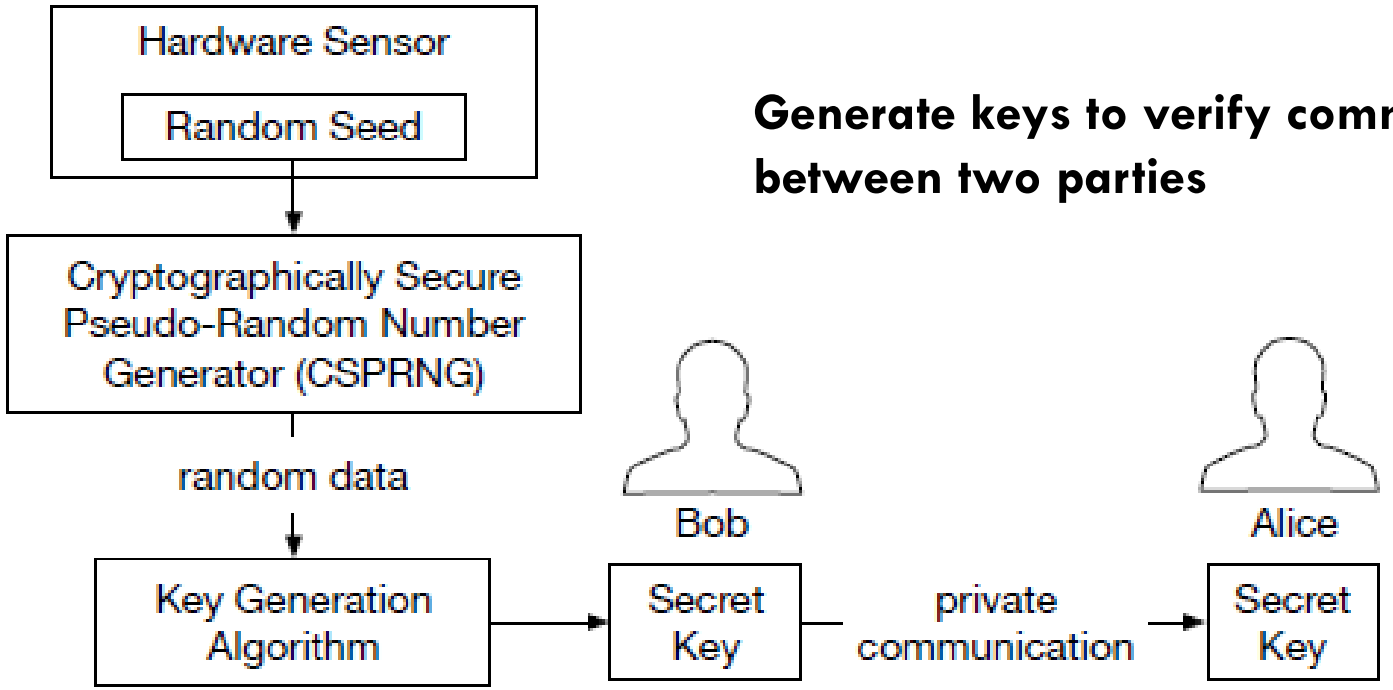
- Integrity – No one should tamper with my data – *MAC/Signatures*



- Freshness – I should get the same data as I viewed earlier – *Integrity Checking*



Cryptographic Keys



Generate keys to verify communication between two parties

Maintain secret

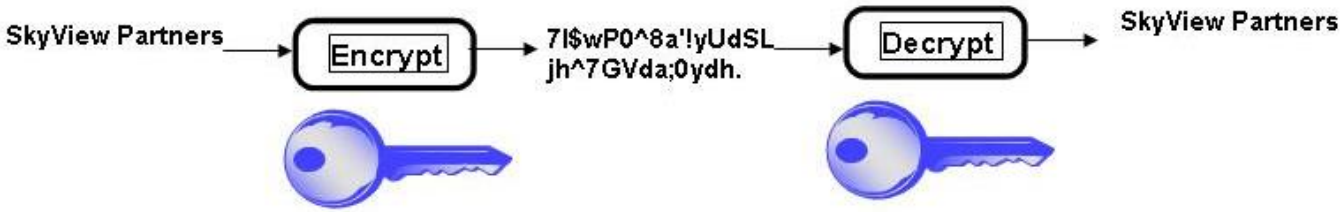
"Intel **SGX Explained"**. Victor Costan and Srinivas Devadas, MIT 2016

Cryptographic Keys

DES
TripleDES
AES
RC5

Symmetric Keys

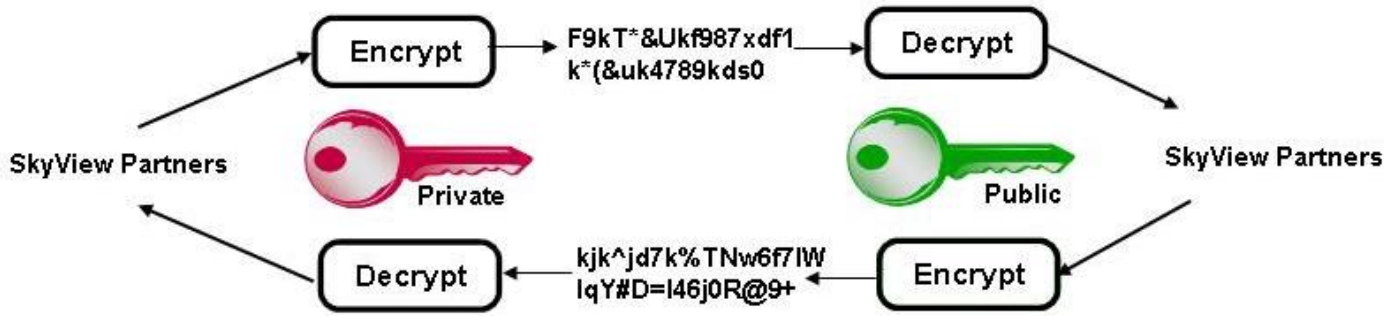
- Encryption and decryption use the **same key**.



RSA
Elliptic Curve

Asymmetric keys

- Encryption and decryption use different keys, a **public key** and a **private key**.



