6.888 Secure Hardware Design

V 0.2

Mengjia Yan

Spring 2022

Today’s Agenda

- Introduce yourself
- Logistics
- Course Overview

Introduce Yourself

Basic Administrivia

- Instructor:
  - Mengjia Yan mengjia@csail.mit.edu
  - Office: 32G-840
  - Office Hours: By Appointment
- TA:
  - Joseph Ravichandran jravi@mit.edu
  - Office: 32-G786
  - Office Hours: Tuesdays 5:00pm to 7:00pm, or by appointment

- Website: http://csg.csail.mit.edu/6.888Yan/
  - Paper readings
  - Syllabus
  - Assignments
- Piazza:
  - Announcements
  - Discussions
- *HotCRP*: Submit paper reviews

Course Logistics

Pre-requisites and Course Organization

- Pre-requisite:
  - Basic computation structure course (**6.004**)

- Study hardware security problems, Research-oriented

- Each topic consists
  - An Overview Lecture
  - 1-2 Paper Discussion Sessions
  - A Lab Assignment

Course Website

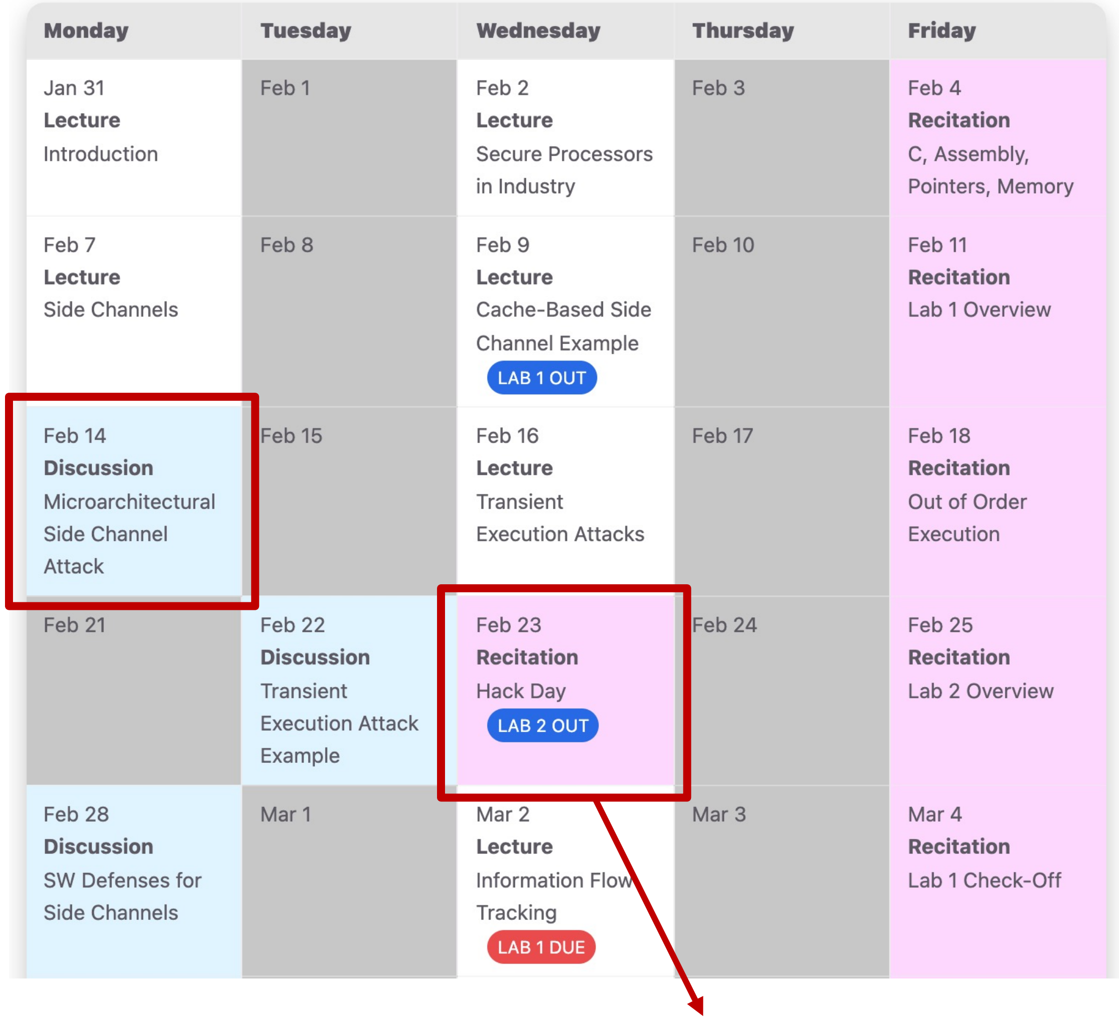
http://csg.csail.mit.edu/6.888Yan/

Generally, paper discussions are scheduled on Monday, except for holidays

Recording:

- Lectures and some recitations will be recorded.
- Paper discussions will not be recorded.

Example by Mengjia



Monday	Tuesday	Wednesday	Thursday	Friday
Jan 31 **Lecture** Introduction	Feb 1	Feb 2 **Lecture** Secure Processors in Industry	Feb 3	Feb 4 **Recitation** C, Assembly, Pointers, Memory
Feb 7 **Lecture** Side Channels	Feb 8	Feb 9 **Lecture** Cache-Based Side Channel Example LAB 1 OUT	Feb 10	Feb 11 **Recitation** Lab 1 Overview
Feb 14 **Discussion** Microarchitectural Side Channel Attack	Feb 15	Feb 16 **Lecture** Transient Execution Attacks	Feb 17	Feb 18 **Recitation** Out of Order Execution
Feb 21	Feb 22 **Discussion** Transient Execution Attack Example	Feb 23 **Recitation** Hack Day LAB 2 OUT	Feb 24	Feb 25 **Recitation** Lab 2 Overview
Feb 28 **Discussion** SW Defenses for Side Channels	Mar 1	Mar 2 **Lecture** Information Flow Tracking LAB 1 DUE	Mar 3	Mar 4 **Recitation** Lab 1 Check-Off

Hack Day only for Lab 1. Note that the other labs **do not** have hack days.

