Chapter 8 - Software-based Power Attacks

Attacking CPUs with Power Side Channels from Software

Mathias Oberhuber

3rd April 2025

CPU Power Management

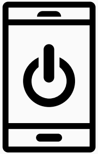
- CPU power management is complex

CPU Power Management

- CPU power management is complex
- In order to save power, you can …

CPU Power Management

- CPU power management is complex
- In order to save power, you can …

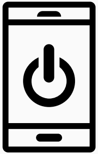


Shut down resources

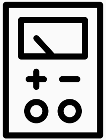
Mathias Oberhuber

CPU Power Management

- CPU power management is complex
- In order to save power, you can ...

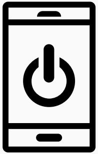


Shut down resources

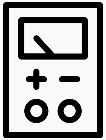


Reduce voltage

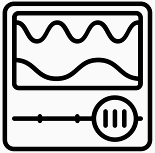
- CPU power management is complex
- In order to save power, you can ...



Shut down resources

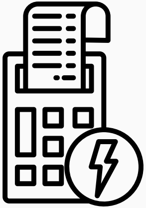


Reduce voltage



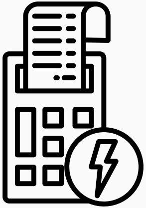
Reduce frequency

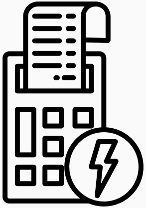
- Therefore, the CPU requires:



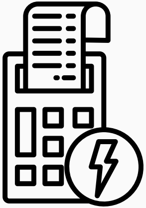
Intel RAPL

- Therefore, the CPU requires:
  - Thermal Management



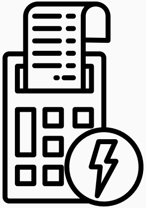


- Therefore, the CPU requires:
  - Thermal Management
  - Platform Power Limiting



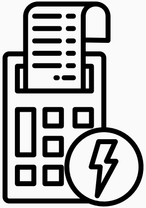
- Therefore, the CPU requires:
    - Thermal Management
    - Platform Power Limiting
    - Power/Performance Budgeting

Intel RAPL



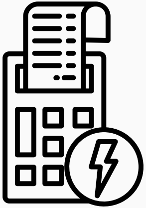
- Therefore, the CPU requires:
    - Thermal Management
    - Platform Power Limiting
    - Power/Performance Budgeting
- Domains: PKG, CORE, MC

Intel RAPL

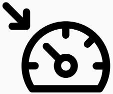


- Therefore, the CPU requires:
    - Thermal Management
    - Platform Power Limiting
    - Power/Performance Budgeting
- Domains: PKG, CORE, MC
- **Intel Running Average Power Limit** (RAPL) provides:

Intel RAPL

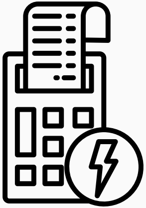


- Therefore, the CPU requires:
  - Thermal Management
  - Platform Power Limiting
  - Power/Performance Budgeting
- Domains: PKG, CORE, MC
- **Intel Running Average Power Limit** (RAPL) provides:

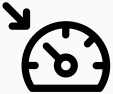


power limiting

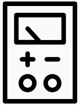
Intel RAPL



- Therefore, the CPU requires:
  - Thermal Management
  - Platform Power Limiting
  - Power/Performance Budgeting
- Domains: PKG, CORE, MC
- **Intel Running Average Power Limit** (RAPL) provides:

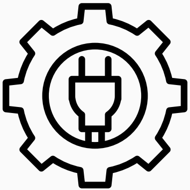


power limiting



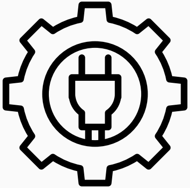
energy reading

Intel RAPL



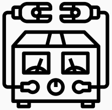
- **Linux**: accessed via `powercap` framework

  `/sys/devices/virtual/powercap/intel-rapl`



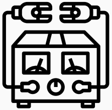
- **Linux**: accessed via `powercap` framework
  `/sys/devices/virtual/powercap/intel-rapl`
- **macOS** and **Windows**: Intel driver needs to be installed

Intel RAPL: Properties



Unprivileged power meter

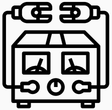
Intel RAPL: Properties



Unprivileged power meter

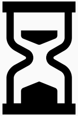
No physical access

Intel RAPL: Properties

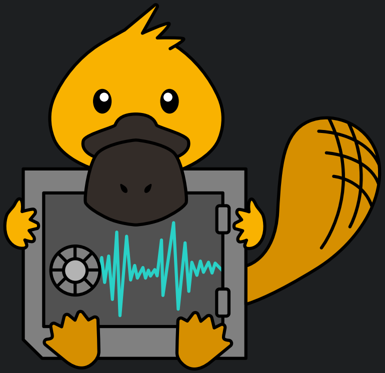


Unprivileged power meter

No physical access

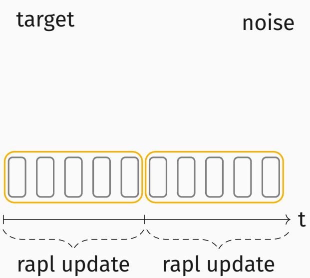


Low refresh rate



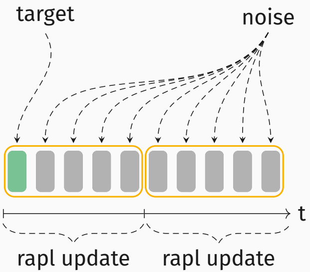
Platypus Attack

RAPL: Measurement Techniques

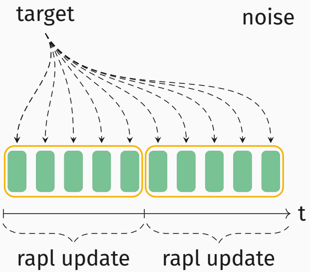


- Measure an **instruction** by

RAPL: Measurement Techniques

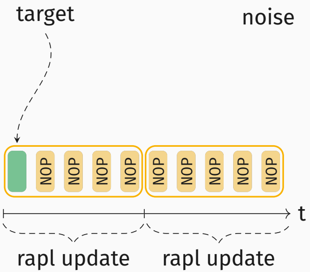


- Measure an **instruction** by
  - executing it once

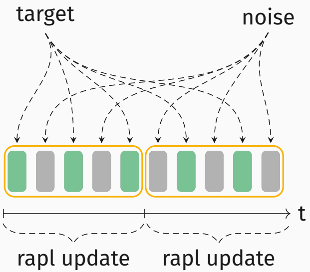


- Measure an **instruction** by
    - executing it once
    - executing it repeatedly

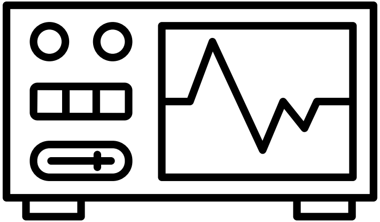
RAPL: Measurement Techniques



- Measure an **instruction** by
  - executing it once
  - executing it repeatedly
  - padding it with known instructions



- Measure an **instruction** by
  - executing it once
  - executing it repeatedly
  - padding it with known instructions
  - reissue the instruction after an interrupt



What can we do with this?

Distinguishing Instructions

- Measure the energy consumption of **different instructions**

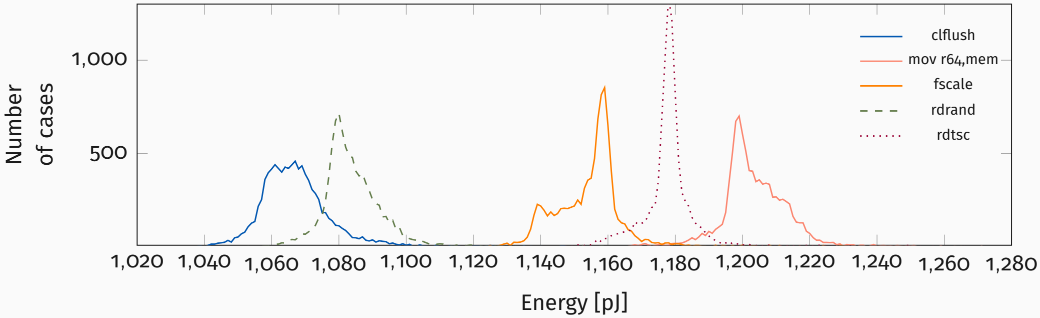
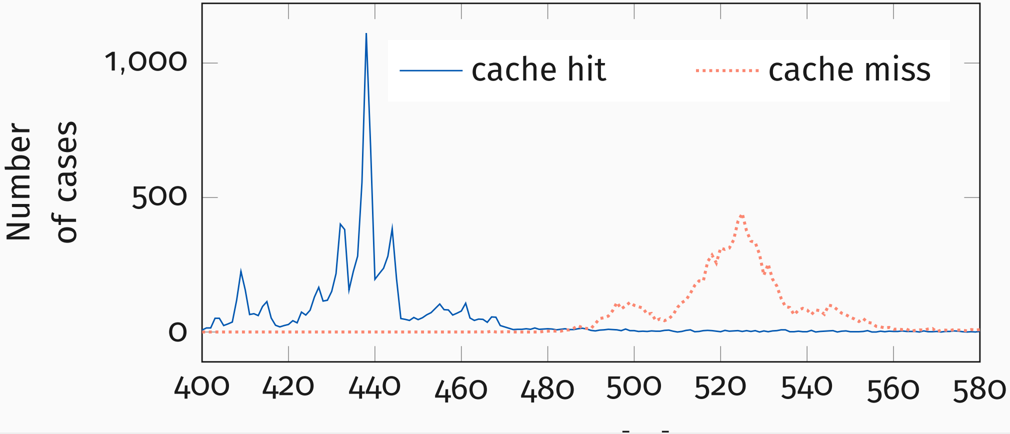


Figure 1: A histogram of the power consumption of various instructions on the i7-6700K (desktop) system.

Distinguishing Load Targets

- Measure the energy consumption of **different load targets**



Distinguishing Operands

- Measure the energy consumption of **different operands**

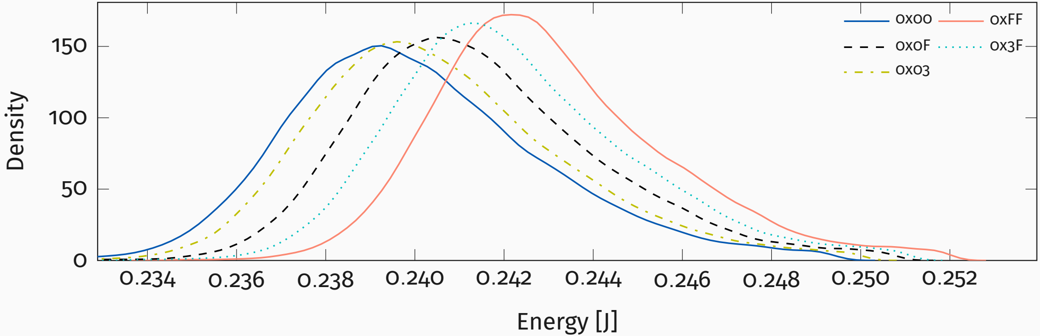
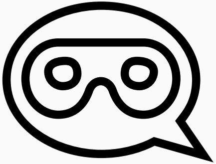


Figure 3: Measured energy consumption of the `imul` instruction with one operand fixed to 8 and the other varying in its Hamming weight.

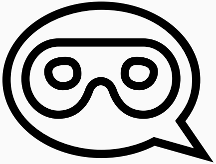
Let’s exploit this!

Covert Channel



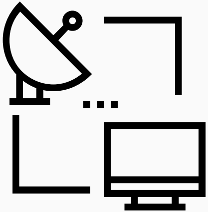
- Hidden communication channel

Covert Channel



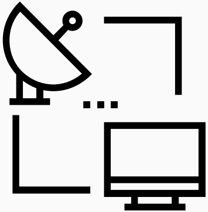
- Hidden communication channel
- Leveraging the power side channel

Covert Channel



- 2 Processes, Sender and Receiver
  - **Send a 1:** Perform energy-consuming instructions
  - **Send a 0:** Idle

Covert Channel



- 2 Processes, Sender and Receiver
  - **Send a 1:** Perform energy-consuming instructions
  - **Send a 0:** Idle
- Receiver measures power consumption

→ **Deduces transmitted bit**

Covert Channel

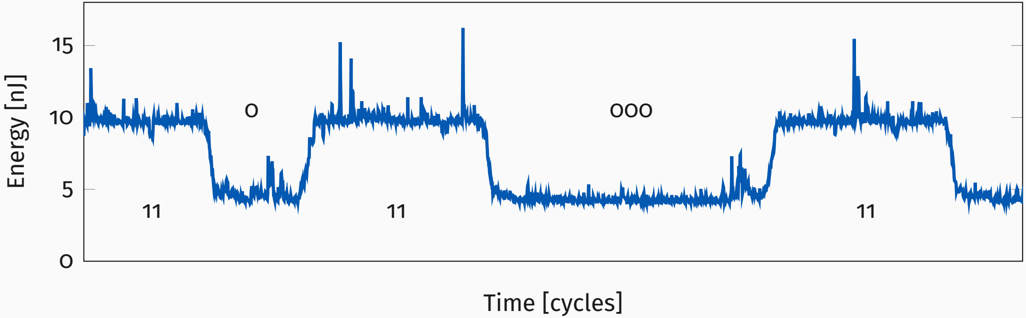


Figure 4: Transmission of bits `1101100011` using the time-less covert channel.