https://www.cybrary.it/0p3n/symmetric-encryption/

MD5
SHA-1

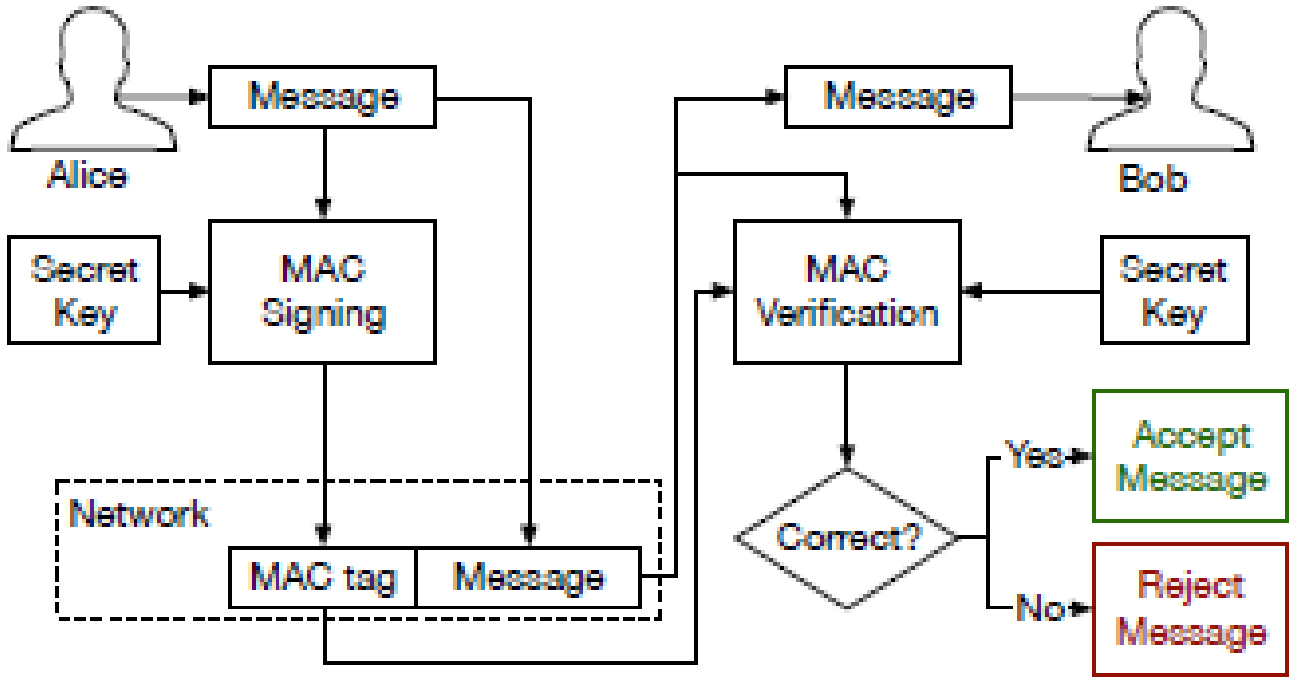
One-way hash

Privacy

Message authentication signatures

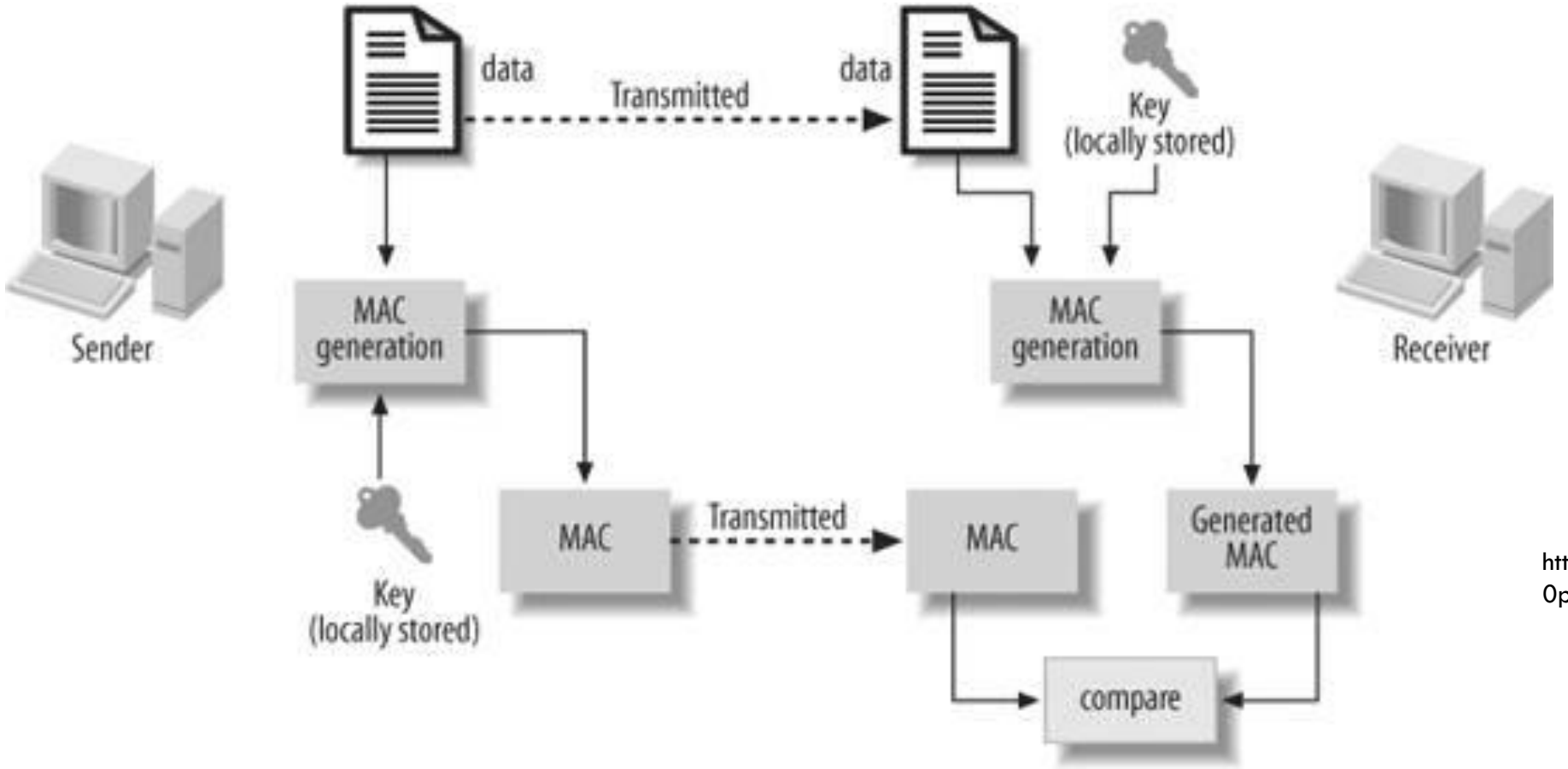
Used to verify that data came from the correct party

MAC : Message authentication code



Data encrypted to ensure attackers cannot view it

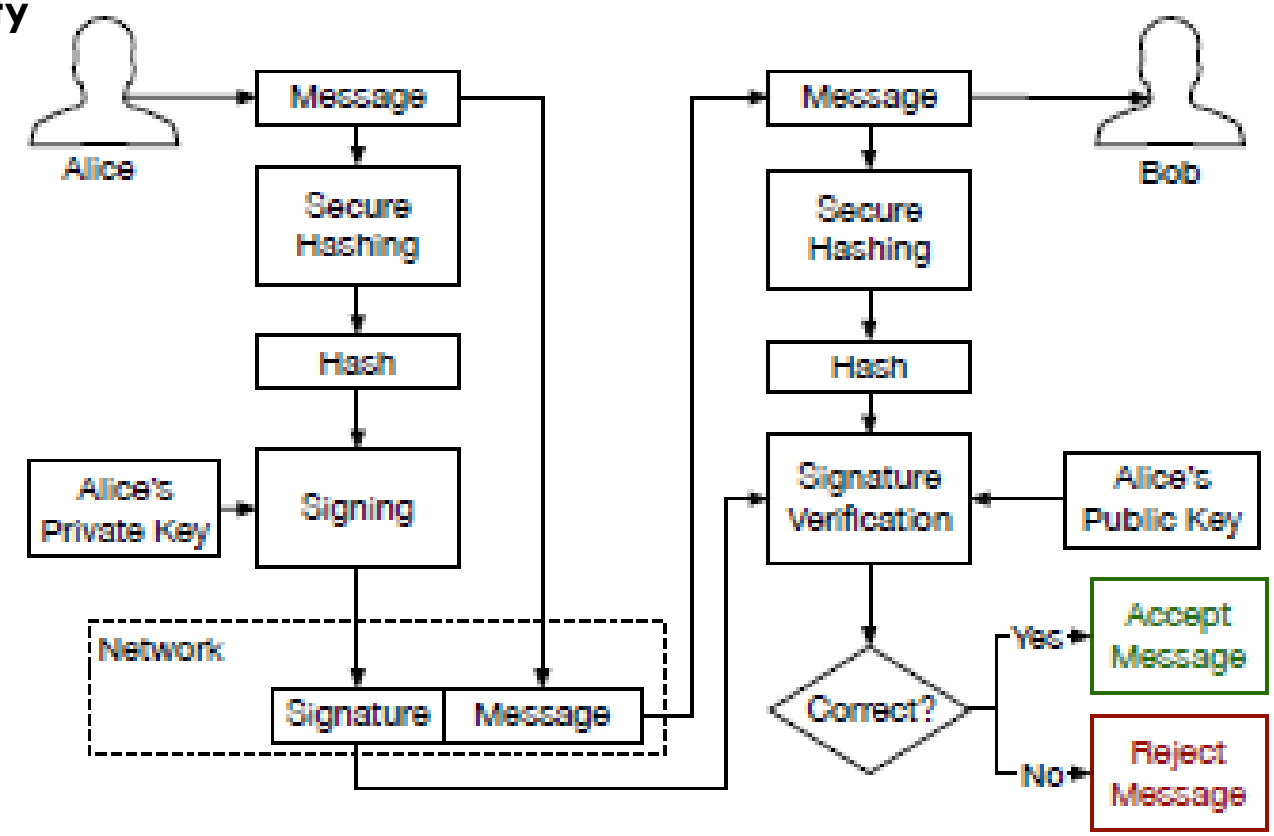
Privacy



https://www.cybrary.it/0p3n/macs/

Data Integrity

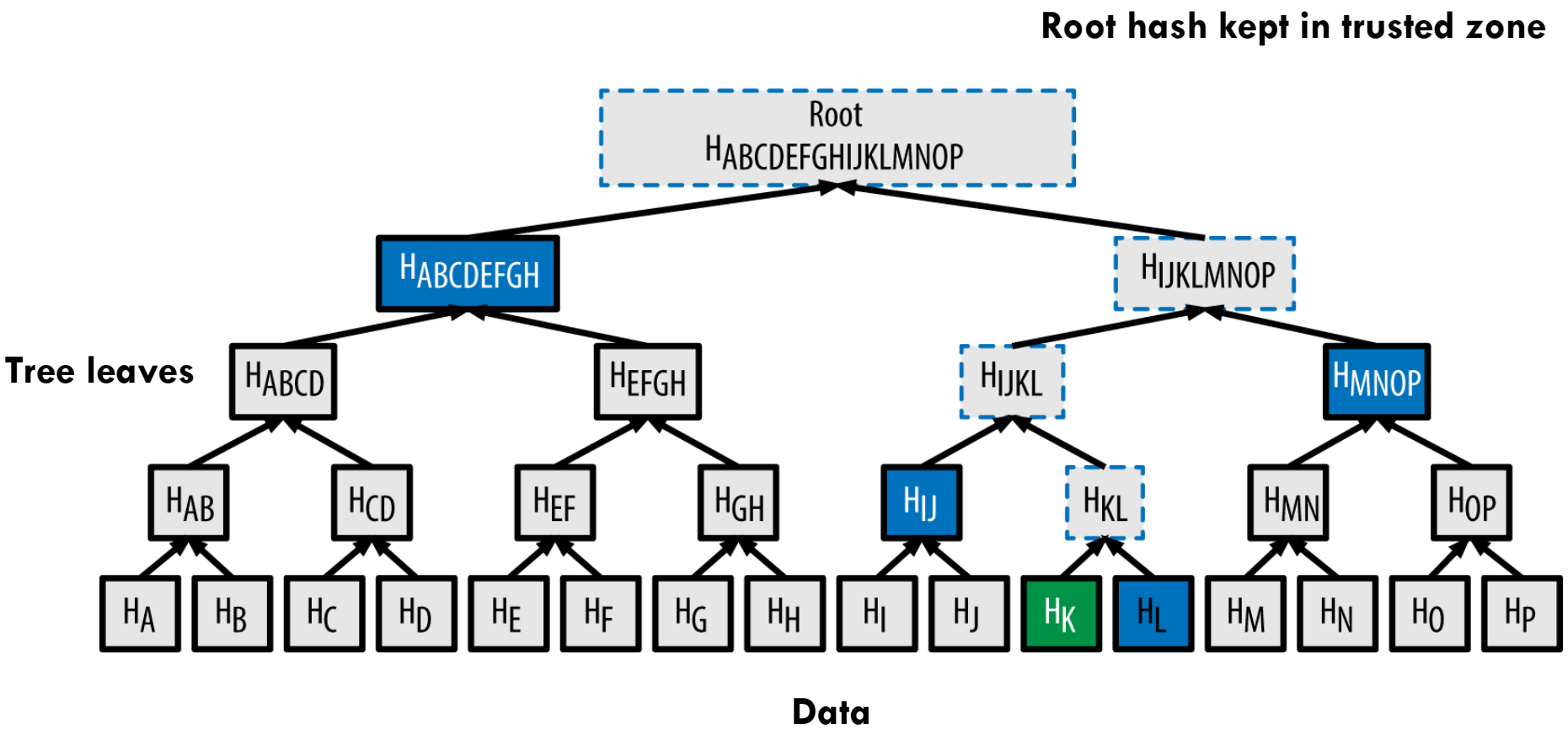
Use hashing mechanisms to verify data integrity