Assignments and Grading

- Paper reviews (~1 paper/week) - 15%
  - summary + 1-2 discussion questions
  - An example on Feb 14 Monday -> First paper summary DUE on Feb 13

- Discussion - 15%
  - Discussion lead for 1 paper - 10%
  - Participation - 5%

- 5 Lab assignments - 70%
  - Each lab - 14%
  - A final project can replace labs 3, 4, and 5, and will be worth 42%

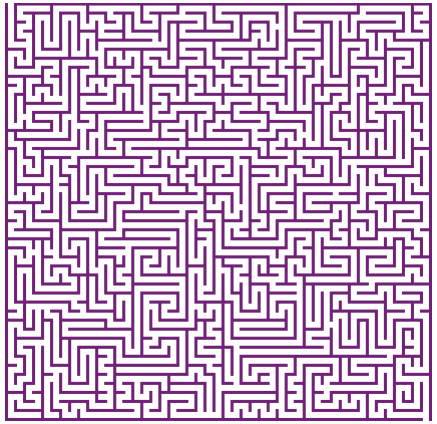
Discussion Format

- Every student will write a review for each paper
  - summary, comments on pros and cons, and key takeaways
  - 1-2 discussion questions
  - Due @midnight before each class
  - Submit via HotCRP (You can see others reviews (anonymous) after submitting yours)

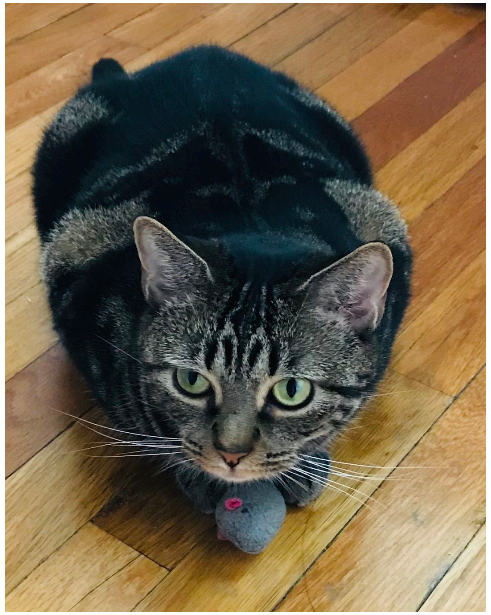
- Each paper will have one student as the lead presenter
  - 45-min presentation content + Discussion
  - **Send slides to me 24 hours before the lecture**

Hardware Security: The Evil and The Good

- Attack modern processors to understand HW vulnerabilities



- Know how to design defenses better



5 Lab Assignments

- Attacks on real processors:
  1. Cache-based Side Channel Attack
  2. Speculative Execution Attack
  3. Website Fingerprinting Attack
  4. Rowhammer Attack
  5. ASLR Bypassing

Lab Contributors

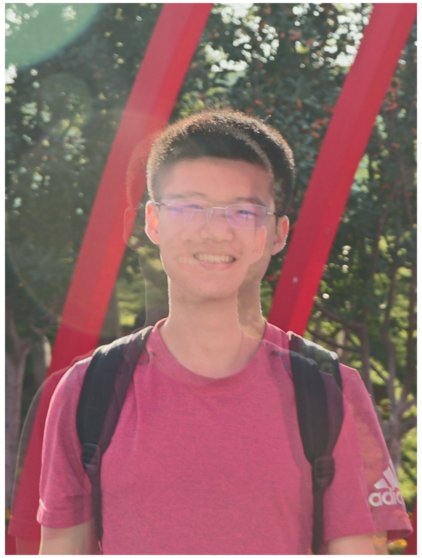
Joseph Ravichandran

Peter Deutsch

Jack Cook

Weon Taek Na

Miguel Gomez-Garcia



Yuheng Yang

Mengyuan Li

Final Project

- Original research project

- Deliverables
  - Proposal (schedule pre-proposal meetings with me)
  - Weekly report (short and informal)
  - Final report + Final presentation

- Open-ended topics
  - Must have some hardware security angle

Collaboration Policy and Warning

- Discussions are always encouraged.
- You should carefully acknowledge all contributions of ideas by others, whether from classmates or from sources you have read.
- MIT academic integrity guidelines

Warning

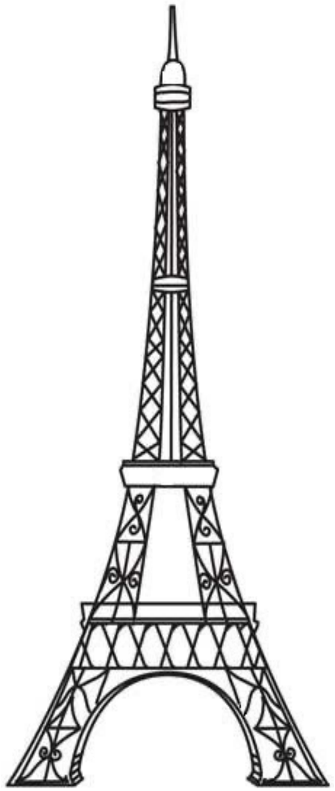
- Please don’t attack other people’s computers or information without their prior permission.
- MIT network rules

Course Overview

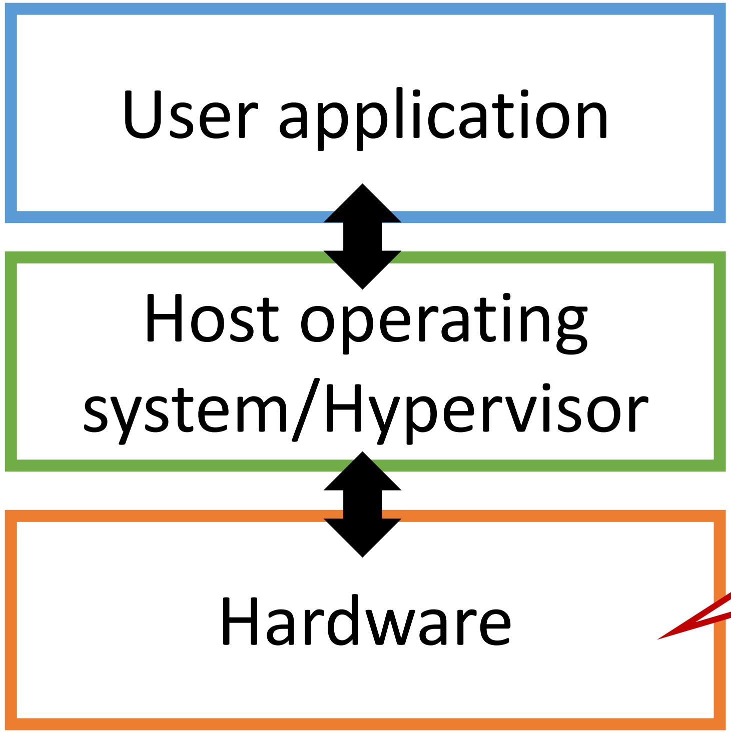
Refresh Basic Computer Architecture

On blackboard

Threat Model and Why Hardware Security?