- Kernel Address Space Layout Randomization (KASLR)

- Kernel Address Space Layout Randomization (KASLR)
- Exploit **energy consumption differences** between

Breaking KASLR

- Kernel Address Space Layout Randomization (KASLR)
- Exploit **energy consumption differences** between
  - Mapped addresses

Breaking KASLR

- Kernel Address Space Layout Randomization (KASLR)
- Exploit **energy consumption differences** between
  - Mapped addresses
  - Unmapped addresses

Breaking KASLR

- Kernel Address Space Layout Randomization (KASLR)
- Exploit **energy consumption differences** between
  - Mapped addresses
  - Unmapped addresses
- **Valid address translations** are cached in the **TLB**

Breaking KASLR

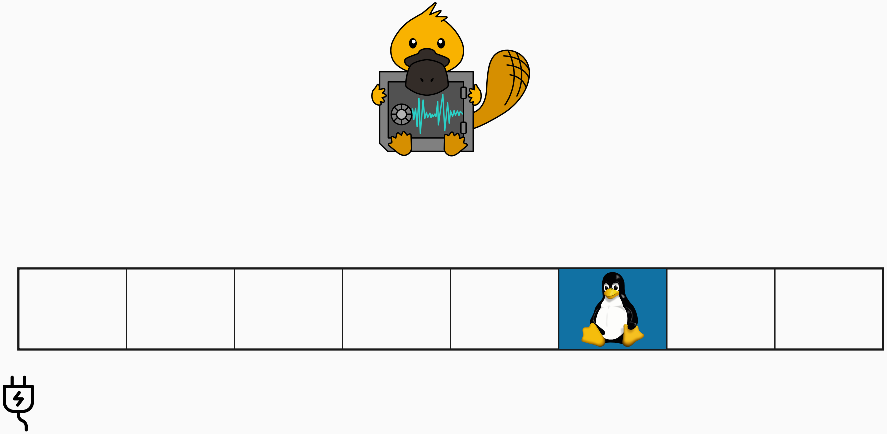


Figure 5: Repeated Page-table walks for unmapped pages require more power

Breaking KASLR

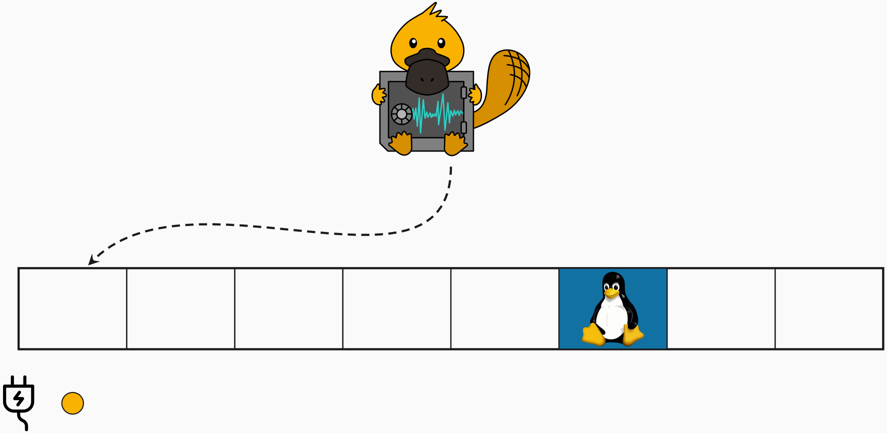


Figure 5: Repeated Page-table walks for unmapped pages require more power

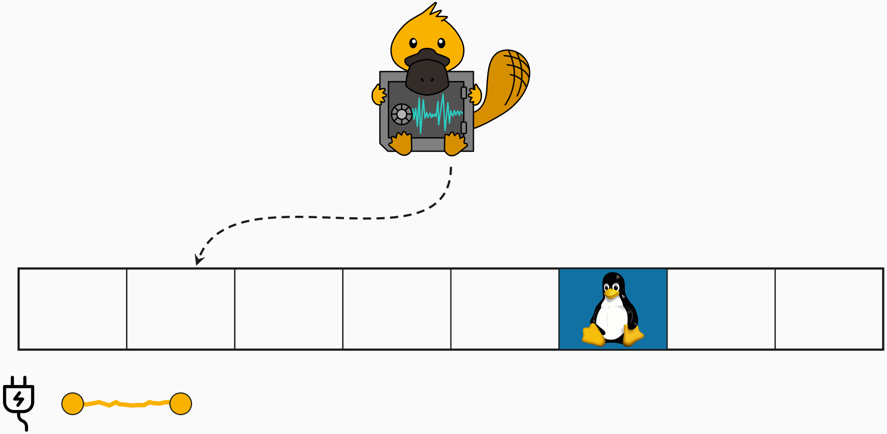


Figure 5: Repeated Page-table walks for unmapped pages require more power

Breaking KASLR

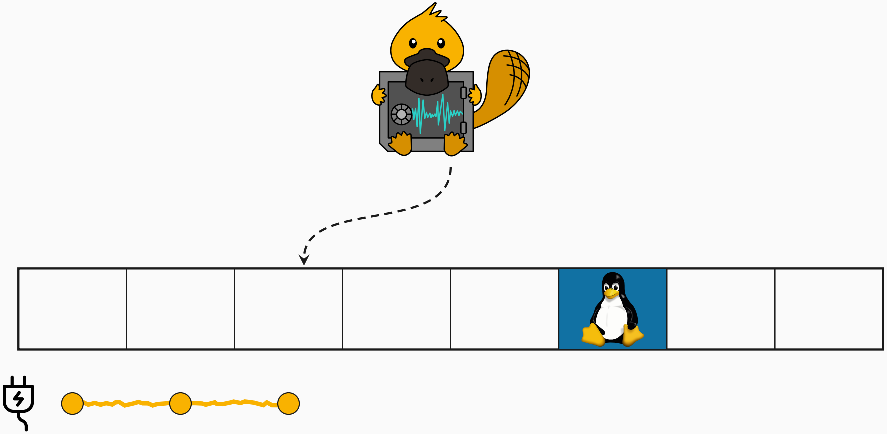


Figure 5: Repeated Page-table walks for unmapped pages require more power

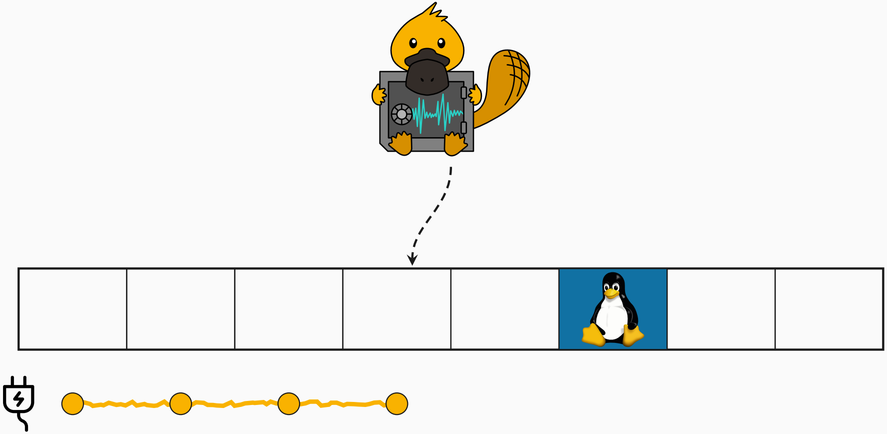


Figure 5: Repeated Page-table walks for unmapped pages require more power

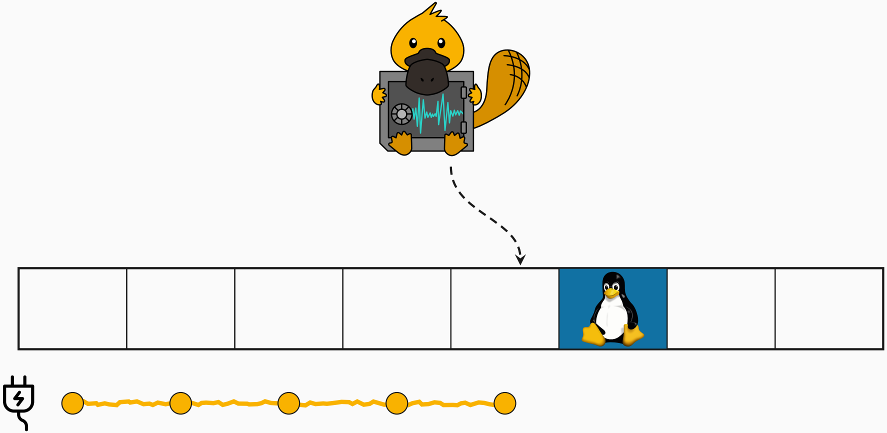


Figure 5: Repeated Page-table walks for unmapped pages require more power

Breaking KASLR

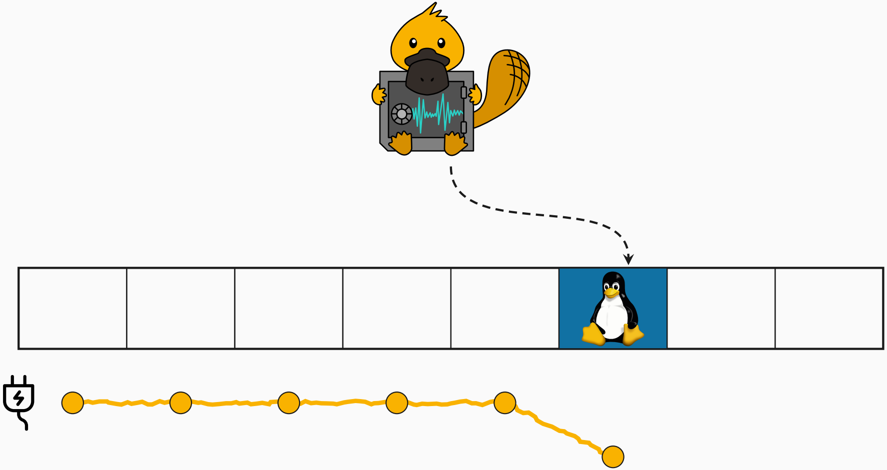


Figure 5: Repeated Page-table walks for unmapped pages require more power

Breaking KASLR

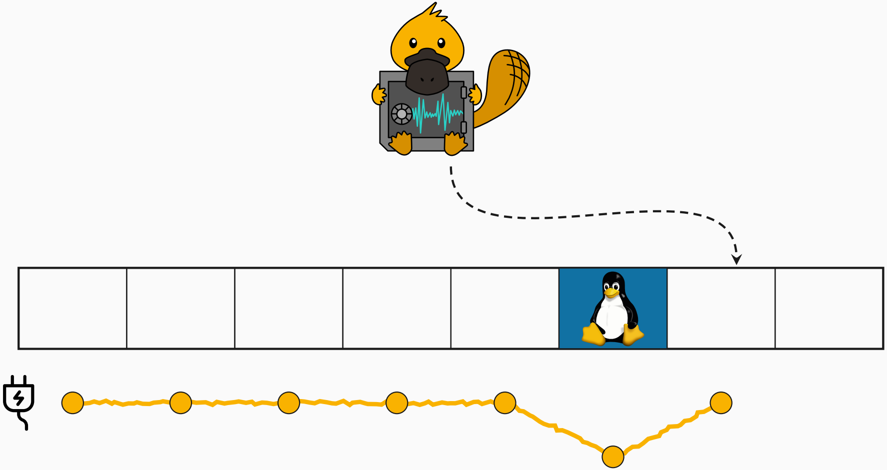


Figure 5: Repeated Page-table walks for unmapped pages require more power

Breaking KASLR

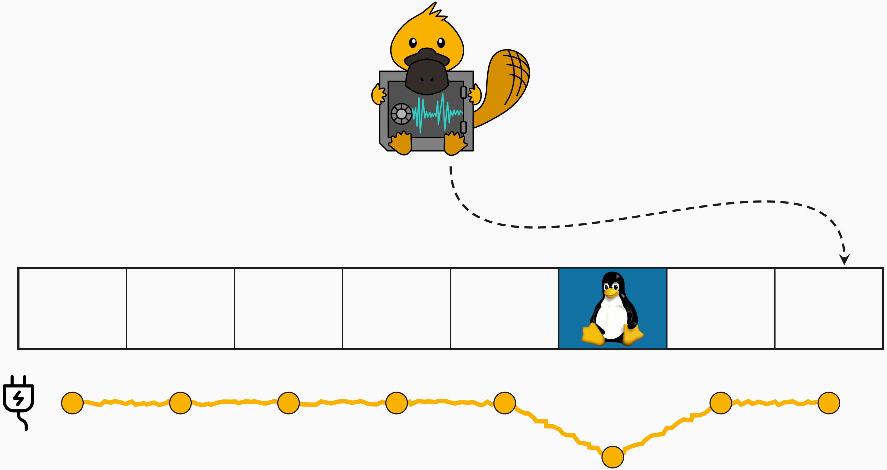
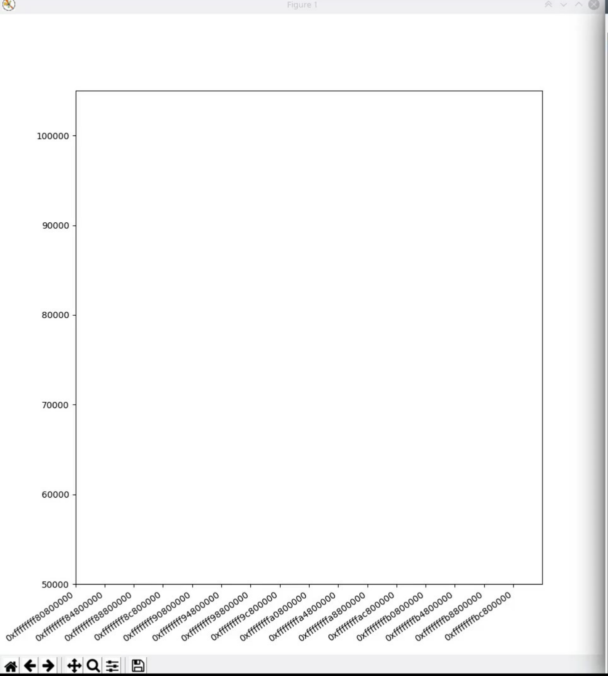


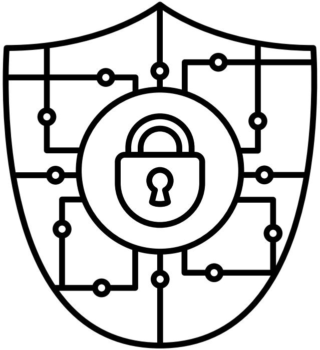
Figure 5: Repeated Page-table walks for unmapped pages require more power



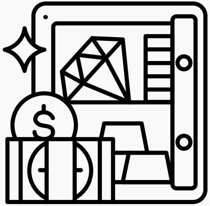
kaslr : zsh — Konsole

File Edit View Bookmarks Settings Help

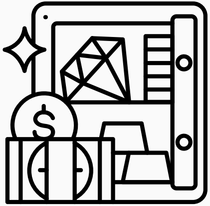
```
michael@hp /tmp/kaslr %
```



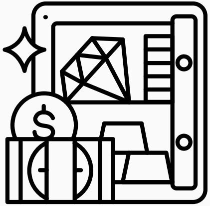
Attacking Intel SGX: RSA Key Recovery



- Instruction-set extension

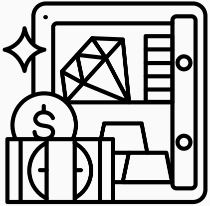


- Instruction-set extension
- Integrity and confidentiality in **untrusted environments**

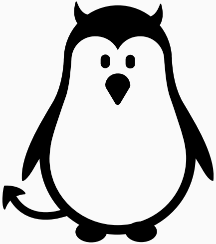


- Instruction-set extension
- Integrity and confidentiality in **untrusted environments**
- **Enclaves** offer protected areas of memory

Attacking Intel SGX

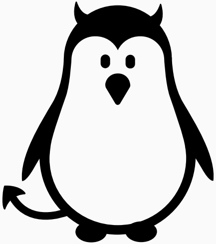


- Instruction-set extension
- Integrity and confidentiality in **untrusted environments**
- **Enclaves** offer protected areas of memory
- Operating system can be **compromised**



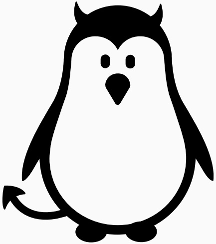
- More power as an evil operating system

Attacking Intel SGX: Threat Model



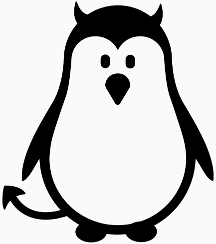
- More power as an evil operating system
- Hook the SGX Enclave exit point

Attacking Intel SGX: Threat Model



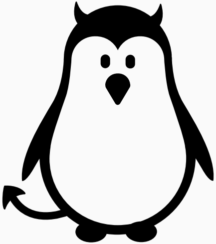
- More power as an evil operating system
- Hook the SGX Enclave exit point
- Directly read out the **RAPL values** from the MSRs

Attacking Intel SGX: Threat Model

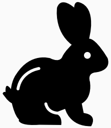


- More power as an evil operating system
- Hook the SGX Enclave exit point
- Directly read out the **RAPL values** from the MSRs
- No operating system overhead!