Keep a hash in a trusted zone

Compute hashes and then compare for security purposes

Hash Tree

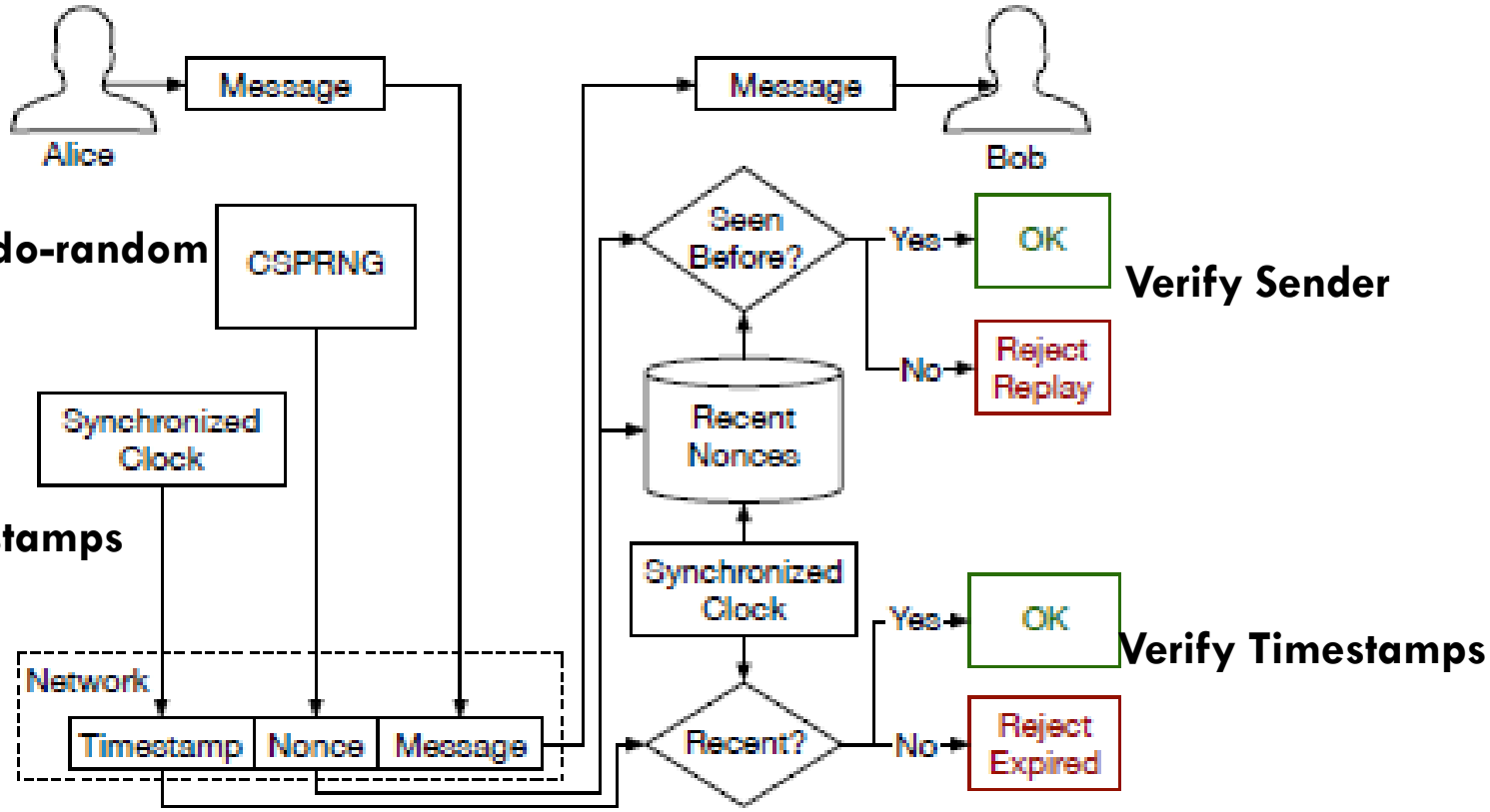


http://bitcoin.stackexchange.com/questions/10479/what-is-the-merkle-root

Data Freshness

Cryptographically secure pseudo-random number generator (CSPRNG)

Data access Timestamps



Verify Sender

Verify Timestamps

Nonce compared with timestamps to verify freshness

A Nonce is an arbitrary number that is used once

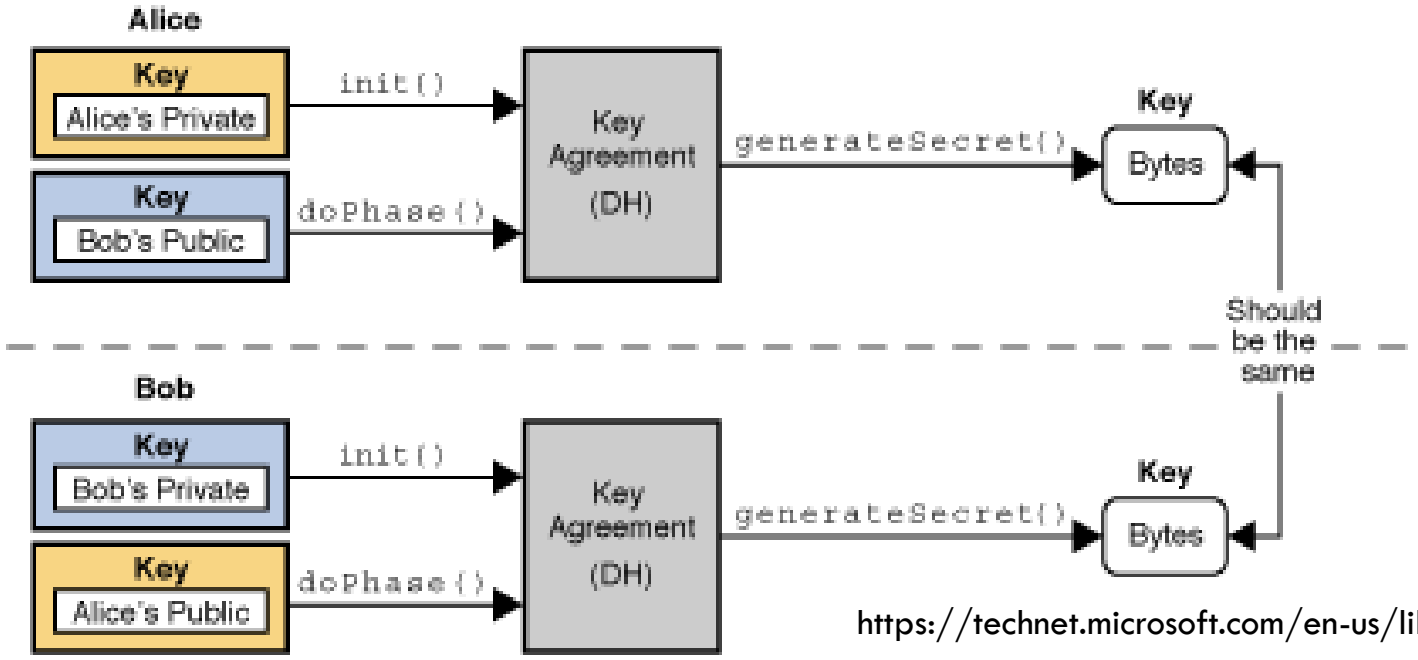
Security Background Outline

- Cryptographic Primitives
- **Cryptographic Constructs**
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Cryptographic Constructs

- Certificate Authorities

- Key Agreement Protocols



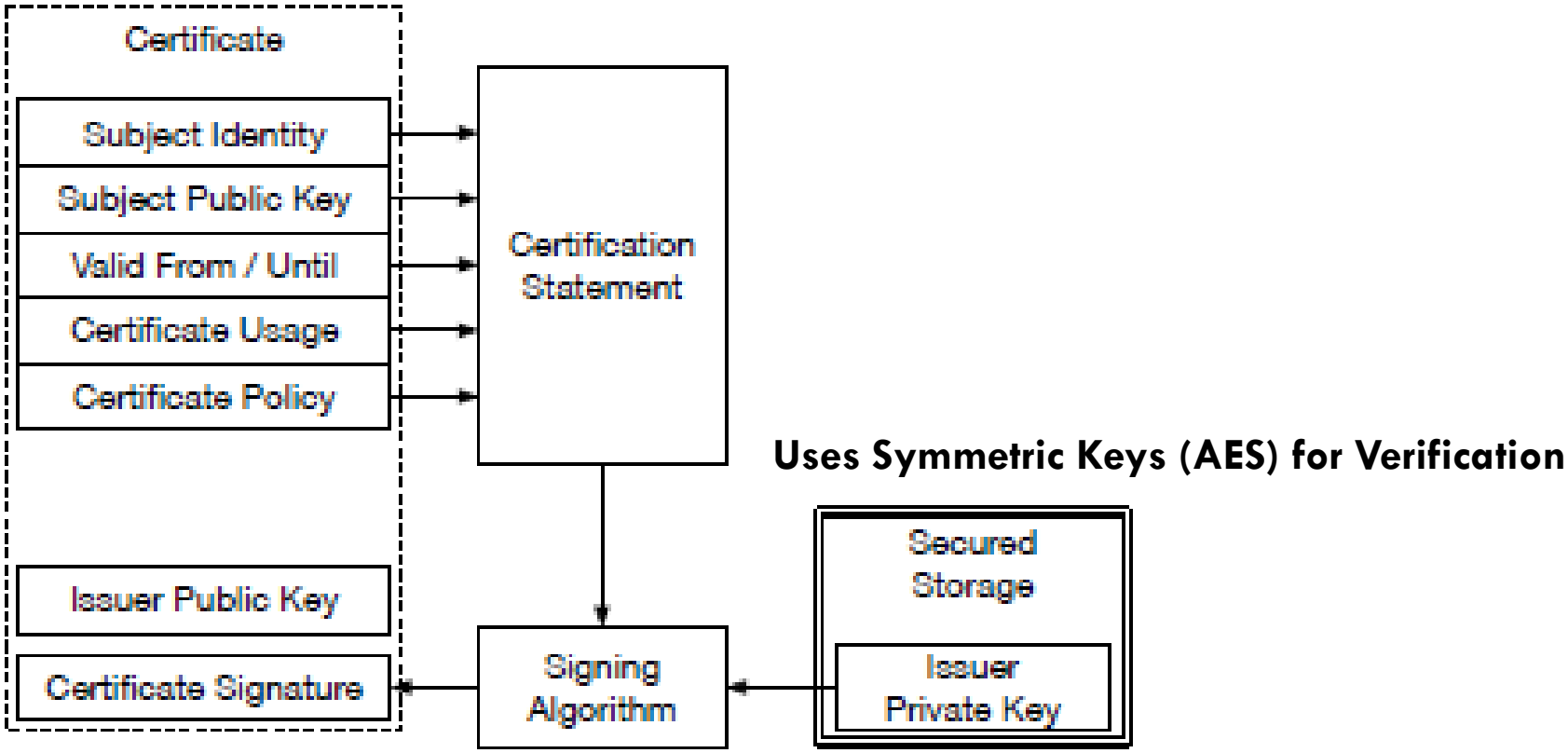
https://technet.microsoft.com/en-us/library/cc962035.aspx