Computing Systems



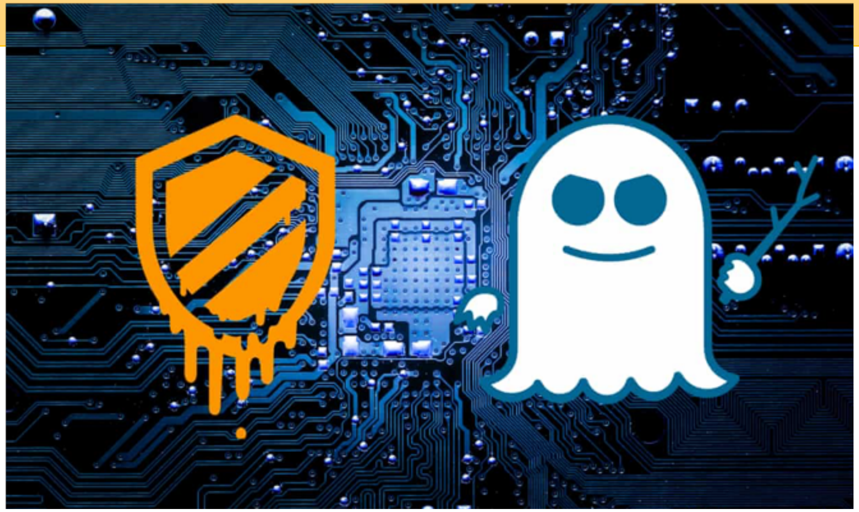
What do you trust when running a bank App on your mobile?

Trusted Computing Base (TCB)

Meltdown & Spectre on the Headlines in 2018

Meltdown and Spectre: ‘worst ever’ CPU bugs affect virtually all computers

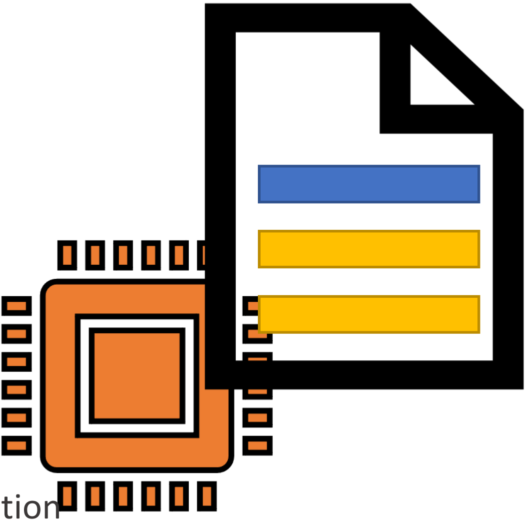
Everything from smartphones and PCs to cloud computing affected by major security flaw found in Intel and other processors – and fix could slow devices.



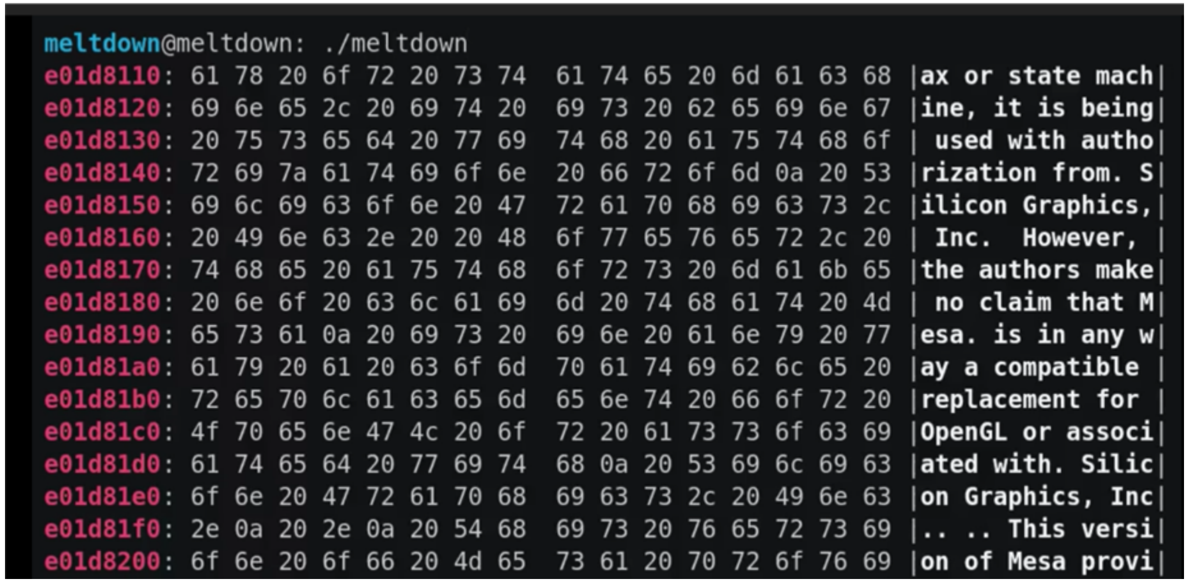
Quotes from https://www.theguardian.com/technology/2018/jan/04/meltdown-spectre-worst-cpu-bugs-ever-found-affect-computers-intel-processors-security-flaw

It is not a bug!

The attacks target the key micro-architecture mechanism of processors: speculative execution.