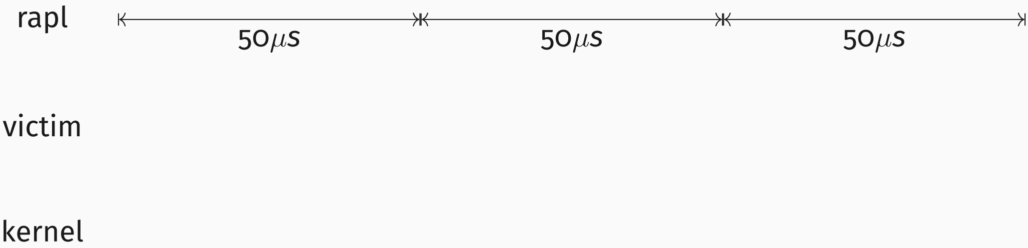
Attacking Intel SGX: Threat Model



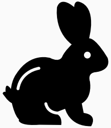
- More power as an evil operating system
- Hook the SGX Enclave exit point
- Directly read out the **RAPL values** from the MSRs
- No operating system overhead!
- Interrupt victim often to increase resolution



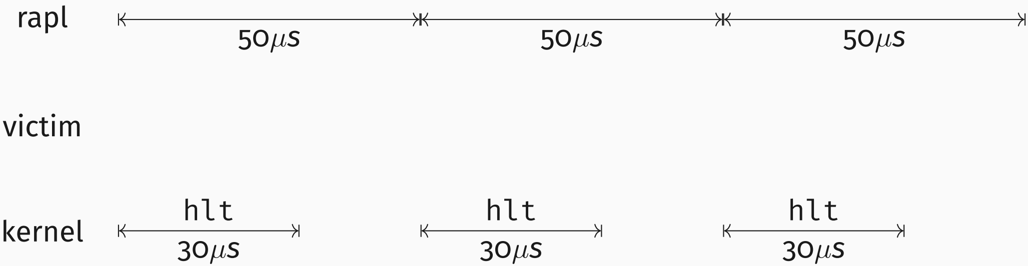
- RAPL domains have a nearly fixed update interval



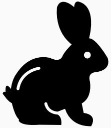
Halt Delay



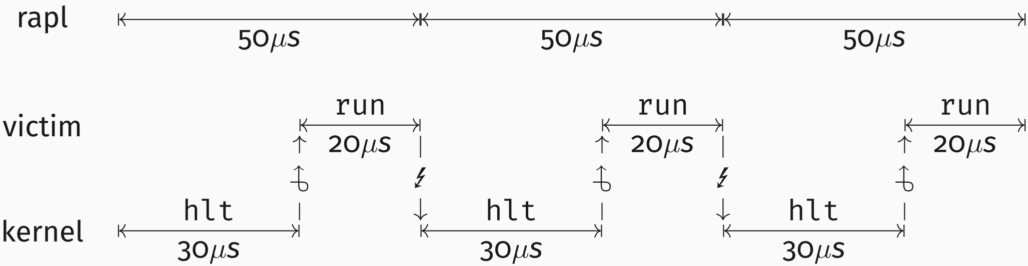
- RAPL domains have a nearly fixed update interval
- Delay the interrupt return with the halt delay in the ISR



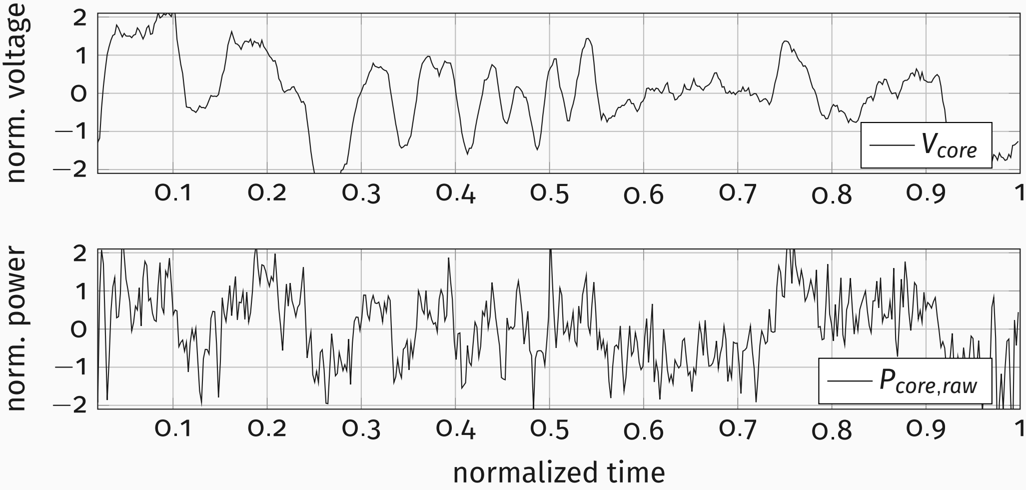
Halt Delay



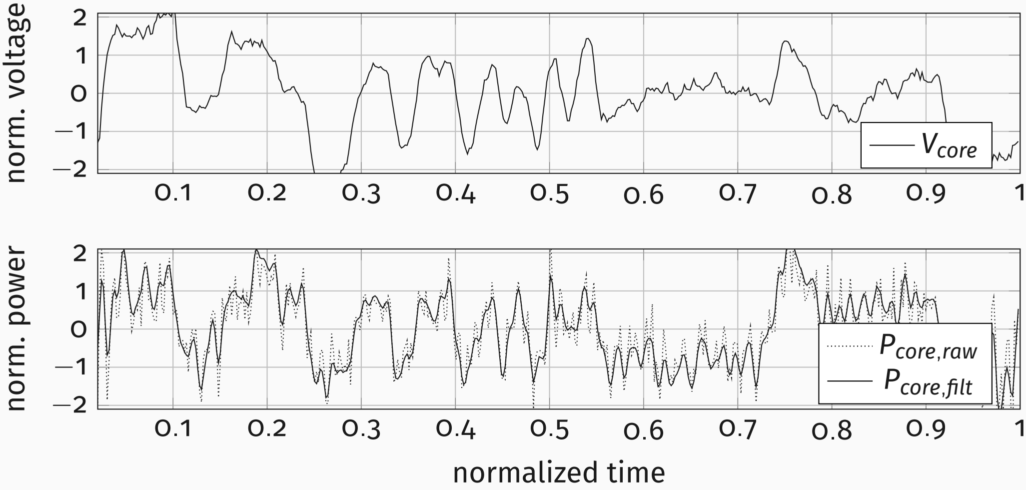
- RAPL domains have a nearly fixed update interval
- Delay the interrupt return with the halt delay in the ISR
- Reduces the execution time of the victim in the current interval



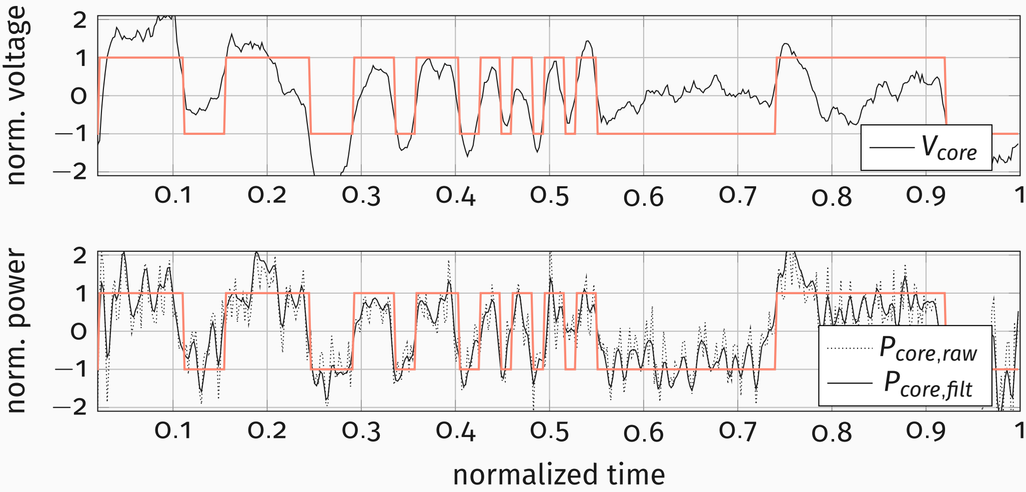
SPA Attack - Results



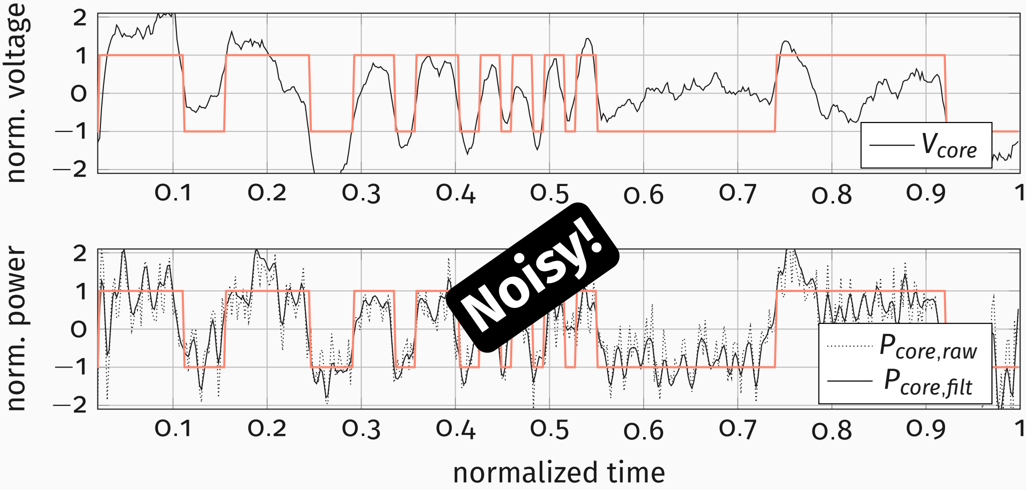
SPA Attack - Results



SPA Attack - Results



SPA Attack - Results



SGX Step

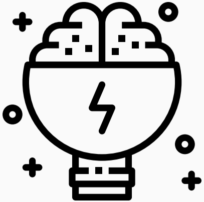
- **SGX-step** is an open-source Linux kernel framework

SGX Step

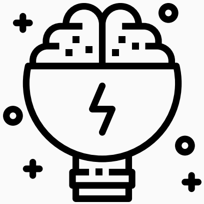
- **SGX-step** is an open-source Linux kernel framework
- Configure APIC timer interrupts

SGX Step

- **SGX-step** is an open-source Linux kernel framework
- Configure APIC timer interrupts
- Single and zero-step enclave execution

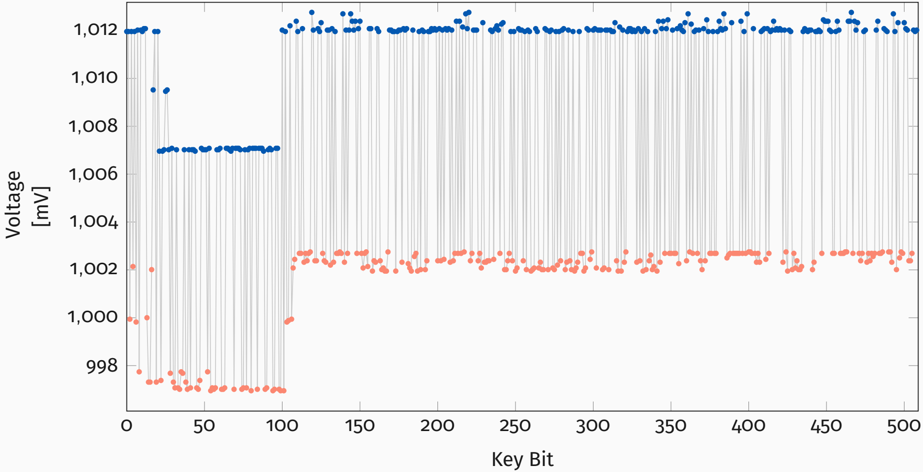


- **Combine** Intel RAPL with SGX-step

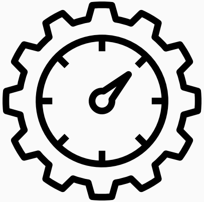


- **Combine** Intel RAPL with SGX-step
- Measure the energy consumption of **single instructions**

Attacking mbed TLS

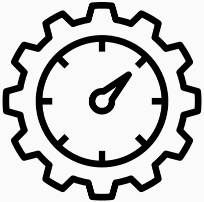


Performance



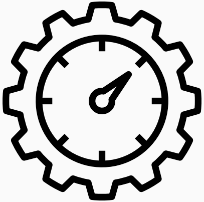
- Time per key bit increases linearly based on the index

Performance



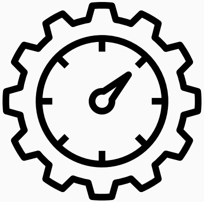
- Time per key bit increases linearly based on the index
- 3h 31m for a **512 bit**

Performance

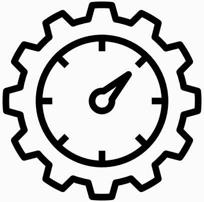


- Time per key bit increases linearly based on the index
- 3h 31m for a **512 bit**
  - 52 minutes for finding target instruction

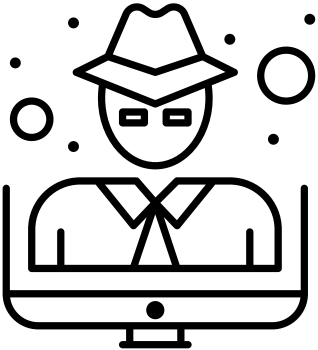
Performance



- Time per key bit increases linearly based on the index
- 3h 31m for a **512 bit**
    - 52 minutes for finding target instruction
- Record 3 samples per key bit