Certificate Authorities (CA)

A digital certificate certifies the ownership of a public key by the named subject of the certificate

The CA is a trusted third party

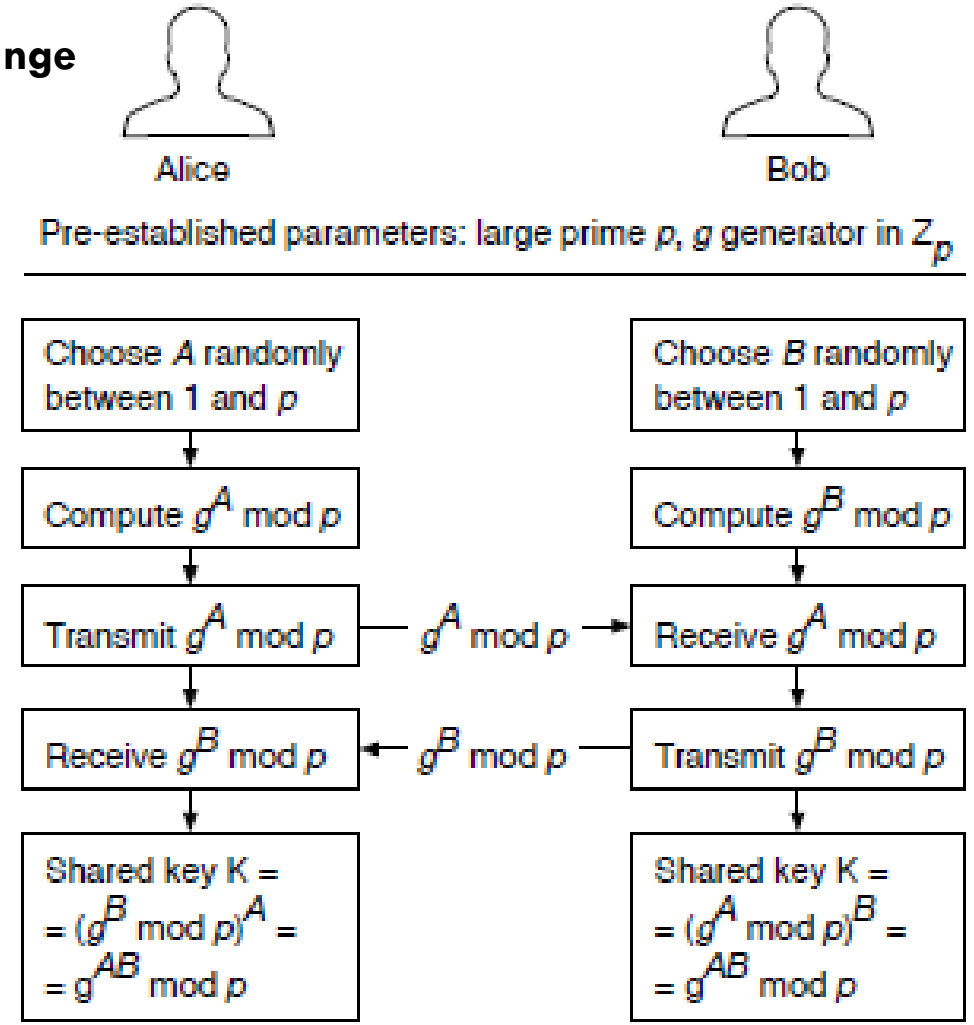
Certificate Parameters allow verification and security



Uses Symmetric Keys (AES) for Verification

Key Agreement Protocols

Diffie Hellman Key Exchange



Alice and Bob want to agree on keys to verify their communication

Users communicate and create keys dynamically

Computationally hard for an eavesdropper to infer keys

Security Background Outline

- Cryptographic Primitives
- Cryptographic Constructs
- **Software Attestation**
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Software Attestation Overview

- Authenticated Key Agreements
- Software Management

http://4.bp.blogspot.com/-mmUu09Gj0cM/UY0xPYG9LxI/AAAAAAAAABE/P_8kwUeGSrg/s1600/Qatar+Attestation.jpg

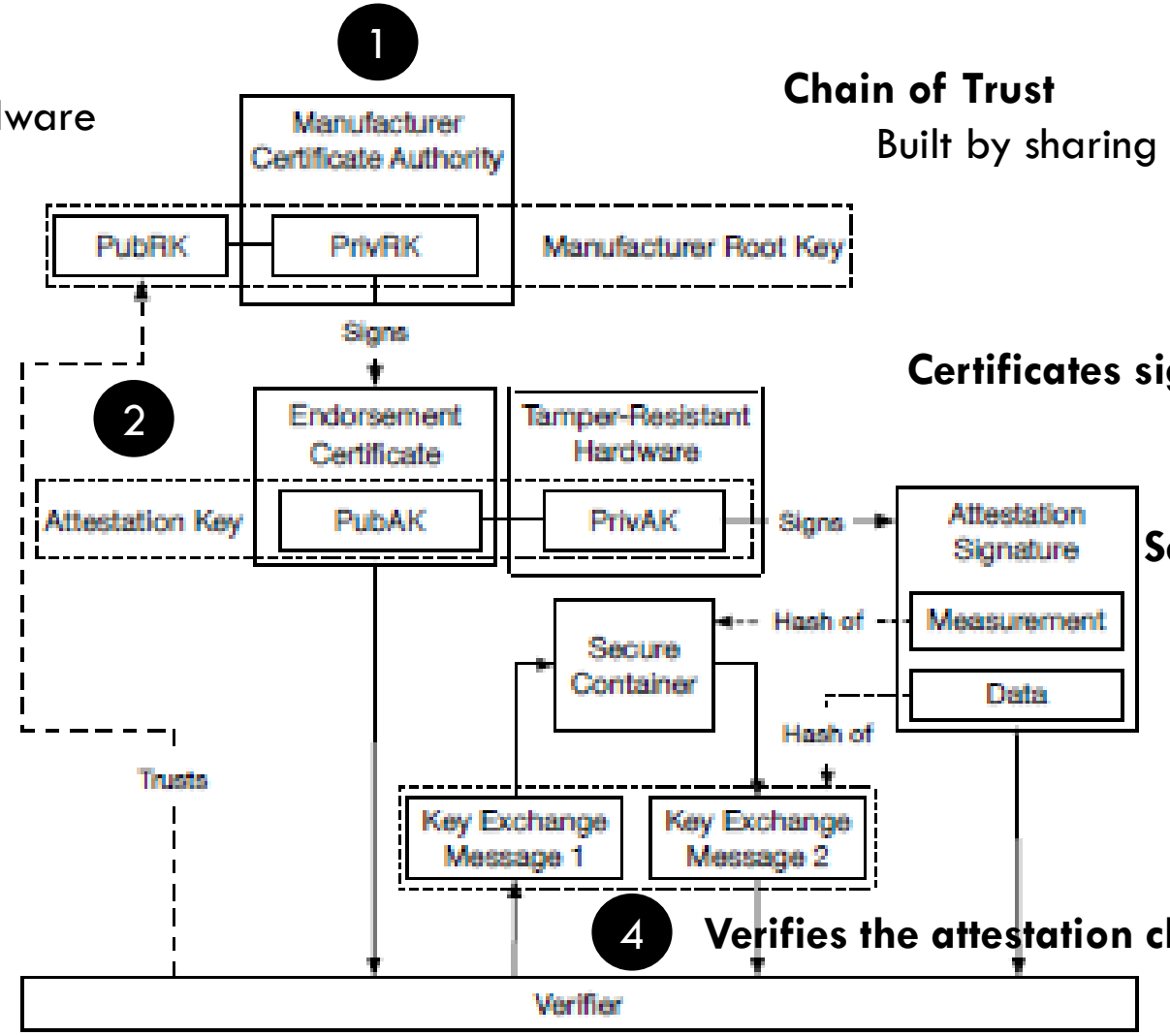
Authenticated Key Agreements and Software Management

Attestation Keys

Given by a hardware manufacturer

Chain of Trust

Built by sharing and verifying certificates



Certificates signed to verify their origin

Secure Container verifies Data Transfer

3

Software verified using these keys.