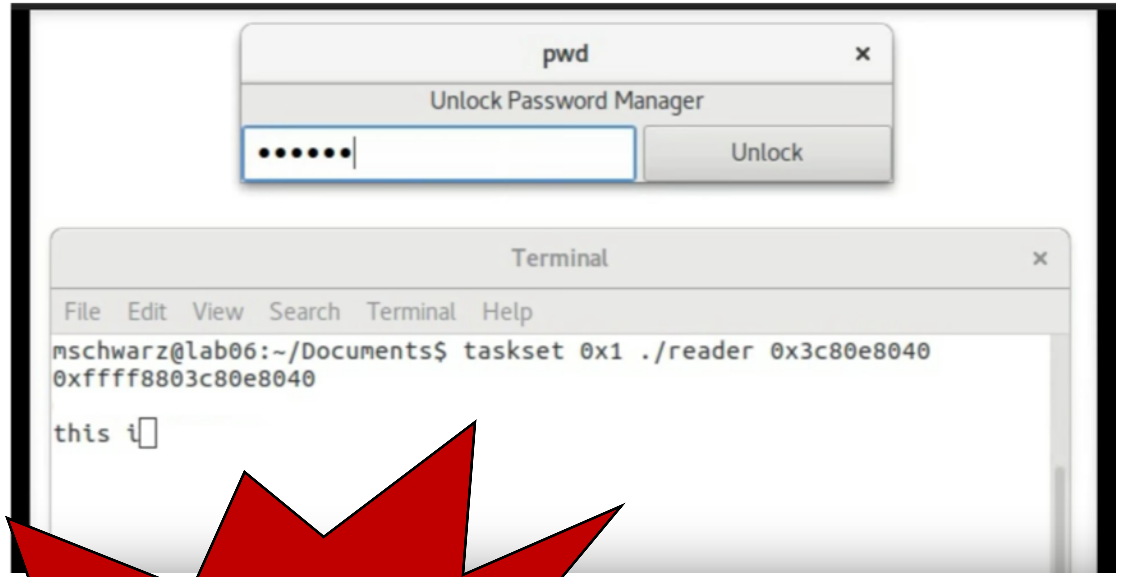


Meltdown & Spectre Break Memory Isolation



```
meltdown@meltdown: ./meltdown
e01d8110: 61 78 20 6f 72 20 73 74  61 74 65 20 6d 61 63 68 |ax or state mach|
e01d8120: 69 6e 65 2c 20 69 74 20  69 73 20 62 65 69 6e 67 |ine, it is being|
e01d8130: 20 75 73 65 64 20 77 69  74 68 20 61 75 74 68 6f | used with autho|
e01d8140: 72 69 7a 61 74 69 6f 6e  20 66 72 6f 6d 0a 20 53 |rization from. S|
e01d8150: 69 6c 69 63 6f 6e 20 47  72 61 70 68 69 63 73 2c |ilicon Graphics,|
e01d8160: 20 49 6e 63 2e 20 20 48  6f 77 65 76 65 72 2c 20 | Inc.  However, |
e01d8170: 74 68 65 20 61 75 74 68  6f 72 73 20 6d 61 6b 65 |the authors make|
e01d8180: 20 6e 6f 20 63 6c 61 69  6d 20 74 68 61 74 20 4d | no claim that M|
e01d8190: 65 73 61 0a 20 69 73 20  69 6e 20 61 6e 79 20 77 |esa. is in any w|
e01d81a0: 61 79 20 61 20 63 6f 6d  70 61 74 69 62 6c 65 20 |ay a compatible |
e01d81b0: 72 65 70 6c 61 63 65 6d  65 6e 74 20 66 6f 72 20 |replacement for |
e01d81c0: 4f 70 65 6e 47 4c 20 6f  72 20 61 73 73 6f 63 69 |OpenGL or associ|
e01d81d0: 61 74 65 64 20 77 69 74  68 0a 20 53 69 6c 69 63 |ated with. Silic|
e01d81e0: 6f 6e 20 47 72 61 70 68  69 63 73 2c 20 49 6e 63 |on Graphics, Inc|
e01d81f0: 2e 0a 20 2e 0a 20 54 68  69 73 20 76 65 72 73 69 |.. .. This versi|
e01d8200: 6f 6e 20 6f 66 20 4d 65  73 61 20 70 72 6f 76 69 |on of Mesa provi|
```

Dump kernel memory content from an ***unprivileged*** *user process.*



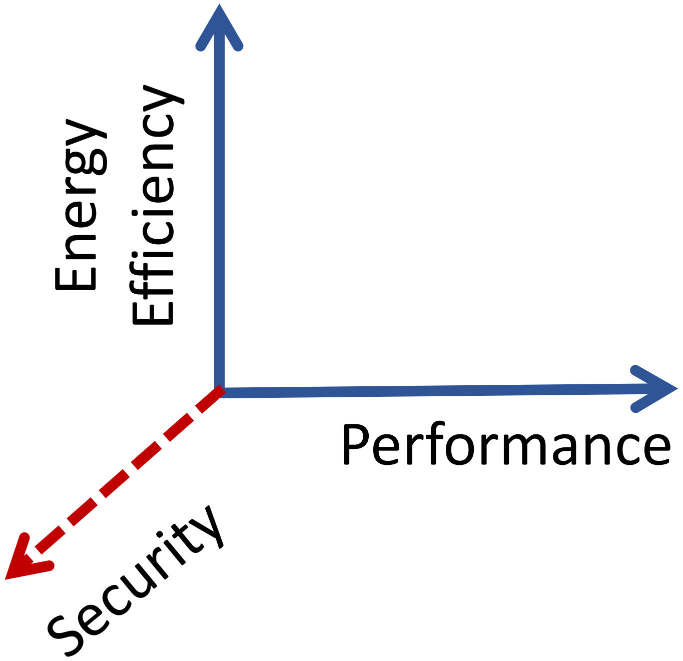
```
mschwarz@lab06:~/Documents$ taskset 0x1 ./reader 0x3c80e8040
0xffff8803c80e8040

this i
```

ther programs.

NO TRACE!

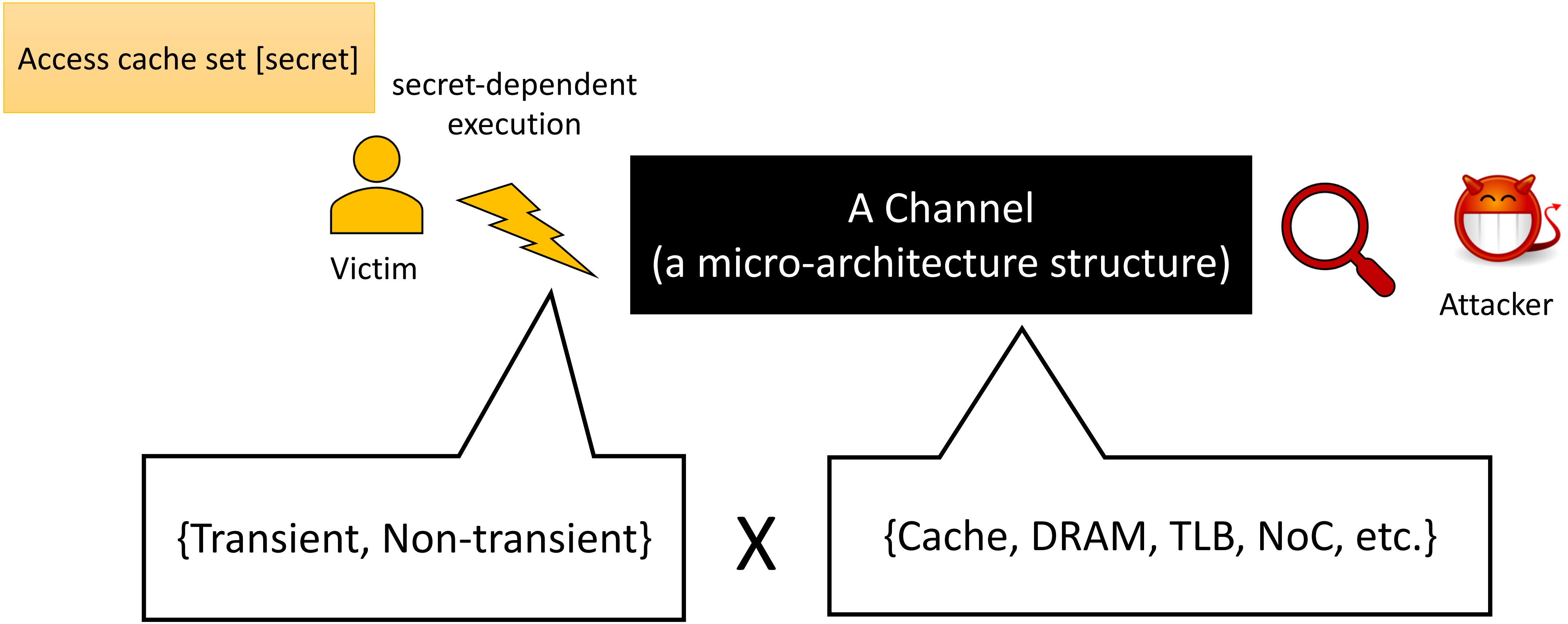
Why We Have Many Hardware Vulnerabilities?