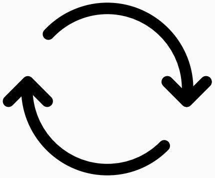


- Time per key bit increases linearly based on the index
- 3h 31m for a **512 bit**
    - 52 minutes for finding target instruction
- Record 3 samples per key bit
    - This could be extend to a single trace attack

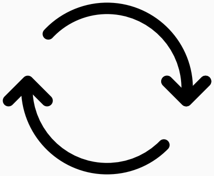


Crypto Attacks from User Space

Repeat Repeat Repeat



- **Difficult** to measure parts without SGX-step



- **Difficult** to measure parts without SGX-step
- **Can** measure over the **overall execution**

Correlation Power Analysis

- Building a power consumption **model** of the device:

- Building a power consumption **model** of the device:

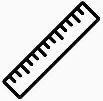
Hamming Weight

Number of bits set

- Building a power consumption **model** of the device:

Hamming Weight

Number of bits set



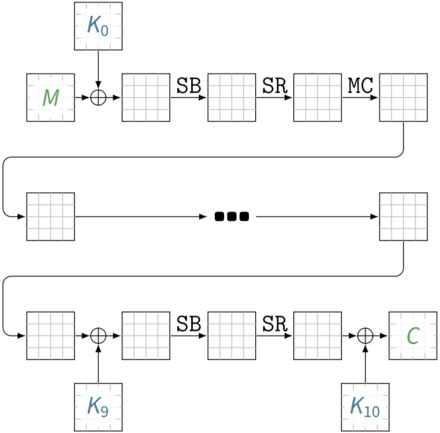
Hamming Distance

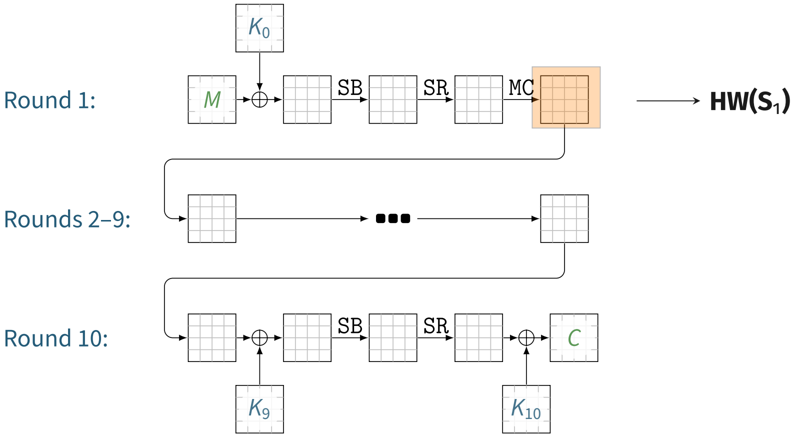
Bits flipping between operations

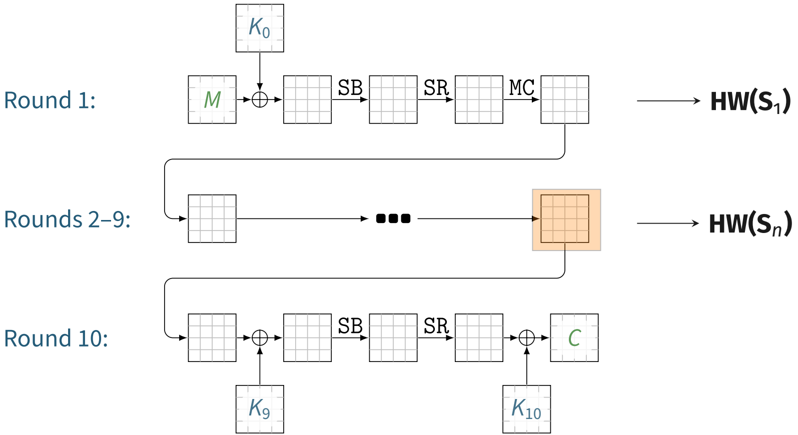
Round 1:

Rounds 2–9:

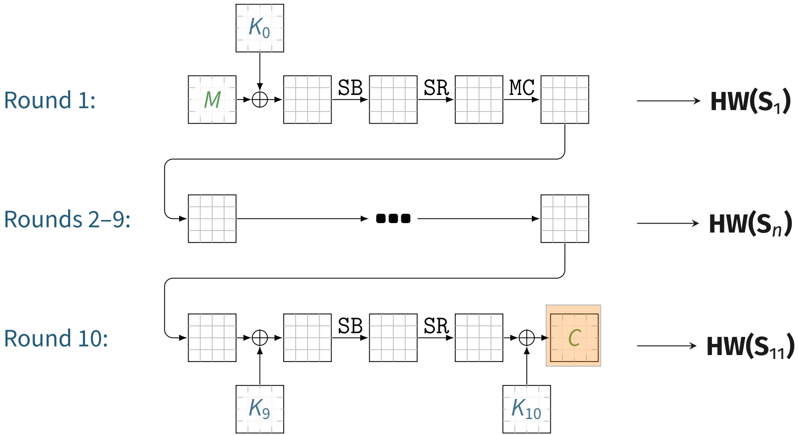
Round 10:




Round 1:
K_0
M
SB
SR
MC
HW(S_1)
Rounds 2–9:
Round 10:
SB
SR
C
K_9
K_{10}



3 / 3

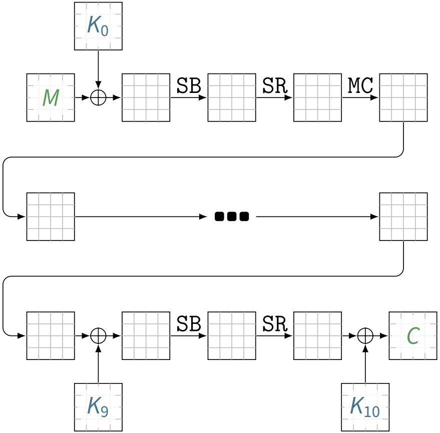


3 / 3

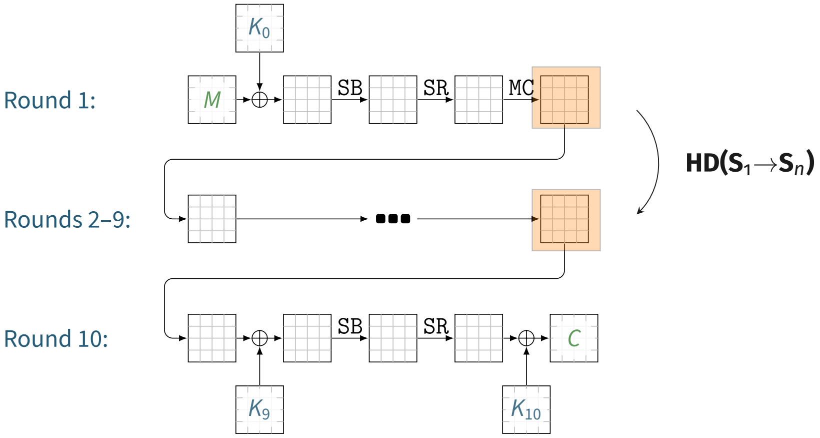
Round 1:

Rounds 2–9:

Round 10:

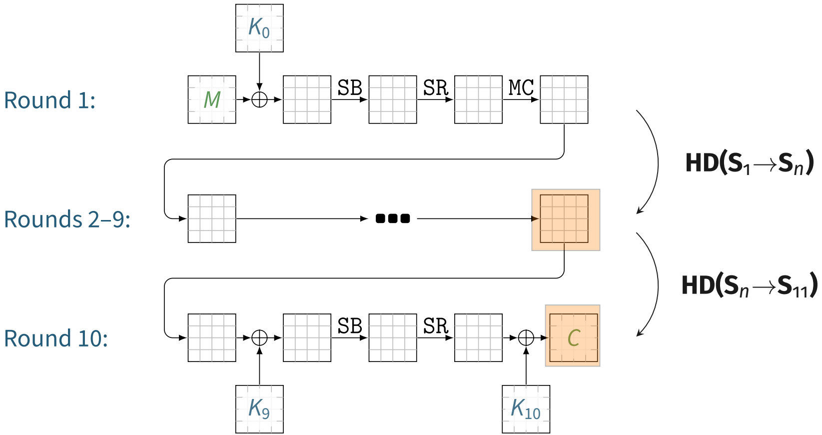


AES



Round 1:
K_0
M
SB
SR
MC
Rounds 2–9:
Round 10:
SB
SR
C
K_9
K_{10}
HD($S_1 \rightarrow S_n$)

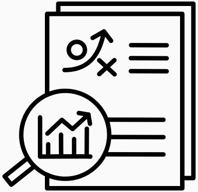
AES



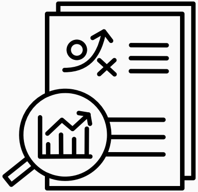
Round 1:
K_0
M
SB
SR
MC
Rounds 2–9:
Round 10:
SB
SR
C
K_9
K_{10}
HD($S_1 \to S_n$)
HD($S_n \to S_{11}$)

3 / 3

Correlation Power Analysis

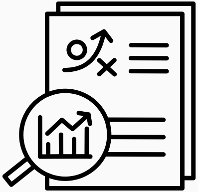


- **AES-NI**: Side-channel resilient instruction-set extension
- Target **AES-NI** in a scenario where we can trigger encryption/decryption of many blocks
  - Disk encryption/decryption
  - TLS
  - (Un)sealing SGX enclave state



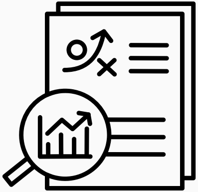
- We control the plain text

Correlation Power Analysis



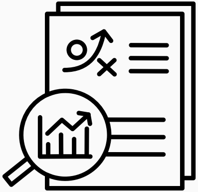
- We control the plain text
- We observe the cipher text

Correlation Power Analysis



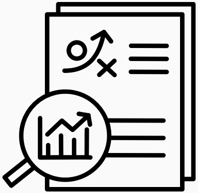
- We control the plain text
- We observe the cipher text
- We measure the energy consumption over many operations

Correlation Power Analysis



- We control the plain text
- We observe the cipher text
- We measure the energy consumption over many operations
- We guess the key

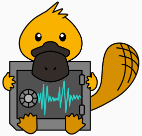
Correlation Power Analysis

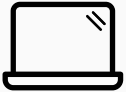


- We control the plain text
- We observe the cipher text
- We measure the energy consumption over many operations
- We guess the key

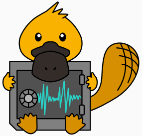
- With our model and all possible values, **where** is the **correlation** the highest?