E.g. Apple is a verifier of software

Verifies the attestation chain internally

Security Background Outline

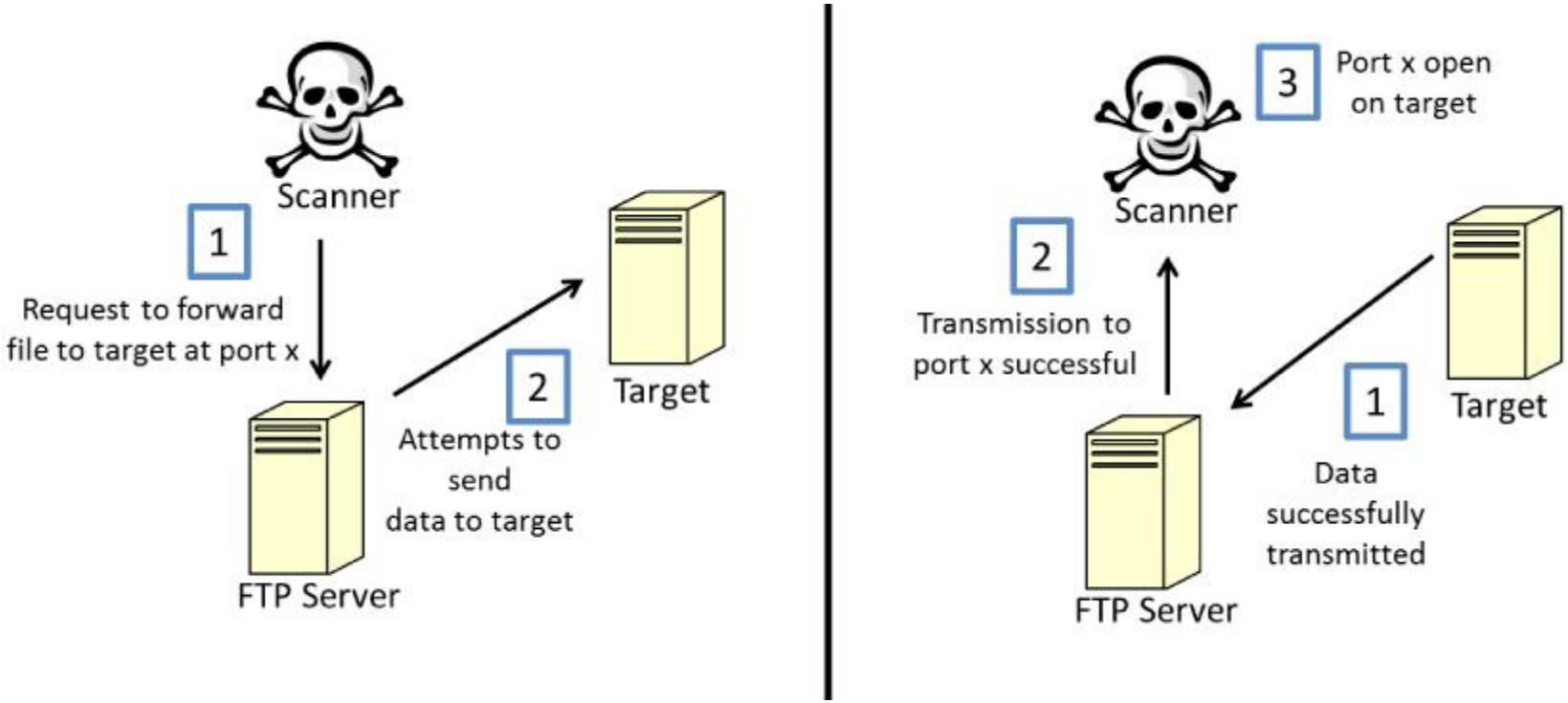
- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- **Physical Attacks**
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Physical Attacks

- Port Attacks
  - A reboot of the system via a port causes vulnerabilities of a system to surface
- Bus Tapping Attacks
  - Monitoring Attacks, Active Attacks, Replay Attacks
- Chip Attacks
  - Physically look/attack into the chip

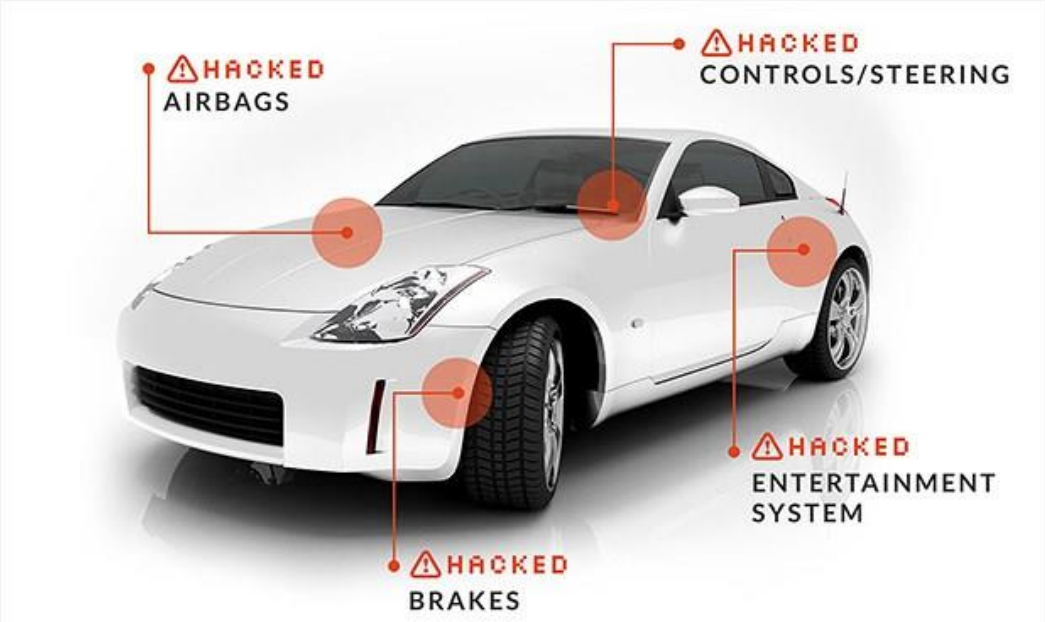
http://www.martybucella.com/

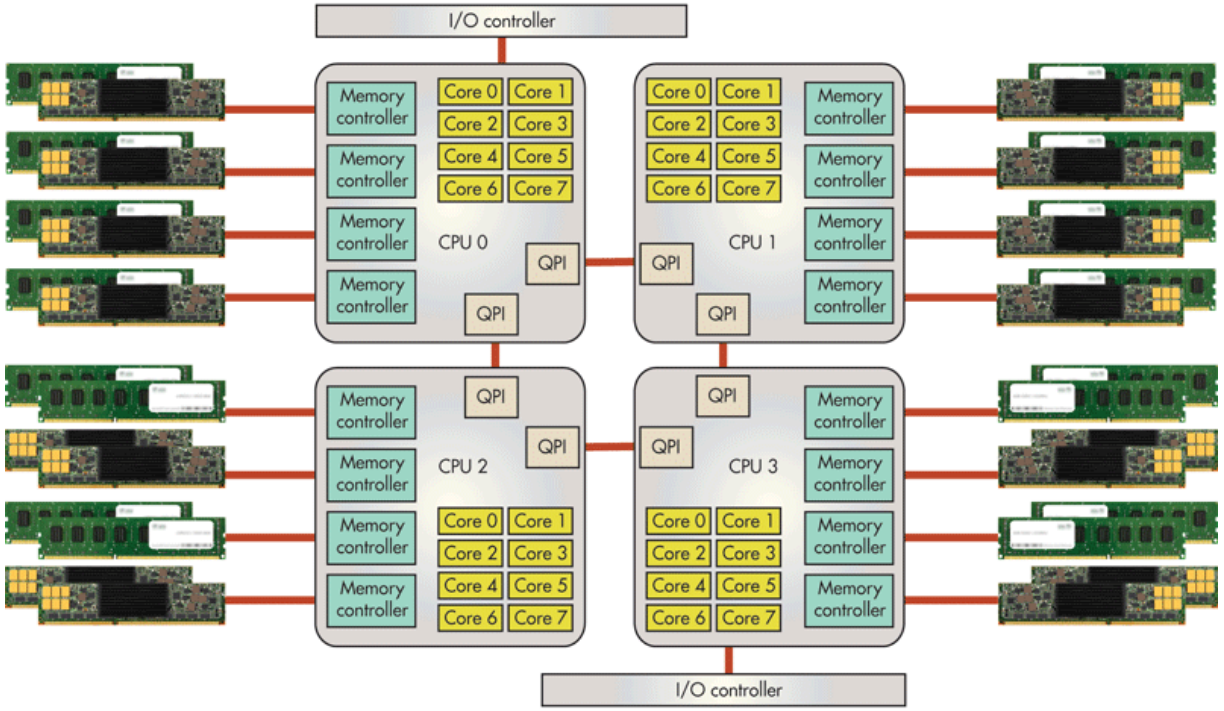
Port Attacks



- Attacker scans/sends communication ports after a system restart/bootup
  - Leaves BIOS insecure and leaves firewalls open
  - Private data can be readily available in such contexts
  - New processors filter data from the bus before committing anything to the state of the system

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.

- Attacks on the DRAM address sequences currently unsuccessful

Chip Attacks

- Reduce temperature of the chip
- Causes chips to go in “hibernation” mode with vulnerabilities
  - Security modules turned off
- Chip access compromised
  - 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.

Security Background Outline

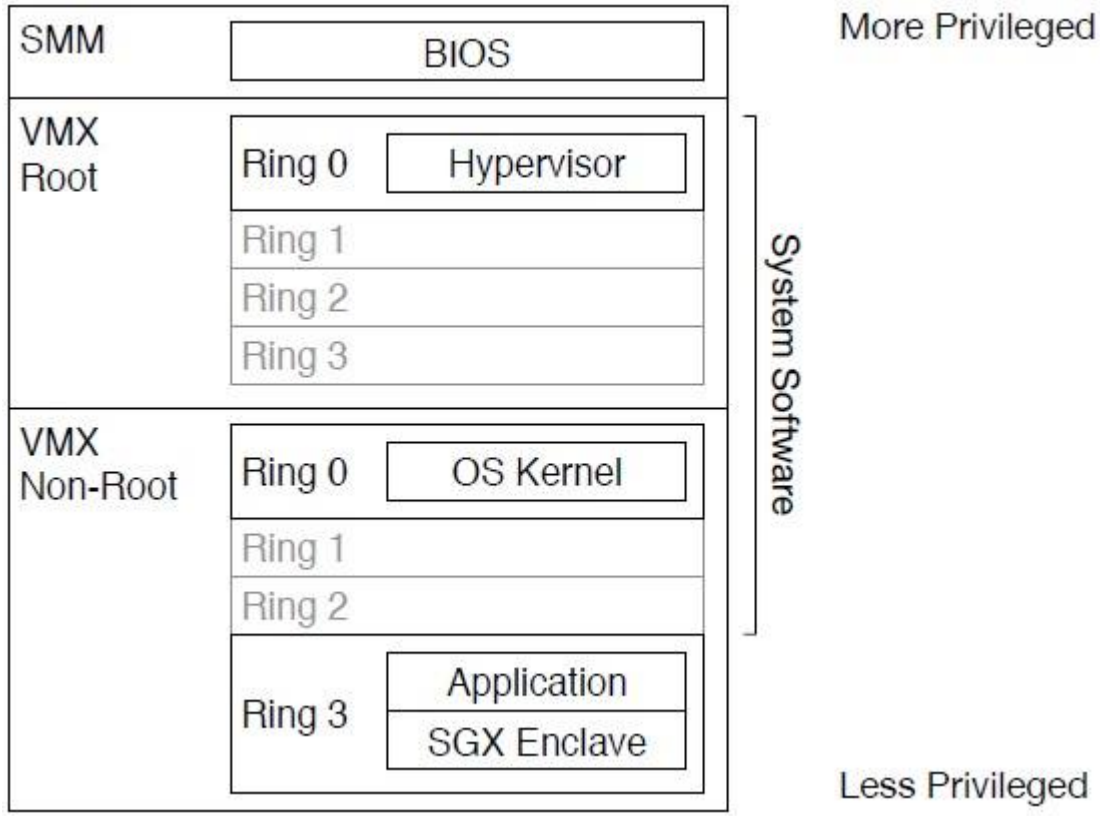
- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- **Privileged Software Attacks**
- Software Attacks on Peripherals
- Address Translation Attacks
- Cache Timing Attacks