Computer Architecture Design Goals

Preview of Selected Topics

Micro-architecture Side Channels



Access cache set [secret]

secret-dependent execution

Victim

A Channel
(a micro-architecture structure)

Attacker

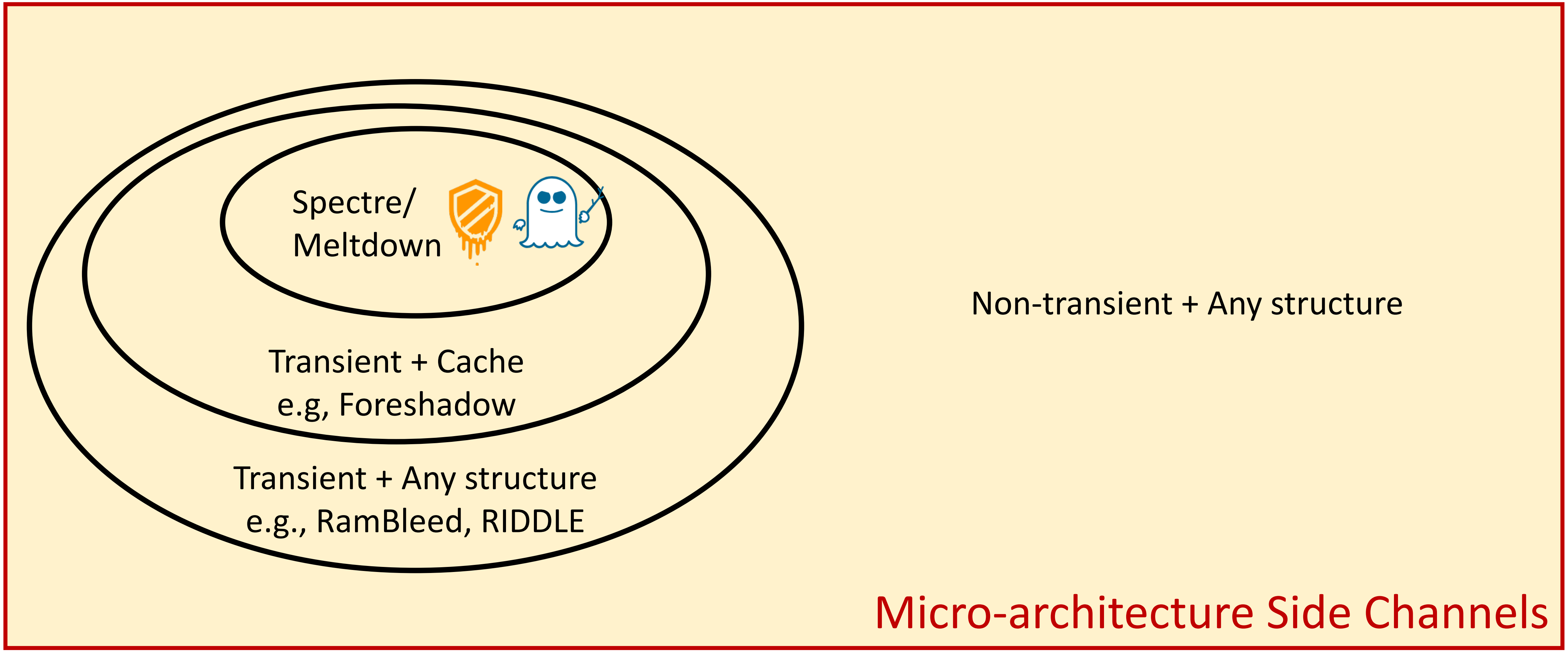
{Transient, Non-transient}

X

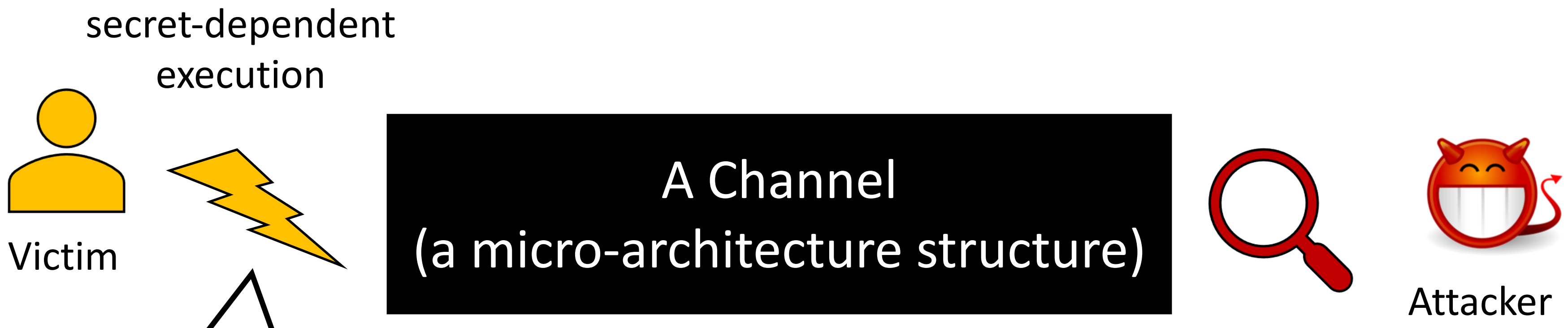
{Cache, DRAM, TLB, NoC, etc.}

[*] *Kiriansky et al. DAWG: a defense against cache timing attacks in speculative execution processors. MICRO'18*

Micro-architecture Side Channel


Spectre/
Meltdown
Transient + Cache
e.g, Foreshadow
Transient + Any structure
e.g., RamBleed, RIDDLE
Non-transient + Any structure
Micro-architecture Side Channels

Micro-architecture Side Channels



secret-dependent execution

Victim

A Channel
(a micro-architecture structure)