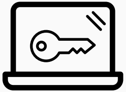
CPA Attack



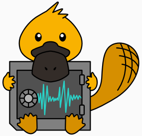


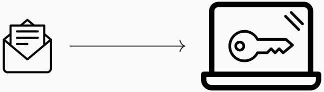
CPA Attack



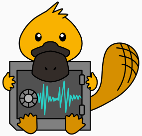


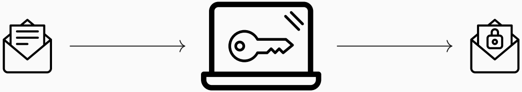
CPA Attack



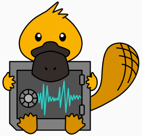


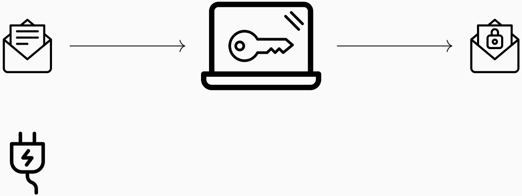
CPA Attack

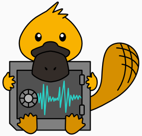


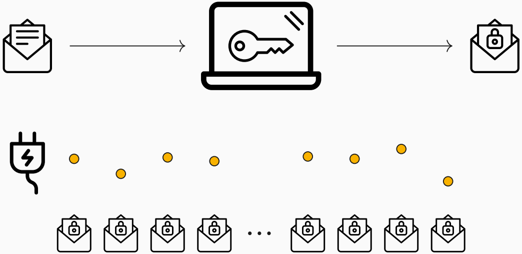


CPA Attack

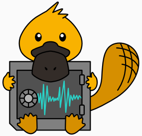


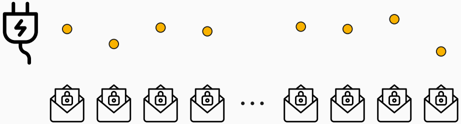




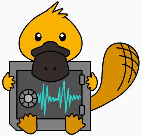


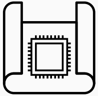
CPA Attack

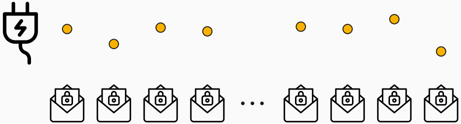




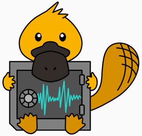
CPA Attack

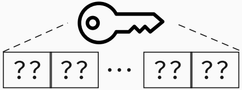


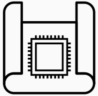


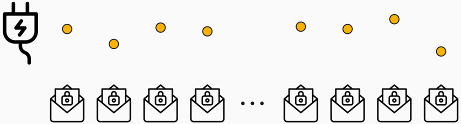


CPA Attack

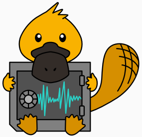


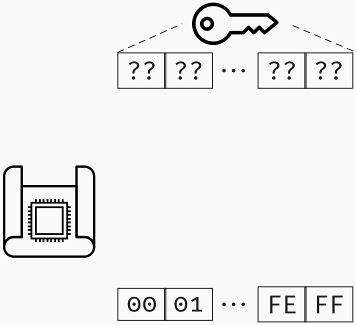


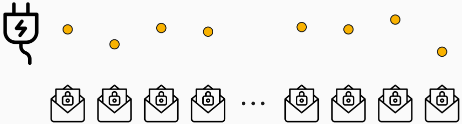




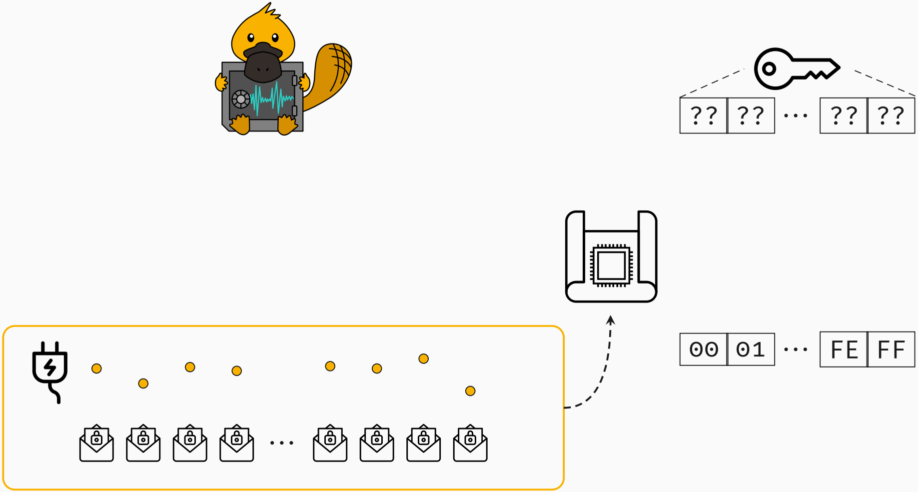
CPA Attack



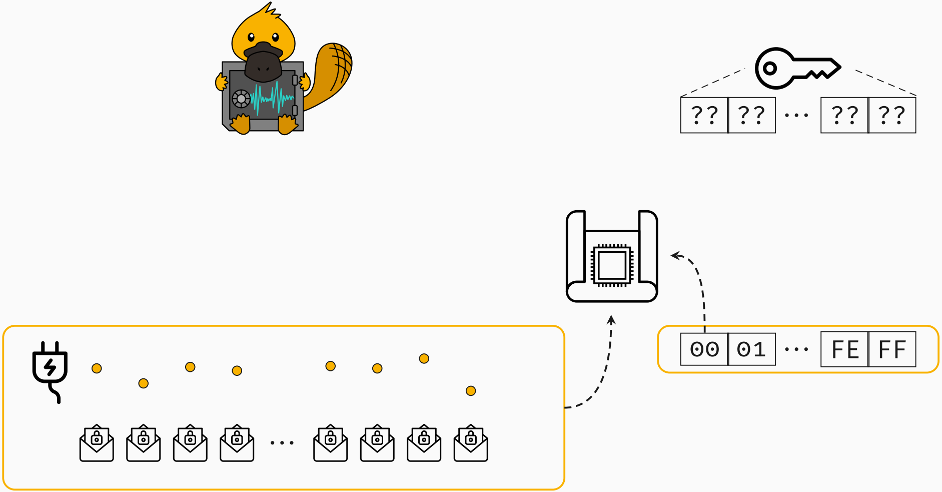




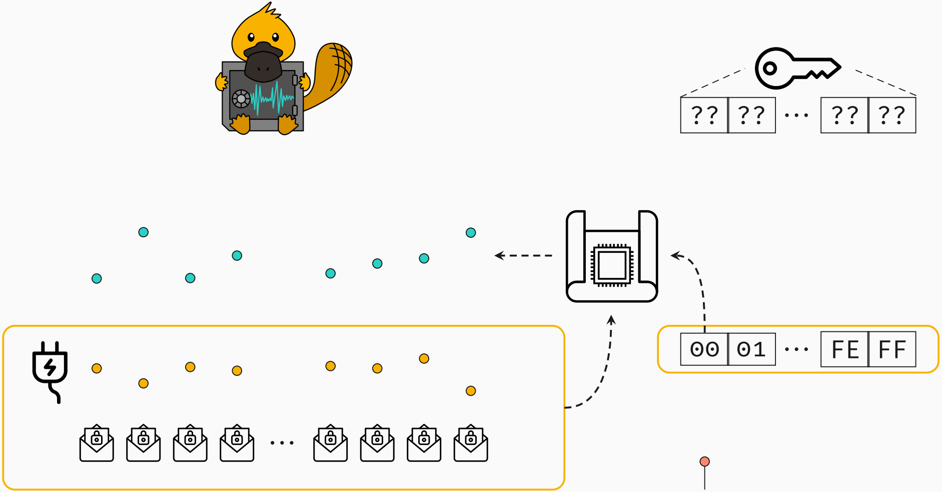
CPA Attack



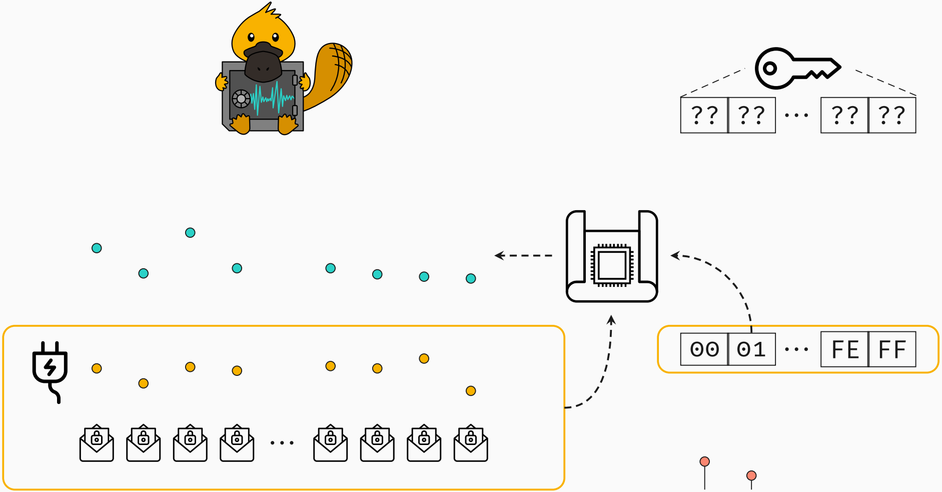
CPA Attack



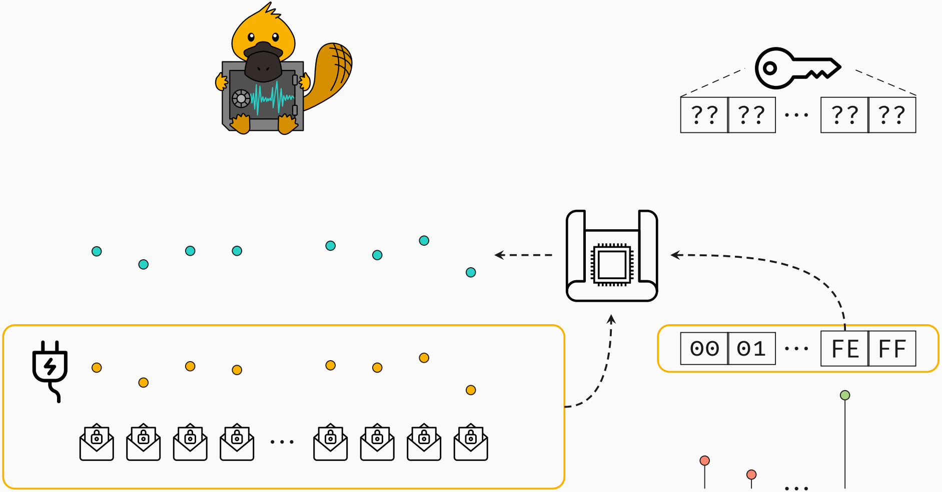
CPA Attack



CPA Attack

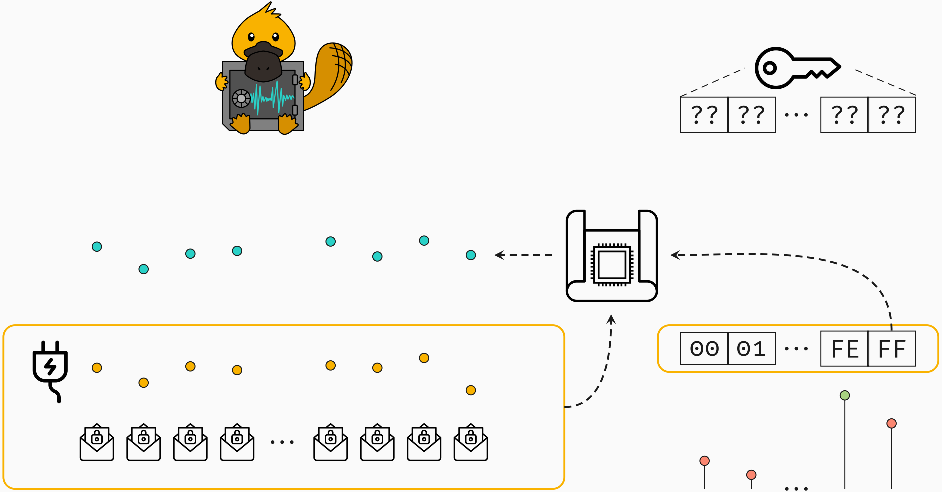


CPA Attack

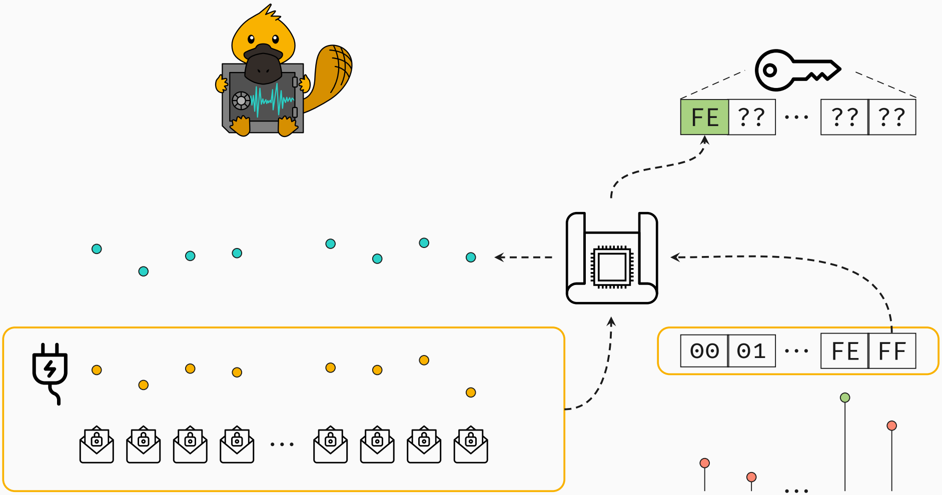


Mathias Oberhuber

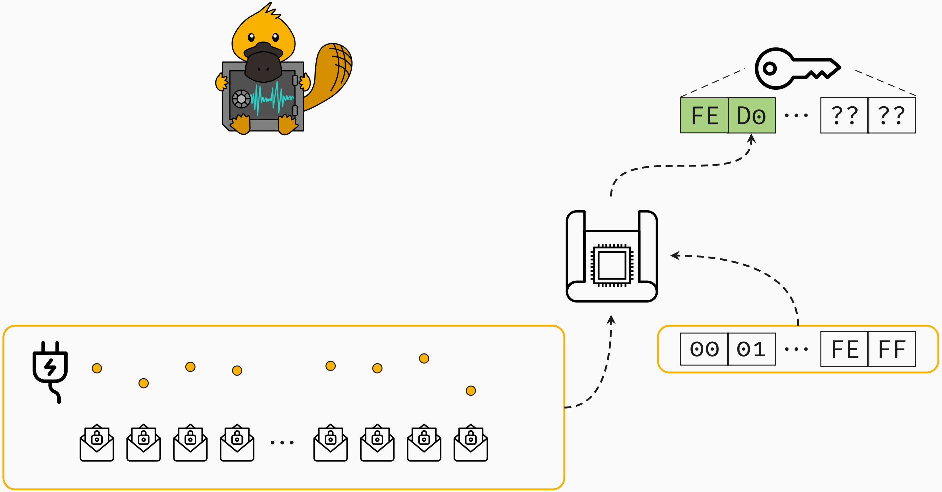
CPA Attack

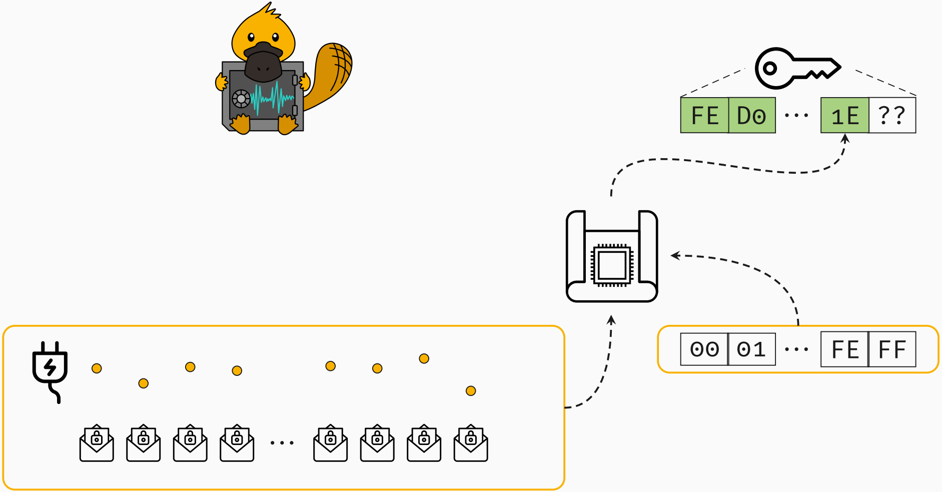


CPA Attack

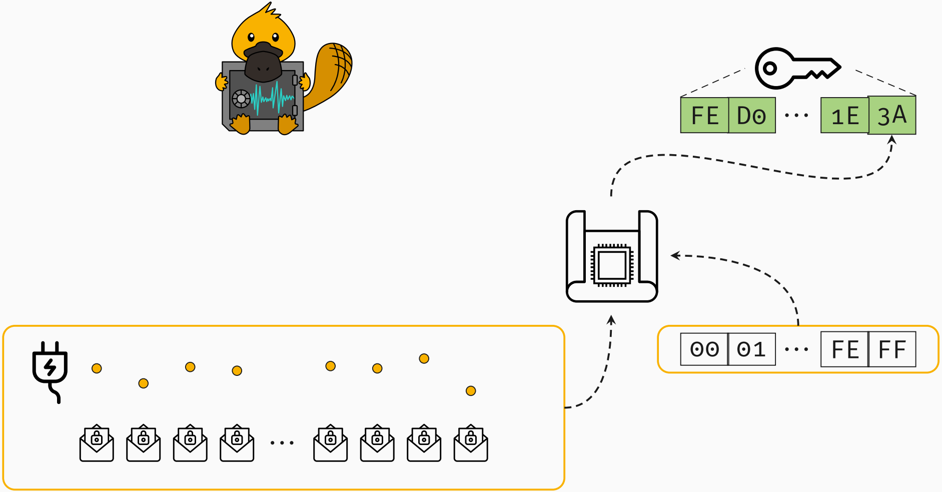


CPA Attack




FE
D0
...
1E
??
00
01
...
FE
FF

CPA Attack



Other CPU Vendors

- AMD affected as well

- AMD affected as well
- Never heard back after disclosure

Other CPU Vendors

- AMD affected as well
- Never heard back after disclosure
- Similar Linux patch as Intel

Countermeasures

Countermeasures

- Remove the unprivileged access to the RAPL MSRs

Countermeasures

- Remove the unprivileged access to the RAPL MSRs
- **1 Line Patch** for the Linux Kernel