Generations of Processor Security by Intel

- Intel Trusted Platform Module (TPM) 2009 -
  - Enabled basic security paradigms
  - Password Security, Disk Encryption, and Integrity Checking
- Intel Trusted Execution Technology (TXT) 2011 -
  - Platform attestation and its operating system
  - Chain of trust
- Intel Software Guard Extensions (SGX) 2016 -
  - Firmware Trusted
  - More to come in subsequent slides

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
- Heavily emphasized in Intel SGX
  - Hypervisor control needed – Isolation
  - OS also isolated from lower levels



Security Background Outline

- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- **Software Attacks on Peripherals**
- Address Translation Attacks
- Cache Timing Attacks

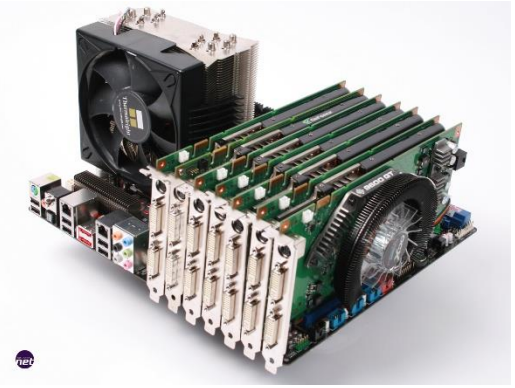
Software Attacks on Peripherals

- PCI Express Attacks
- DRAM Attacks
- The Performance Monitoring Side Channels
- Attacks on Boot Firmware and Intel ME
- Accounting for Software Attacks on 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
- Intel TXT additionally compromised this way
- New added checks in the DMA arbiter

Single Computing Node

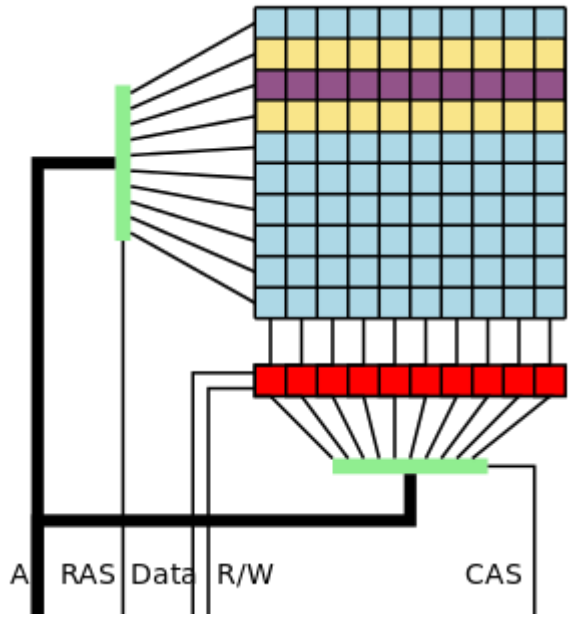


Multi-Node Supercomputer connected over PCI buses

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

DRAM Attacks

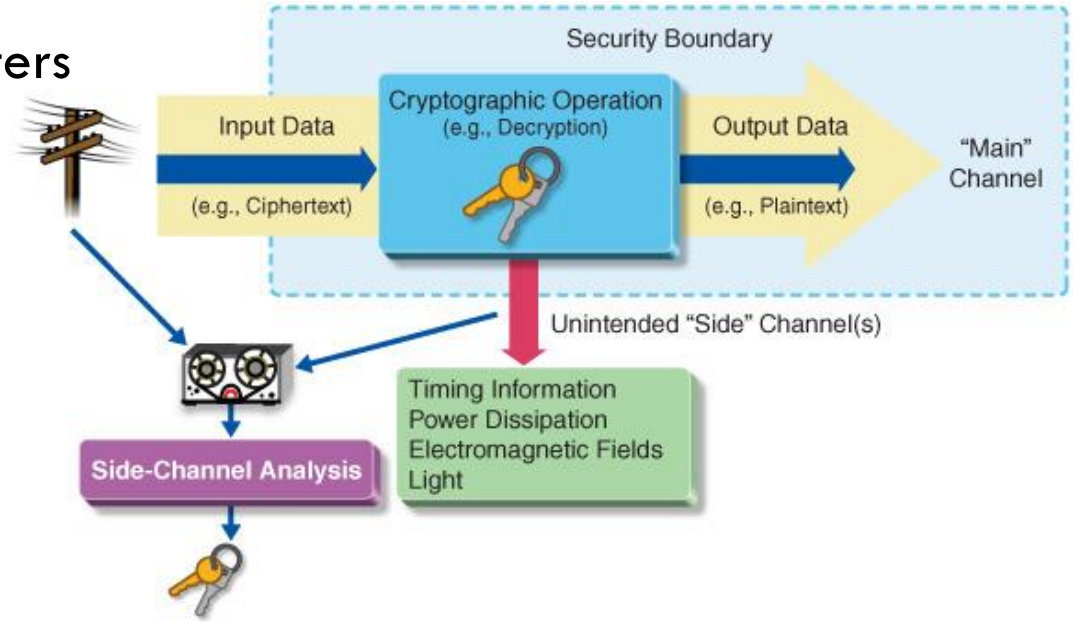
- Huge Class of attacks
  - **RowHammer** Attack
  - Bit flips on DRAM refresh
  - OR current leaks in memory can allow privileged access
  - Attacker then reads page table bits if leaked
  - Modification of page tables possible
- Isolation of page tables required in hardware
  - Hash checks



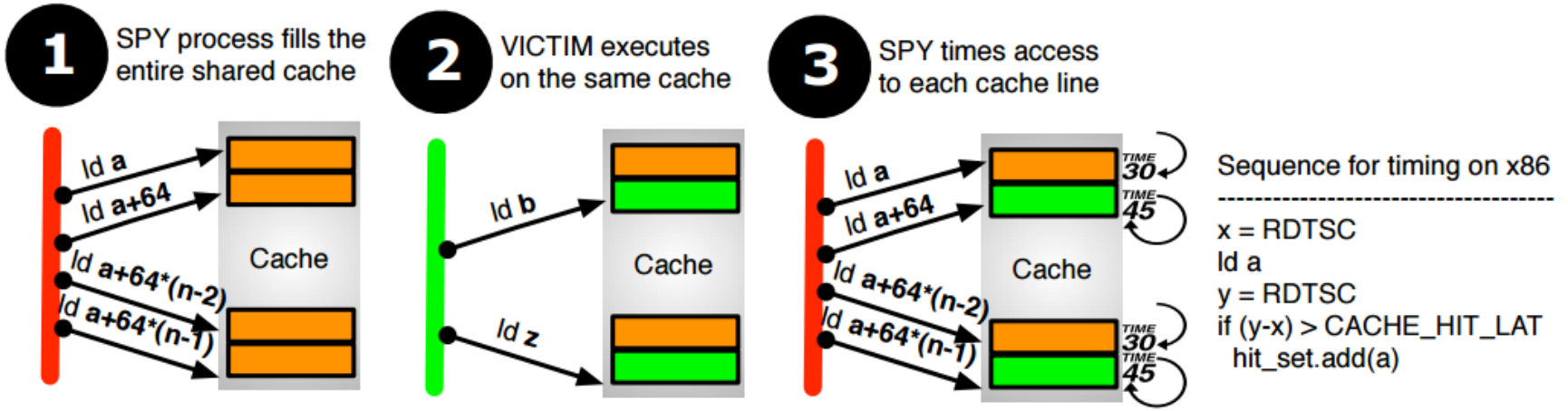
Bits leaked to adjacent cells

The Performance Monitoring Side Channels

- An attacker can gain access to performance counters
  - Read Model specific registers (MSRs)
  - Power analysis attacks
  - Other Side channel attacks



- Isolation of performance counters required
  - The victim thread must not be able to access the counters