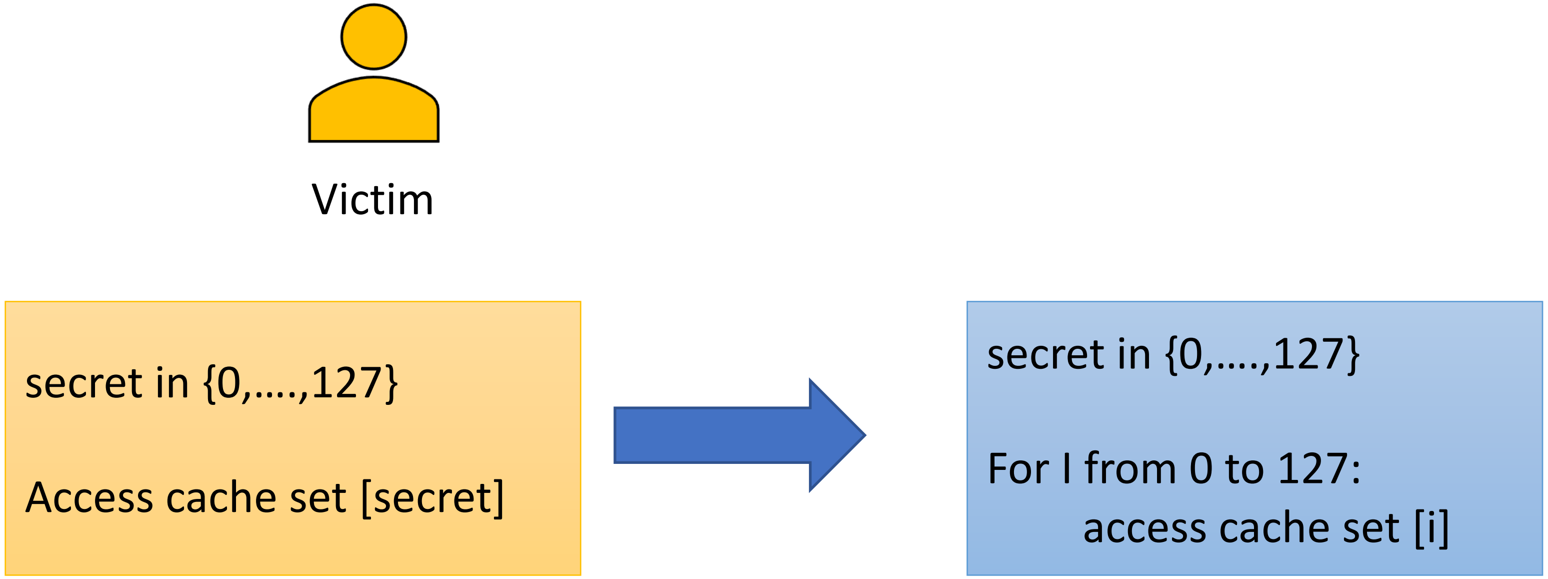
Attacker

Defenses:

Block creation of signals:
Oblivious execution,
speculative execution defenses, etc.

[*] *Kiriansky et al. DAWG: a defense against cache timing attacks in speculative execution processors. MICRO’18*

Oblivious Programming



Victim

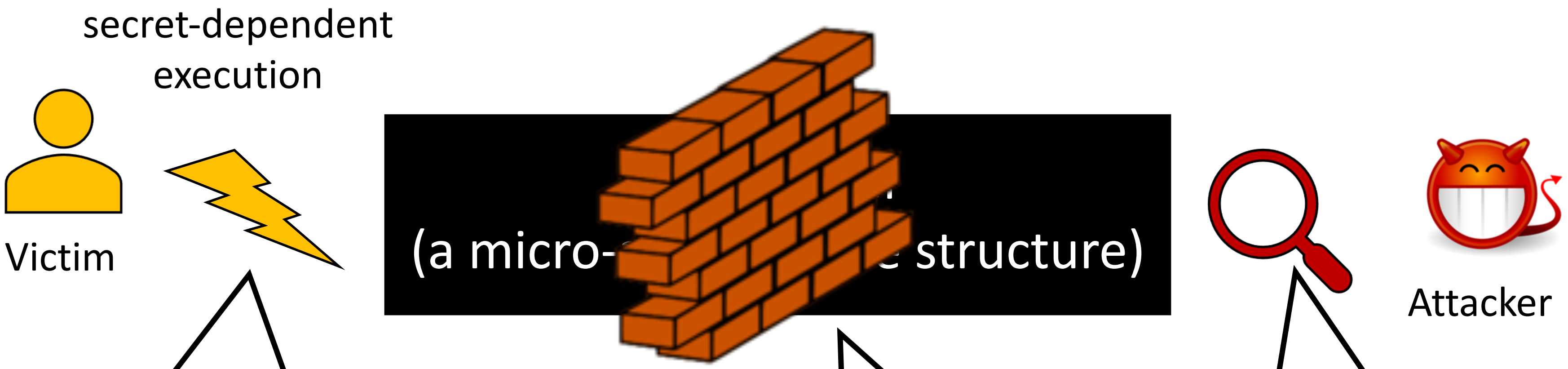
secret in {0,….,127}

Access cache set [secret]

```
secret in {0,….,127}

For I from 0 to 127:
      access cache set [i]
```

Micro-architecture Side Channels



Defenses:

Block creation of signals:
Oblivious execution,
speculative execution defenses, etc.

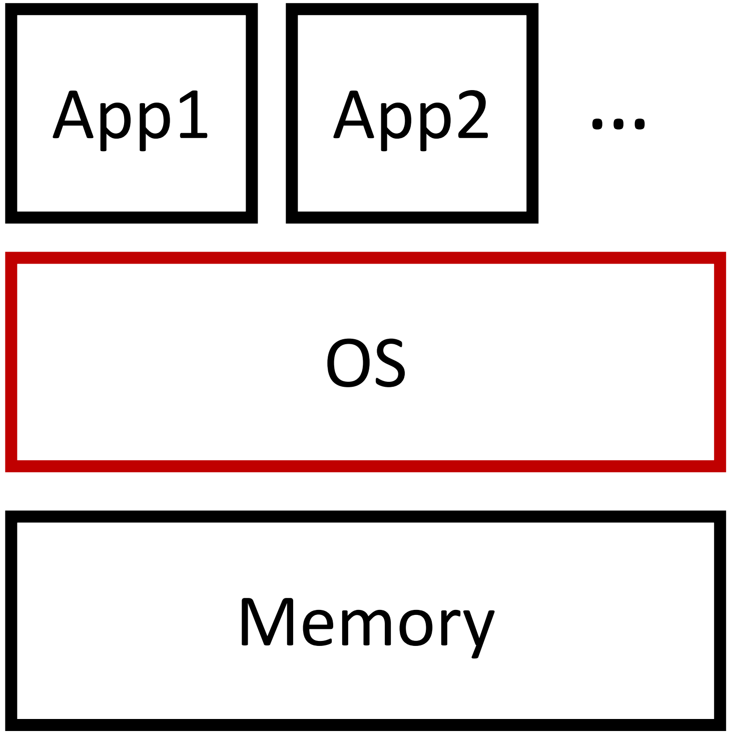
Close the channel:
Isolation, etc.