Countermeasures

- Threat model of SGX allows a **compromised operating system**

- Threat model of SGX allows a **compromised operating system**
    - Operating system patch does not help

Countermeasures

- Threat model of SGX allows a **compromised operating system**
  - Operating system patch does not help
- **Microcode updates** are necessary

Countermeasures

- Threat model of SGX allows a **compromised operating system**
    - Operating system patch does not help
- **Microcode updates** are necessary
    - Fallback to a model of the energy consumption

Countermeasures

- Threat model of SGX allows a **compromised operating system**
  - Operating system patch does not help
- **Microcode updates** are necessary
  - Fallback to a model of the energy consumption
  - Does **not allow** to distinguish data/operands any more

Countermeasures

- Threat model of SGX allows a **compromised operating system**
  - Operating system patch does not help
- **Microcode updates** are necessary
  - Fallback to a model of the energy consumption
  - Does **not allow** to distinguish data/operands any more
  - **Constant-time implementations** are necessary

Takeaways

- Power side-channel attacks can be exploited **from software** on modern CPUs

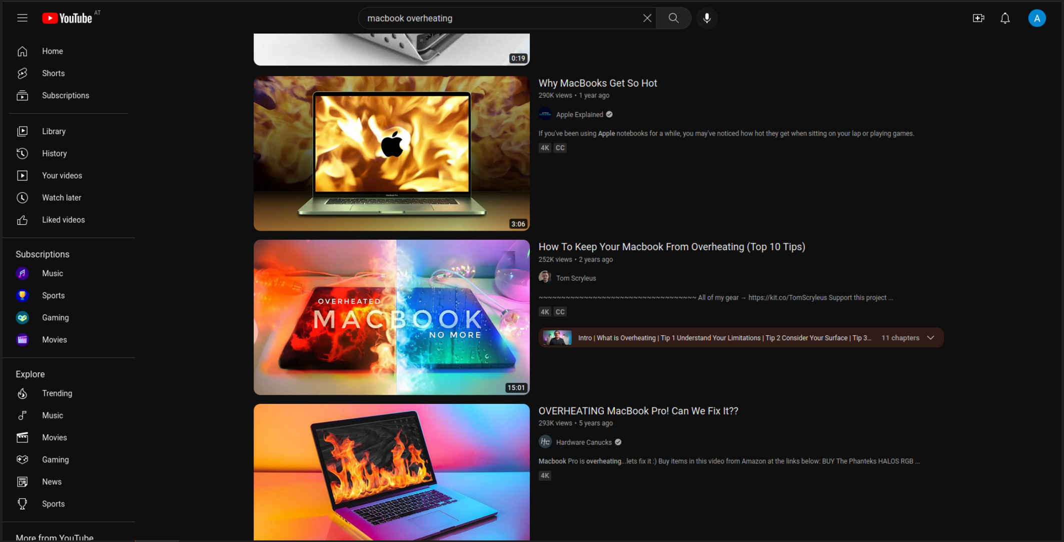
Takeaways

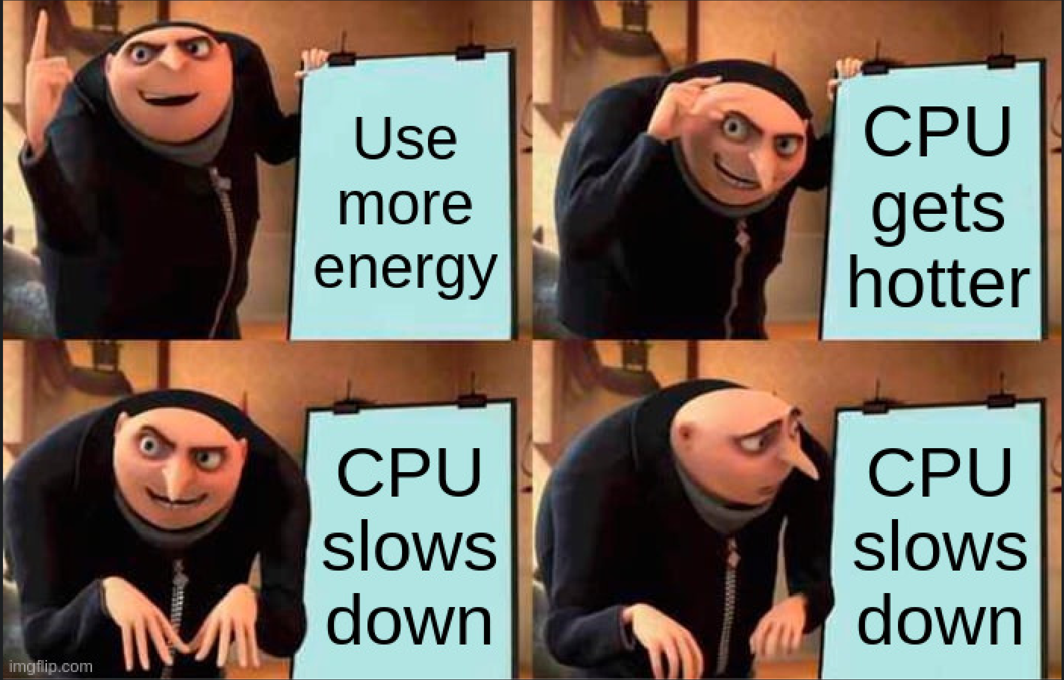
- Power side-channel attacks can be exploited **from software** on modern CPUs
- Threat model of Intel SGX requires more complex mitigations

Takeaways

- Power side-channel attacks can be exploited **from software** on modern CPUs
- Threat model of Intel SGX requires more complex mitigations

Remove Interface = The End?

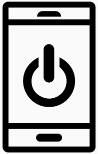

YouTube AT
macbook overheating
Home
Shorts
Subscriptions
Library
History
Your videos
Watch later
Liked videos
Subscriptions
Music
Sports
Gaming
Movies
Explore
Trending
Music
Movies
Gaming
News
Sports
More from YouTube
0:19
Why MacBooks Get So Hot
290K views • 1 year ago
Apple Explained
If you've been using Apple notebooks for a while, you may've noticed how hot they get when sitting on your lap or playing games.
4K CC
3:06
OVERHEATED MACBOOK NO MORE
15:01
How To Keep Your Macbook From Overheating (Top 10 Tips)
252K views • 2 years ago
Tom Scryleus
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ All of my gear → https://kit.co/TomScryleus Support this project ...
4K CC
Intro | What is Overheating | Tip 1 Understand Your Limitations | Tip 2 Consider Your Surface | Tip 3...
11 chapters
OVERHEATING MacBook Pro! Can We Fix It??
293K views • 5 years ago
Hardware Canucks
Macbook Pro is overheating...lets fix it :) Buy items in this video from Amazon at the links below: BUY The Phanteks HALOS RGB ...
4K
A
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



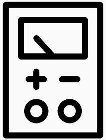
Use more energy
CPU gets hotter
CPU slows down
CPU slows down
imgflip.com

Remember?

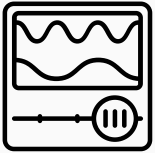
- CPU power management is complex
- In order to save power, you can …



Shut down resources



Reduce voltage



Reduce frequency

Hertzbleed

- The Hertzbleed attack from Wang et al. shows:
- If more **energy** is used

Hertzbleed

- The Hertzbleed attack from Wang et al. shows:
- If more **energy** is used
- The CPU gets **hotter**

Hertzbleed

- The Hertzbleed attack from Wang et al. shows:
- If more **energy** is used
- The CPU gets **hotter**
- Until the frequency is no longer sustainable

Hertzbleed

- The Hertzbleed attack from Wang et al. shows:
- If more **energy** is used
- The CPU gets **hotter**
- Until the frequency is no longer sustainable

→ The runtime of the executed code slows down

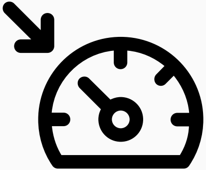
Hertzbleed

- The Hertzbleed attack from Wang et al. shows:
- If more **energy** is used
- The CPU gets **hotter**
- Until the frequency is no longer sustainable

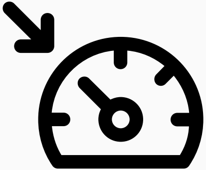
→ The runtime of the executed code slows down

- Measure with fixed clock, e.g., `rdtsc`

How to Throttle the CPU

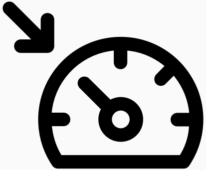


- RAPL provides energy **limits**



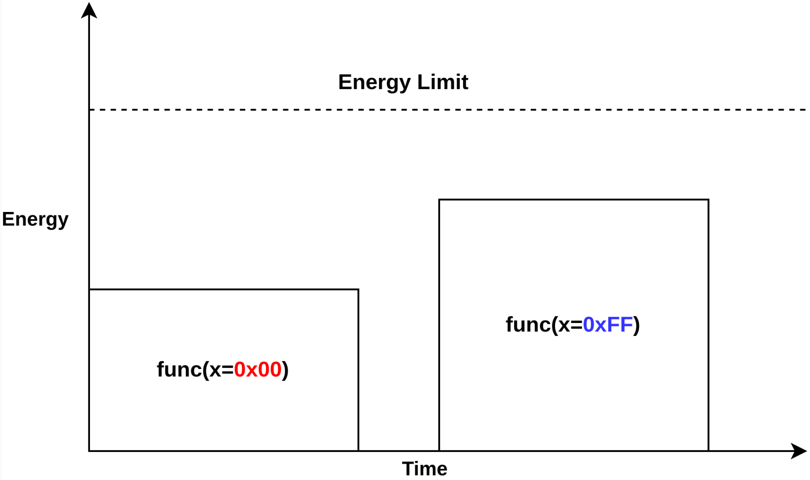
- RAPL provides energy **limits**
  - If exhausted CPU throttles the frequency
- Run **Stress** on the system

How to Throttle the CPU

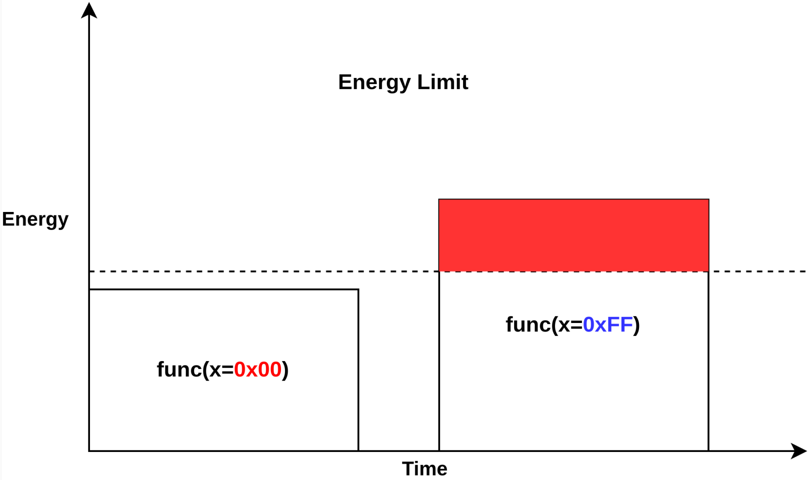


- RAPL provides energy **limits**
  - If exhausted CPU throttles the frequency
- Run **Stress** on the system
  - CPUs start throttling when using many threads

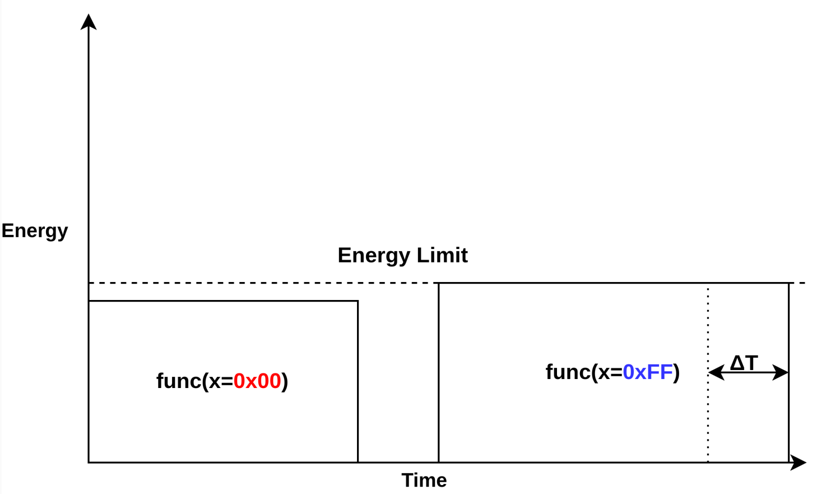
Converting Energy Differences



Convert Energy Differences



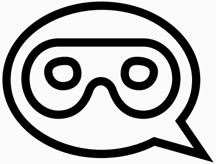
Convert Energy Differences



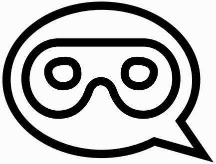

NYPD
POWER
TIME
imgflip.com

What can we do with this?