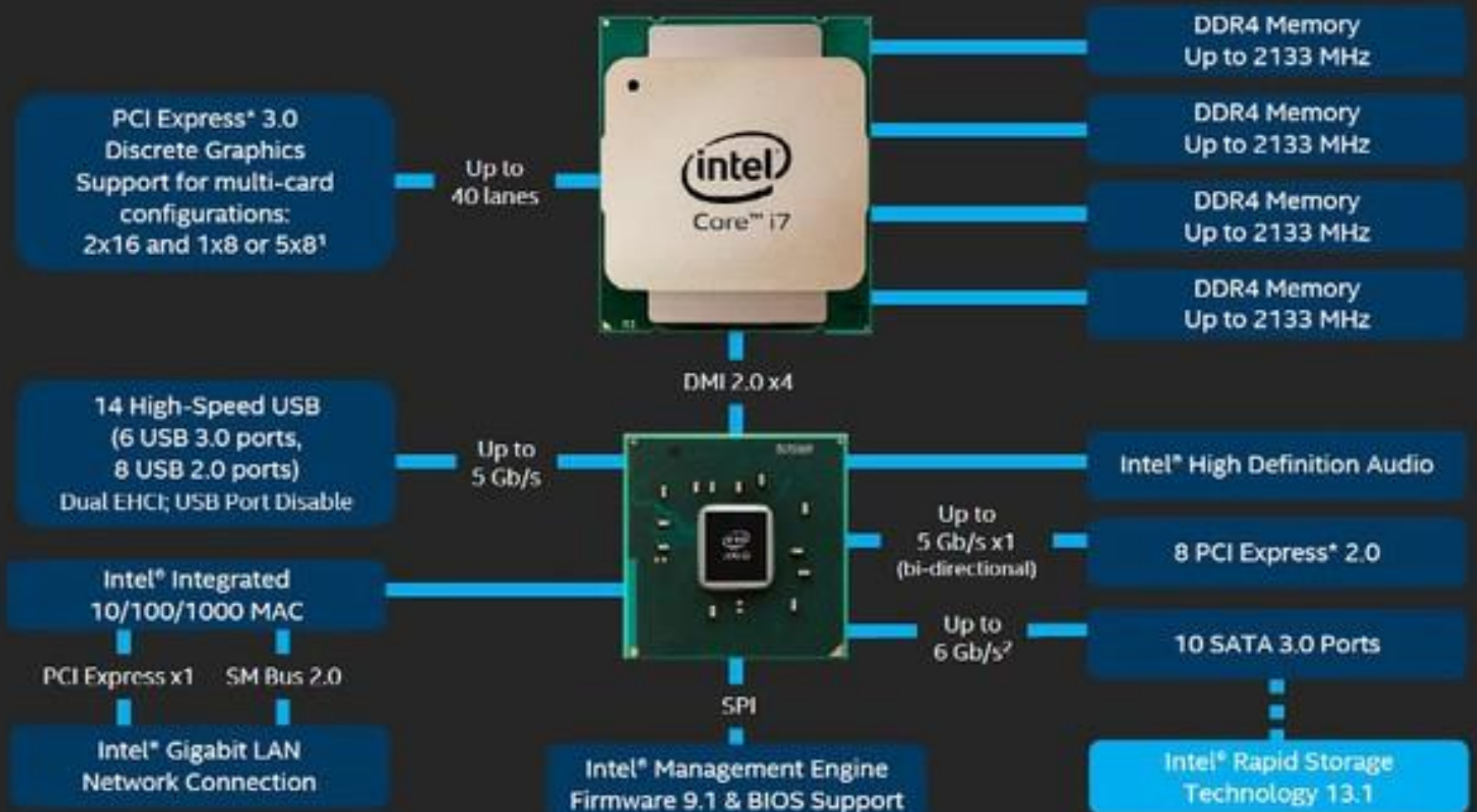


F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee. Last-Level Cache Side-Channel Attacks are Practical. In Proceedings of the 36th IEEE Symposium on Security and Privacy (S&P'15), 2015.

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
  - Security measures removed to ensure speedy machine start ups
- Isolation of firmware required

Intel® Core™ i7 High End Desktop Platform Overview



[1] 3 slots available, but need additional logic on board to support more slots. 5x8 configuration requires additional system clocks to be provided by 3rd party components.

[2] All SATA ports capable of 6 Gb/s.

intel Look Inside.

Accounting for Software Attacks on Peripherals

- 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

Security Background Outline

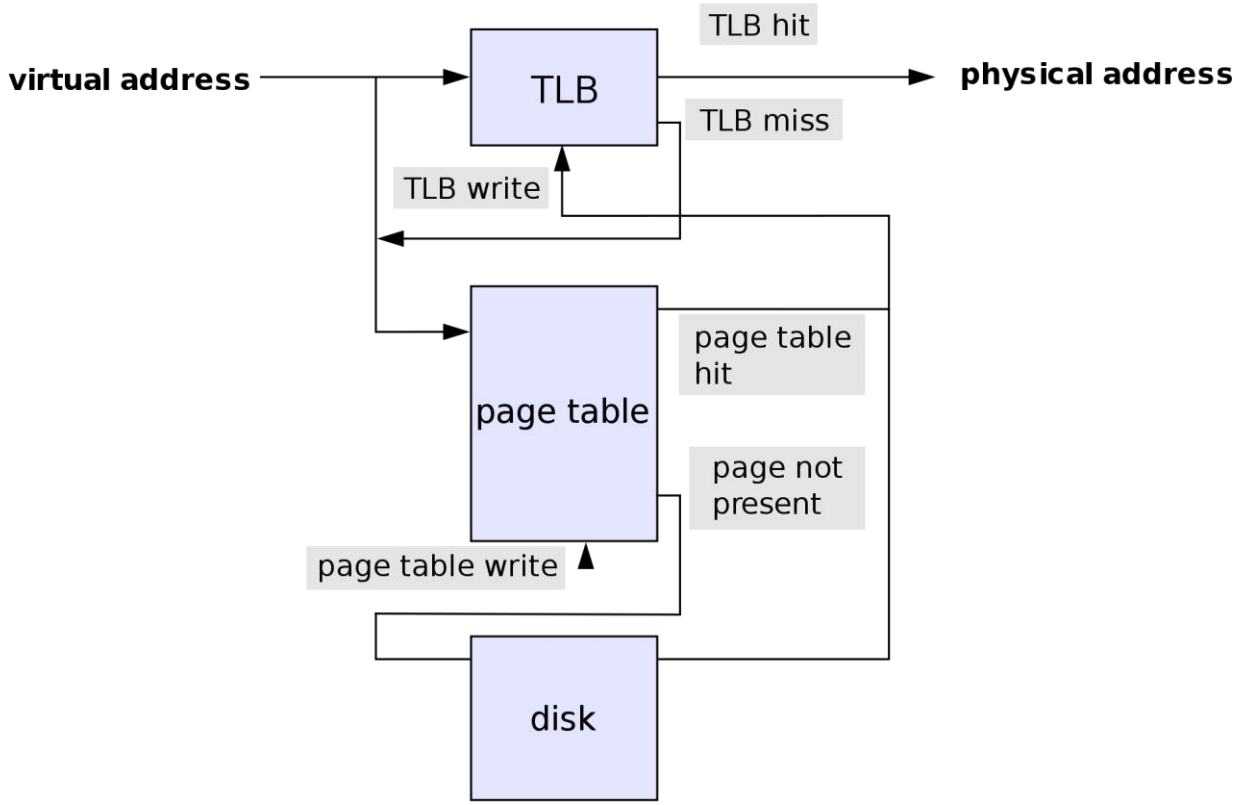
- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- **Address Translation Attacks**
- Cache Timing Attacks

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

Straightforward Active Attacks

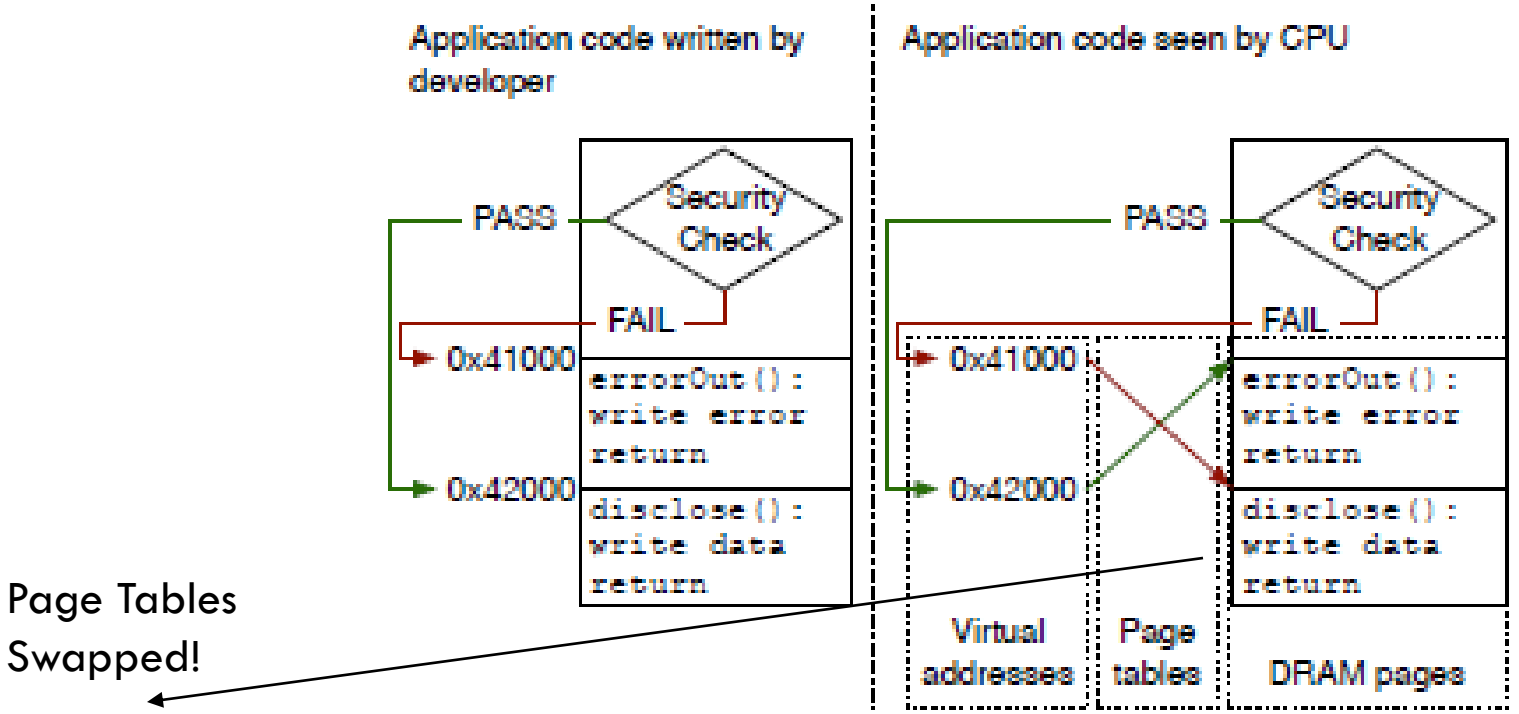
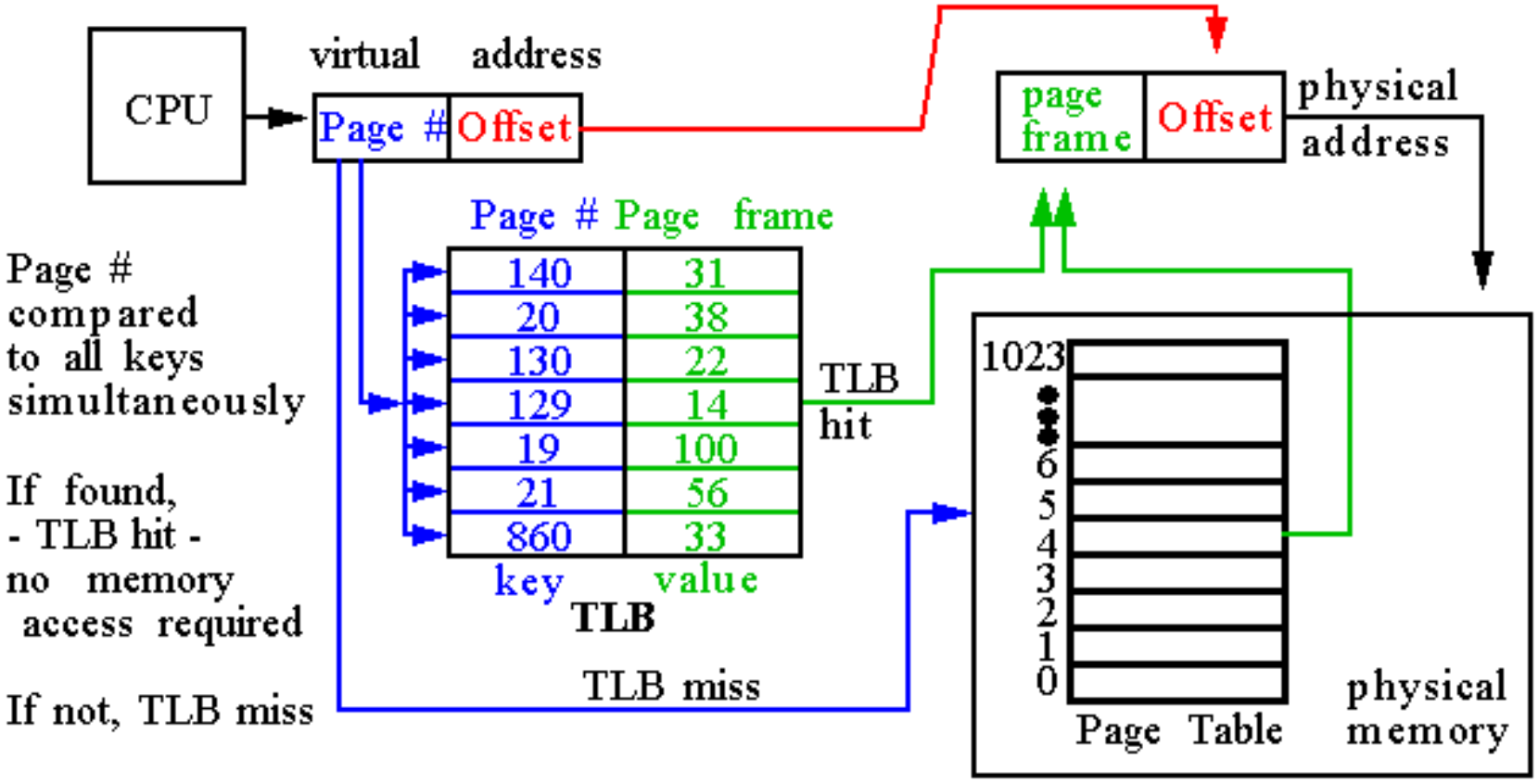