Block detection of signals:
Randomization, etc.

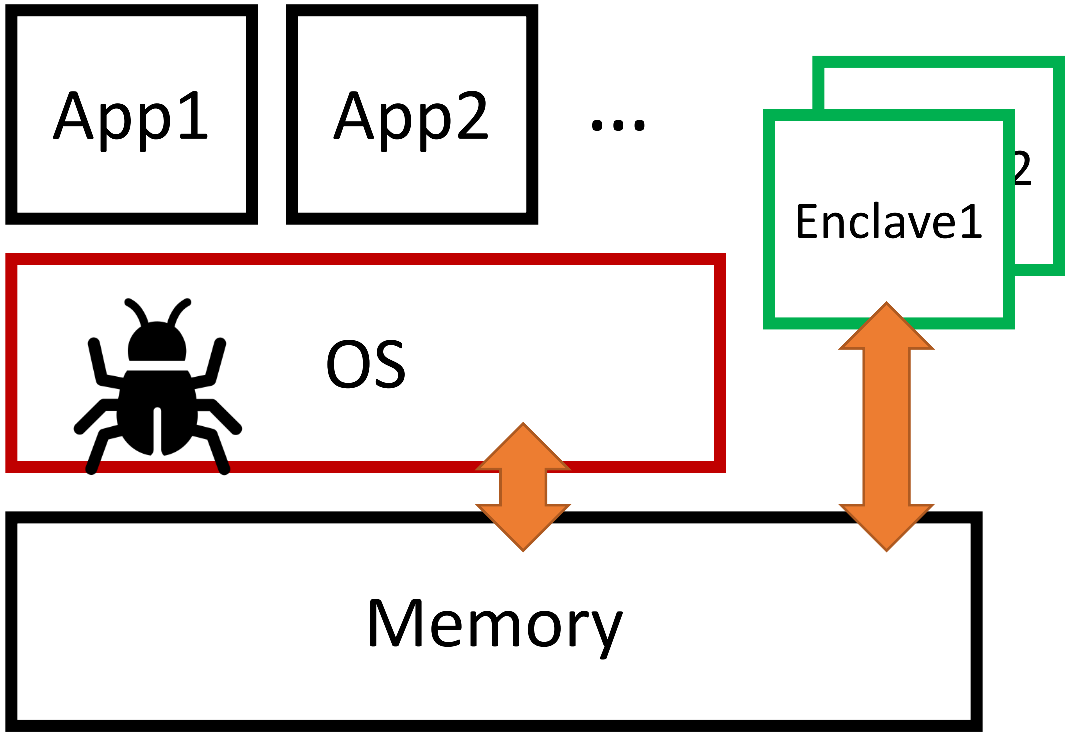
[*] *Kiriansky et al. DAWG: a defense against cache timing attacks in speculative execution processors. MICRO'18*