Covert Channel

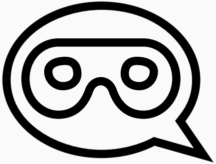


- Hidden communication channel

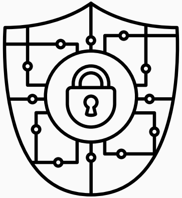


- Hidden communication channel
- No power interface required

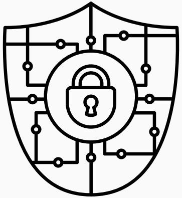
Covert Channel



- Hidden communication channel
- No power interface required
- Time/Frequency measurements proxy power interface

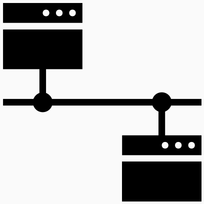


- AES Correlation Power Analysis
    - Measure execution time of AES encryptions



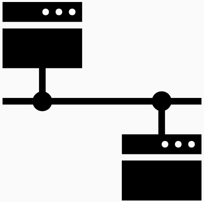
- AES Correlation Power Analysis
  - Measure execution time of AES encryptions
  - → Apply CPA technique to recover key

Remote Attack



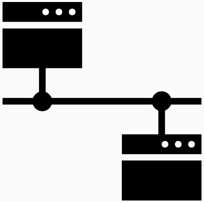
- Remote attacker requests service from server

Remote Attack



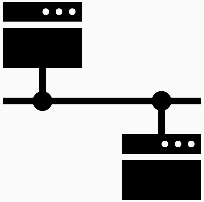
- Remote attacker requests service from server
    - Cryptographic operation, i.e. encryption, signature

Remote Attack



- Remote attacker requests service from server
  - Cryptographic operation, i.e. encryption, signature
- Server computes respose using secret

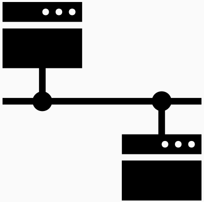
Remote Attack



- Remote attacker requests service from server
  - Cryptographic operation, i.e. encryption, signature
- Server computes respose using secret

→ **Hertzbleed-effect** influences response times

Remote Attack

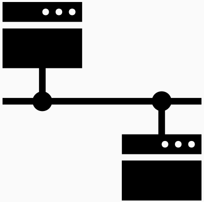


- Remote attacker requests service from server
  - Cryptographic operation, i.e. encryption, signature
- Server computes respose using secret

→ **Hertzbleed-effect** influences response times
  - **Calculations using secret** influences server CPU frequency

Mathias Oberhuber

Remote Attack

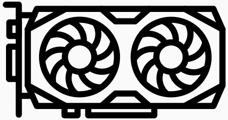


- Remote attacker requests service from server
  - Cryptographic operation, i.e. encryption, signature
- Server computes respose using secret

→ **Hertzbleed-effect** influences response times
  - **Calculations using secret** influences server CPU frequency

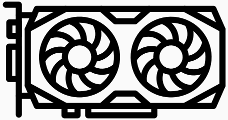
→ Attacker recovers secret using collected timings

GPU Throttling



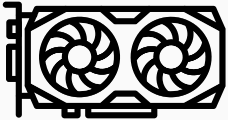
- **Integrated** GPUs share power limits with the CPU

GPU Throttling

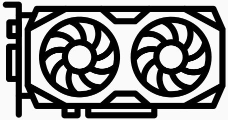


- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption

GPU Throttling

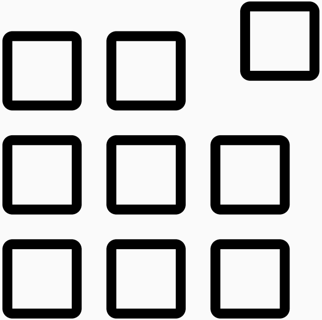


- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption
- **Dedicated** GPUs have power limits too



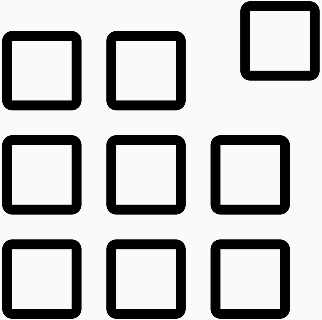
- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption
- **Dedicated** GPUs have power limits too
  → **Observable** by timing a GPU workload

Pixel Stealing



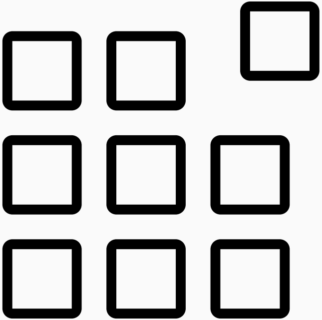
- What secrets are “*inside*” a GPU?

Pixel Stealing



- What secrets are *“inside”* a GPU?
  - GPU renders windows and screen

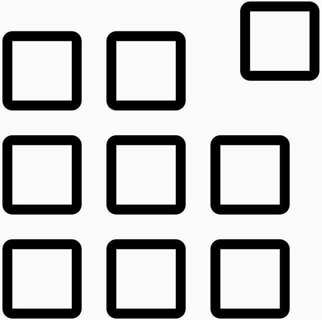
Pixel Stealing



- What secrets are *“inside”* a GPU?
    - GPU renders windows and screen

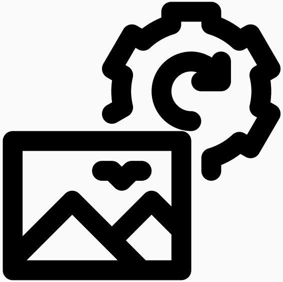
  → **Privacy** related information

Mathias Oberhuber



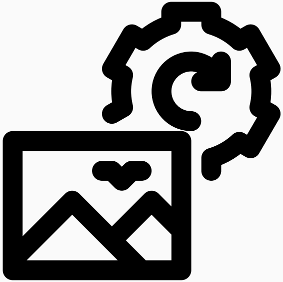
- What secrets are *“inside”* a GPU?
  - GPU renders windows and screen
  - → **Privacy** related information
- **Pixel** color represents the information

Pixel Stealing



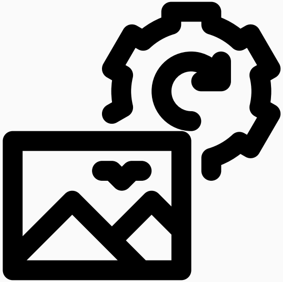
- **Post-processing** without revealing the pixels

Pixel Stealing



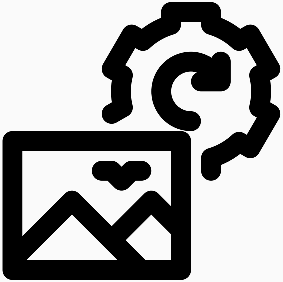
- **Post-processing** without revealing the pixels
- Pixel value is the **data operand**

Pixel Stealing



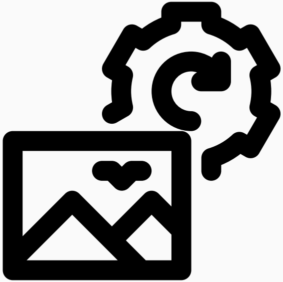
- **Post-processing** without revealing the pixels
- Pixel value is the **data operand**
- Distinguishable power consumption

Pixel Stealing



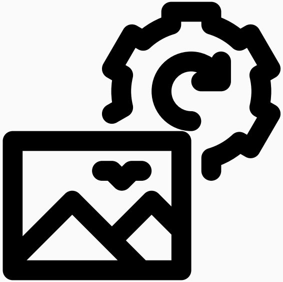
- **Post-processing** without revealing the pixels
- Pixel value is the **data operand**
- Distinguishable power consumption
  - **Bright** pixel → less power

Pixel Stealing



- **Post-processing** without revealing the pixels
- Pixel value is the **data operand**
- Distinguishable power consumption
  - **Bright** pixel → less power
  - **Dark** pixel → more power

Pixel Stealing



- **Post-processing** without revealing the pixels
- Pixel value is the **data operand**
- Distinguishable power consumption
  - **Bright** pixel → less power
  - **Dark** pixel → more power

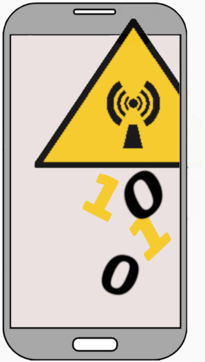
→ Measure timing and infer pixel value

The End?

Are there other exploitable **power-related** signals?

About

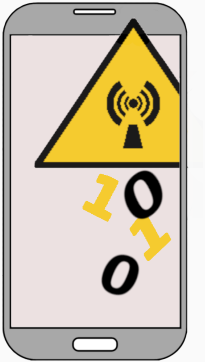
Android **power-related** side channel



About

Android **power-related** side channel

- Android sensor interface as a proxy for power measurements purely from software



About

Android **power-related** side channel

- Android sensor interface as a proxy for power measurements purely from software
- Systematic analysis of 9 Android smartphones:
  - Recovering leakage properties: Integration interval, rotation-dependent leakage

About

Android **power-related** side channel

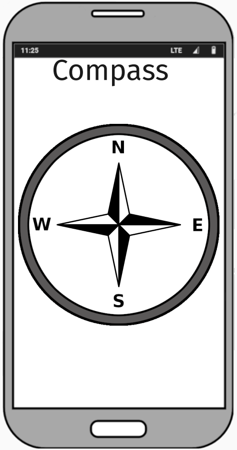
- Android sensor interface as a proxy for power measurements purely from software
- Systematic analysis of 9 Android smartphones:
  - Recovering leakage properties: Integration interval, rotation-dependent leakage
- Local attack:
  - Malicious app leaking processed AES key bytes

About

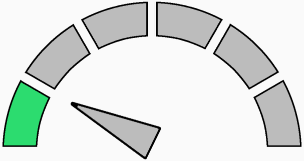
Android **power-related** side channel

- Android sensor interface as a proxy for power measurements purely from software
- Systematic analysis of 9 Android smartphones:
  - Recovering leakage properties: Integration interval, rotation-dependent leakage
- Local attack:
  - Malicious app leaking processed AES key bytes
- Remote web-based JavaScript attack:
  - JavaScript sensor-based pixel-stealing attack leaking cross-origin pixels up to 5 s/pixel