Figure 53: An example of an active memory mapping attack. The application's author intends to peform a security check, and only disclose a piece of sensitive information if the check passes. Malicious system software maps the virtual address of the procedure called when the security check fails to a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

- Use a similar attack methodology for a TLB (Translation Look-aside Buffer)
  - Maps virtual to physical address and vice versa
- Assumes that the TLB is not subject to security checks

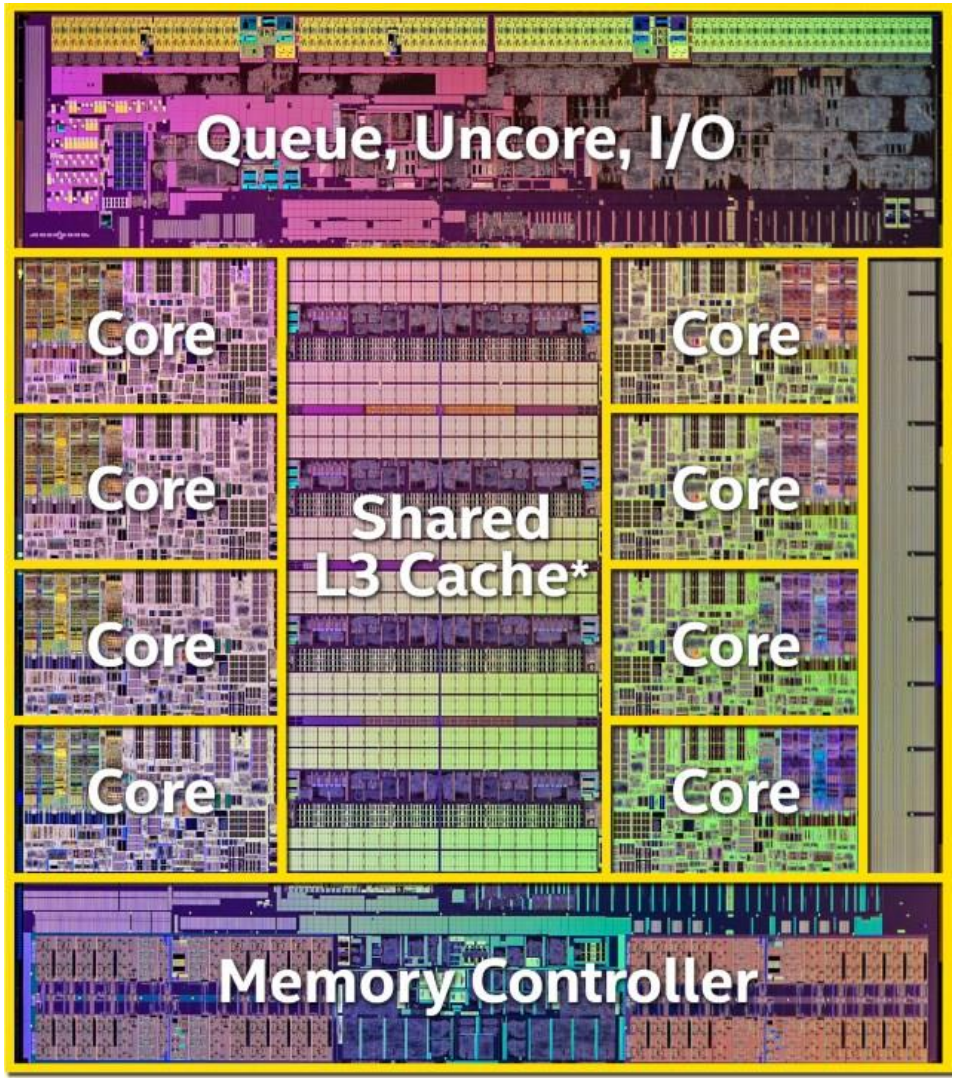


Security Background Outline

- Cryptographic Primitives
- Cryptographic Constructs
- Software Attestation
- Physical Attacks
- Privileged Software Attacks
- Software Attacks on Peripherals
- Address Translation Attacks
- **Cache Timing Attacks**

Cache Timing Attacks

- Practical Considerations
- Theory
- Known Cache Timing Attacks
- Defending against Cache Timing Attacks



http://www.intel.com/

Practical Considerations

The attacker needs access to **performance counters** or some instructions to gain timing knowledge

Shared cache required

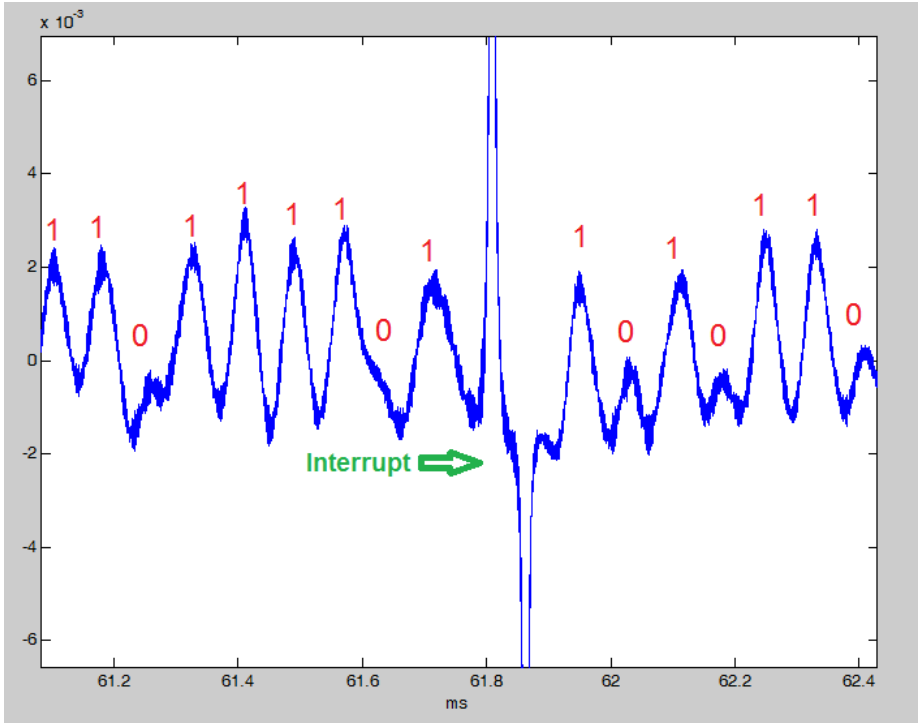
Out-of-Order cores can cause noisy channels

Memory prefetching causes further noise or an even easier channel

Theory 1

1. Attacker accesses memory such that it fills all the ways with the victims interesting memory locations
2. Then the victim accesses cache locations
3. Some location are evicted from the shared cache in this process
4. Remaining cache lines are important!

Attack on AES : AES key leaked via bit leakage



E. Tromer, D. A. Osvik, and A. Shamir, "Efficient cache attacks in AES, and countermeasures," J. Cryptology, no. 2, pp. 37–71, Jan 2010.

Theory 2

1.The Attacker accesses memory such that it fills all the ways with the victims interesting memory locations

2. Then the victim accesses cache locations

3. Some locations miss the cache to go off-chip, the attacker times the accesses to determine this

4. The Attacker knows which cache lines came from the DRAM

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.

Defending against Cache Timing Attacks

- Isolation
  - Static cache partitioning
- Page separation
  - Isolation page tables
  - Use integrity + freshness checks on every access
- The use of Oblivious algorithms
  - Add noise to data accesses by adding dummy accesses
  - The attacker cannot distinguish between private and dummy accesses anymore
- More in store by Chenglu