Enclaves

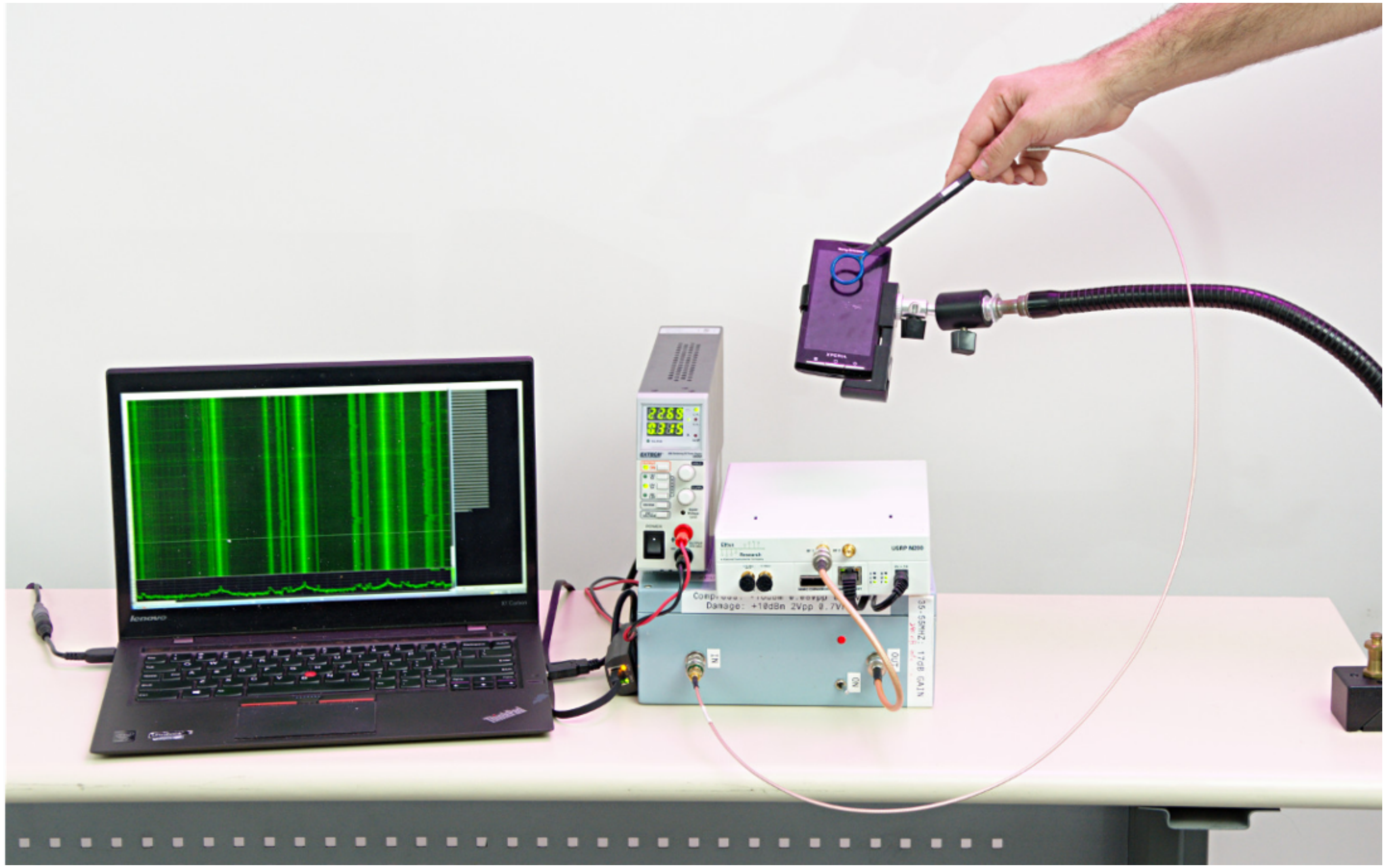
Process Isolation



Enclave Isolation



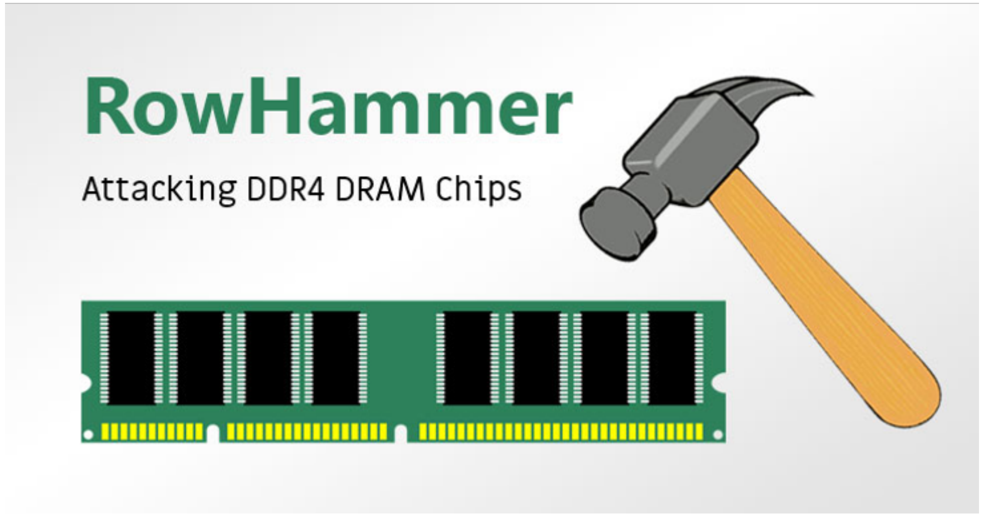
Physical Attacks

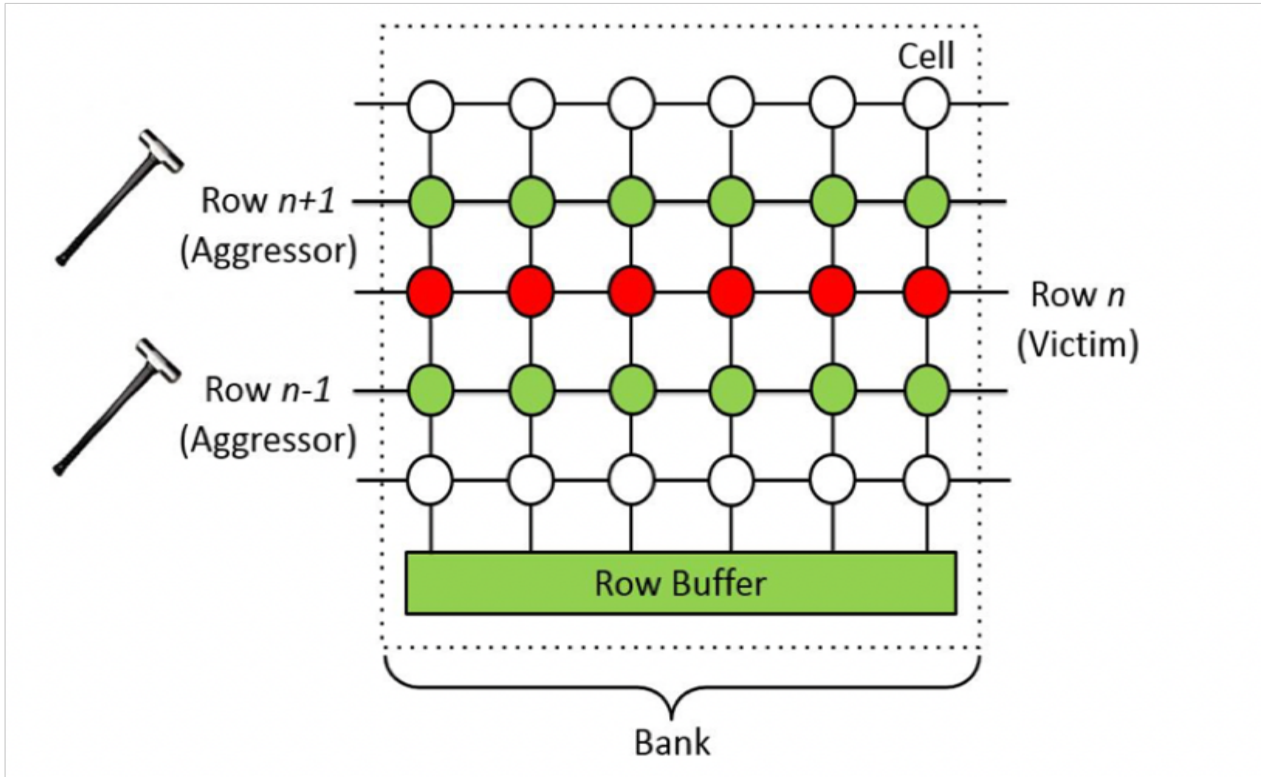


ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels

Physical Attacks

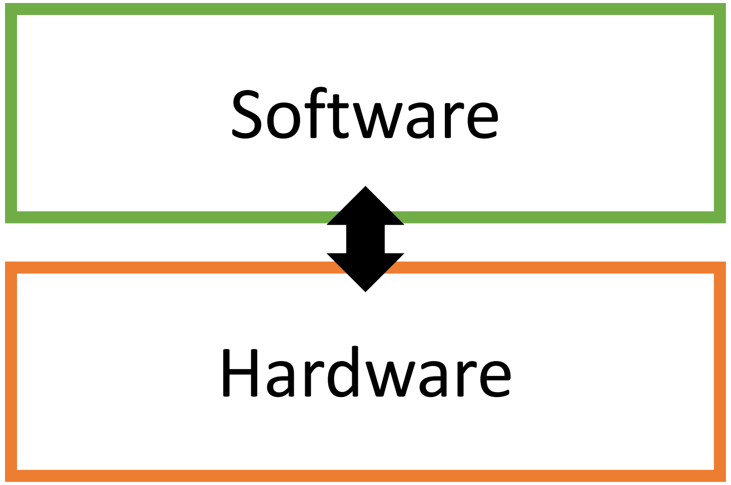
- Modern physical side channels can be done remotely





Memory Safety

- Classical memory corruptions bugs
  - E.g., buffer overflow
- HW: accelerators for security checks
- A more interesting question: what is a good abstraction?



Next:

Secure Processors in Industry