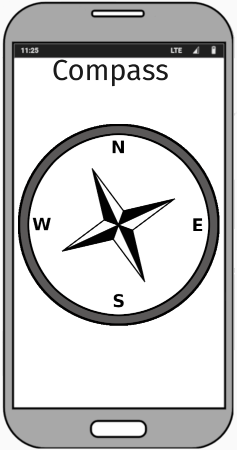
Motivation



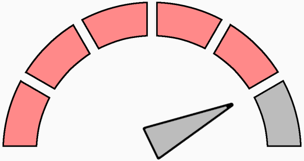
CPU utilization



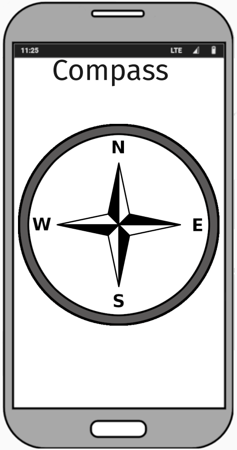
Motivation



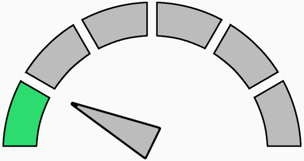
CPU utilization



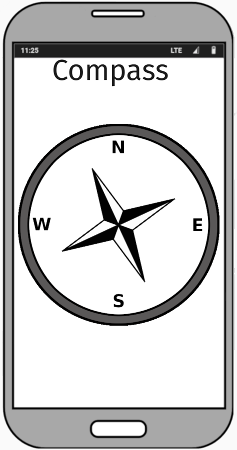
Motivation



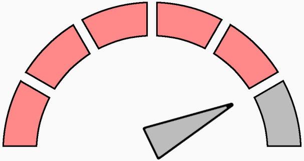
CPU utilization



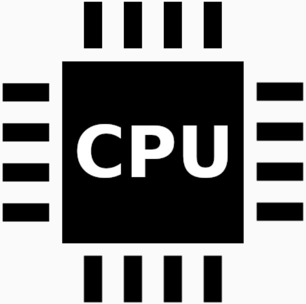
Motivation

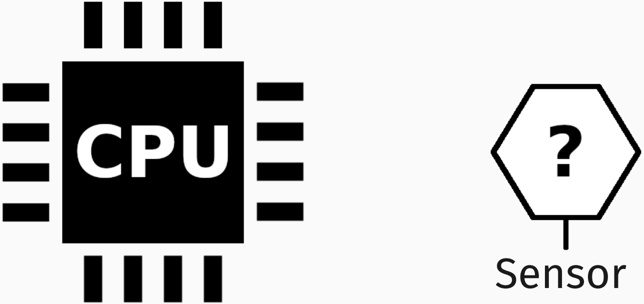


CPU utilization

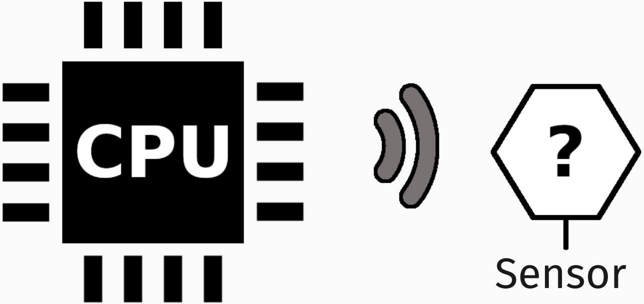


Motivation

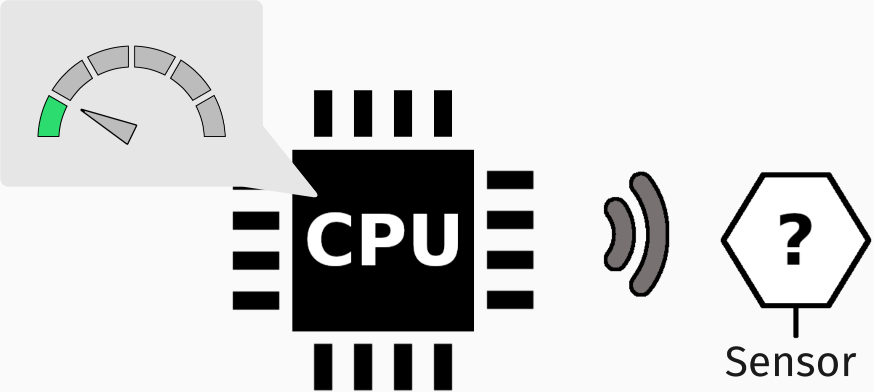



CPU
?
Sensor

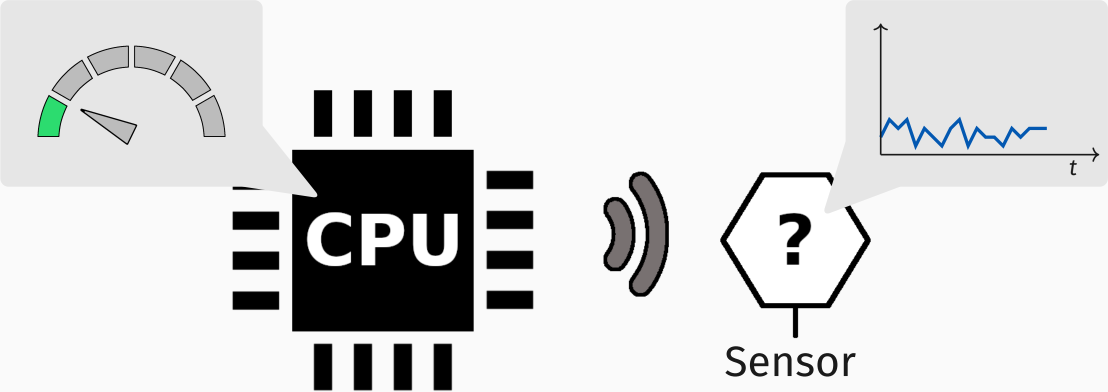
Motivation



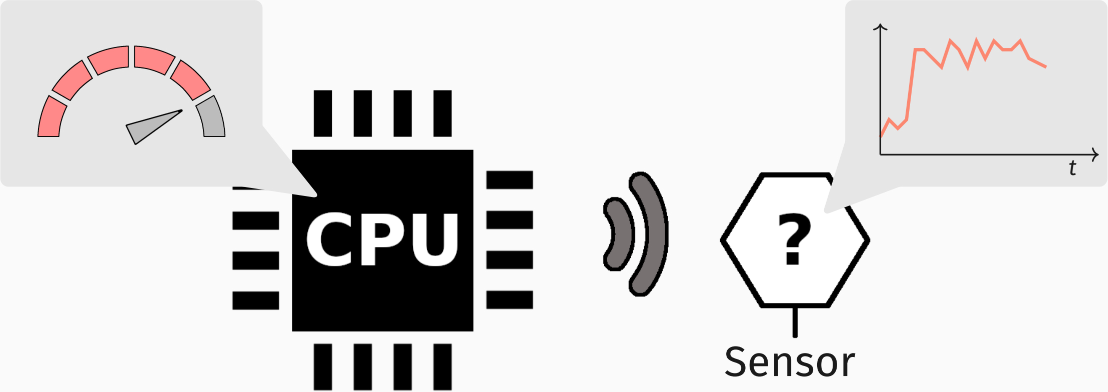
Motivation



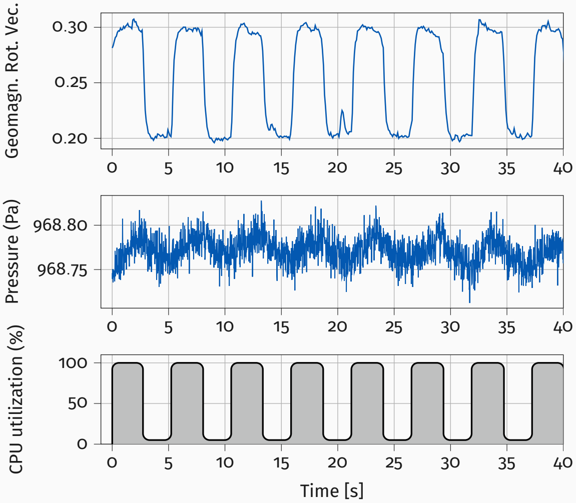
Motivation



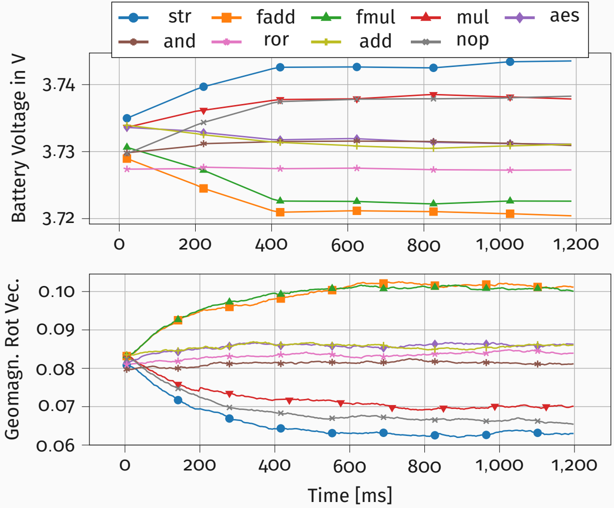
Motivation



Systematic Evaluation: Varying CPU Utilization



Systematic Sensor Analysis: Executed Instructions



Systematic Sensor Analysis: Varying Data Operands

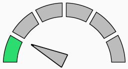
$$a \oplus b = c$$

Systematic Sensor Analysis: Varying Data Operands

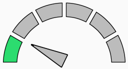
$$a \oplus b = c$$

$$a_0 \oplus b_0 = 00_2$$

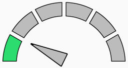
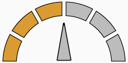
Systematic Sensor Analysis: Varying Data Operands

$a \oplus b = c$	Power
$a_0 \oplus b_0 = 00_2$	

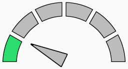
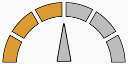
Systematic Sensor Analysis: Varying Data Operands

a	⊕	b	=	c	Power
a_0	$\oplus$	b_0	$=$	00_2	
a_2	$\oplus$	b_2	$=$	01_2	

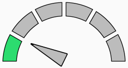
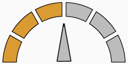
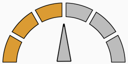
Systematic Sensor Analysis: Varying Data Operands

a	$\oplus$	b	=	c	Power
a_0	$\oplus$	b_0	=	00_2	
a_2	$\oplus$	b_2	=	01_2	

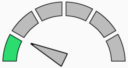
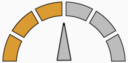
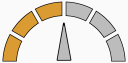
Systematic Sensor Analysis: Varying Data Operands

a	$\oplus$	b	=	c	Power
a_0	$\oplus$	b_0	=	00_2	
a_2	$\oplus$	b_2	=	01_2	
a_2	$\oplus$	b_2	=	10_2	

Mathias Oberhuber

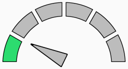
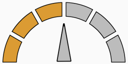
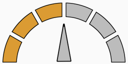
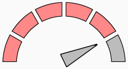
a	⊕	b	=	c	Power
a_0	$\oplus$	b_0	$=$	00_2	
a_2	$\oplus$	b_2	$=$	01_2	
a_2	$\oplus$	b_2	$=$	10_2	

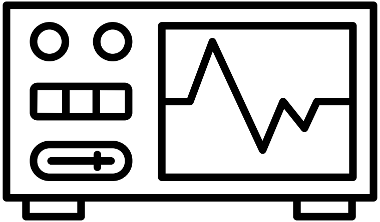
Systematic Sensor Analysis: Varying Data Operands

a	⊕	b	=	c	Power
a_0	$\oplus$	b_0	$=$	00_2	
a_2	$\oplus$	b_2	$=$	01_2	
a_2	$\oplus$	b_2	$=$	10_2	
a_4	$\oplus$	b_4	$=$	11_2	

Mathias Oberhuber

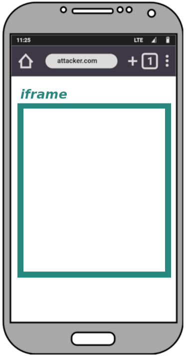
Systematic Sensor Analysis: Varying Data Operands

a	$\oplus$	b	=	c	Power
a_0	$\oplus$	b_0	=	00_2	
a_2	$\oplus$	b_2	=	01_2	
a_2	$\oplus$	b_2	=	10_2	
a_4	$\oplus$	b_4	=	11_2	



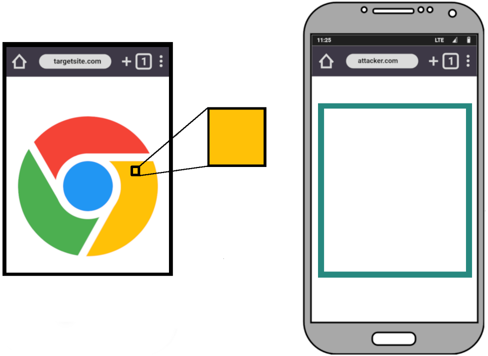
What can we do with this?

JavaScript Pixel-Stealing Attack

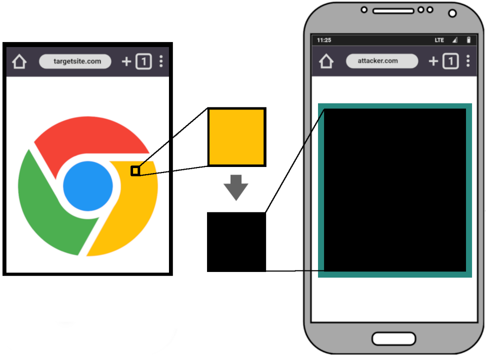


JavaScript Pixel-Stealing Attack

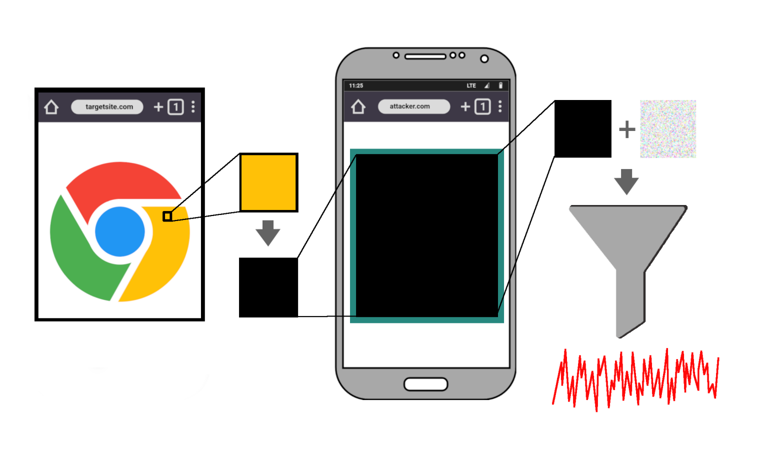
JavaScript Pixel-Stealing Attack

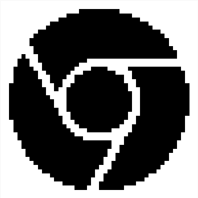


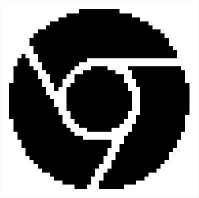
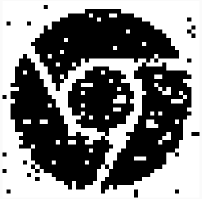
JavaScript Pixel-Stealing Attack



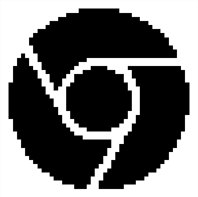
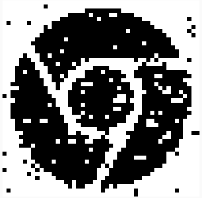
JavaScript Pixel-Stealing Attack



			
Image: **Time/Pixel (s):** **Accuracy (%):**	Original		

			
Image:	Original	Magnetometer	
Time/Pixel (s):		5	
Accuracy (%):		90.2	

JavaScript Pixel Stealing: Evaluation

			
Image:	Original	Magnetometer	Abs. Orientation
Time/Pixel (s):		5	10
Accuracy (%):		90.2	70

AES Attack Case Study

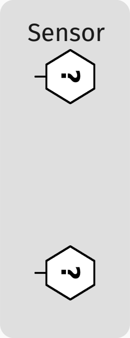
round	AES
1	aese aesmc

AES Attack Case Study

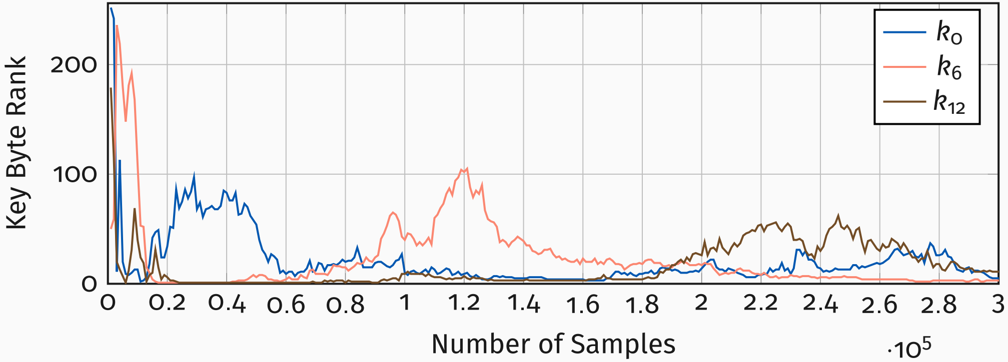
round	AES
1	aese aesmc
2	aese aesmc
	⋮

AES Attack Case Study

round	AES
1	aese aesmc
2	aese aesmc
	⋮



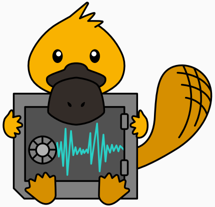
AES Correlation Power Analysis



How can we **transform** power side channels towards a broader scope?

Motivation

Software-based Power Side Channels



Software-based Power Side Channels

- **Specific** targets: Algorithms

Motivation

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

Motivation

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

Transient Execution Attacks

Motivation

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

Transient Execution Attacks

- **Generic** targets: CPU components

Motivation

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

Transient Execution Attacks

- **Generic** targets: CPU components
- Leak arbitrary data

Software-based Power Side Channels

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

Transient Execution Attacks

- **Generic** targets: CPU components
- Leak arbitrary data
- **Agnostic** to side channels

Motivation

Collide+Power

Software-based Power Side C

- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channel

ecution Attacks

- targets: CPU components
- Leak arbitrary data
- **Agnostic** to side channels

Collide+Power

- **Collide+Power** exploits leakage between:

Collide+Power - Overview

- **Collide+Power** exploits leakage between:
  - **Guess $\mathcal{G}$:** Attacker-controlled data

- **Collide+Power** exploits leakage between:
  - **Guess $\mathcal{G}$:** Attacker-controlled data
  - **Value $\mathcal{V}$:** Victim secret data

- **Collide+Power** exploits leakage between:
  - **Guess $\mathcal{G}$:** Attacker-controlled data
  - **Value $\mathcal{V}$:** Victim secret data

💡 Hamming distance: $\text{hd}(\mathcal{G}, \mathcal{V})$

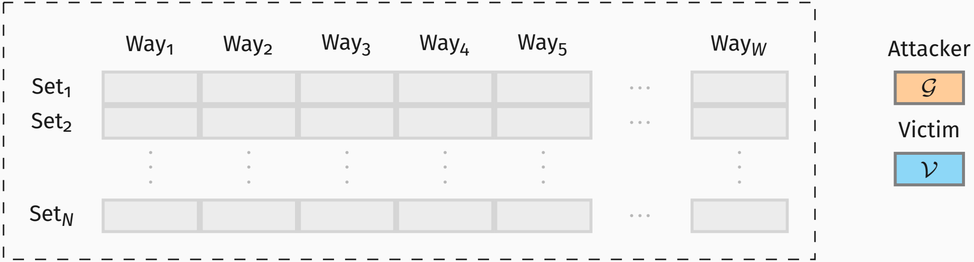
Collide+Power - Overview

- **Collide+Power** exploits leakage between:
  - **Guess $\mathcal{G}$:** Attacker-controlled data
  - **Value $\mathcal{V}$:** Victim secret data

Hamming distance: $\text{hd}(\mathcal{G}, \mathcal{V})$

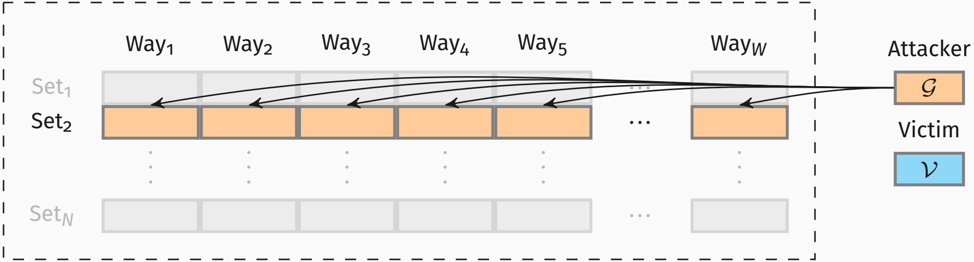
→ How to exploit this limited information?

Collide+Power - Memory Subsystem

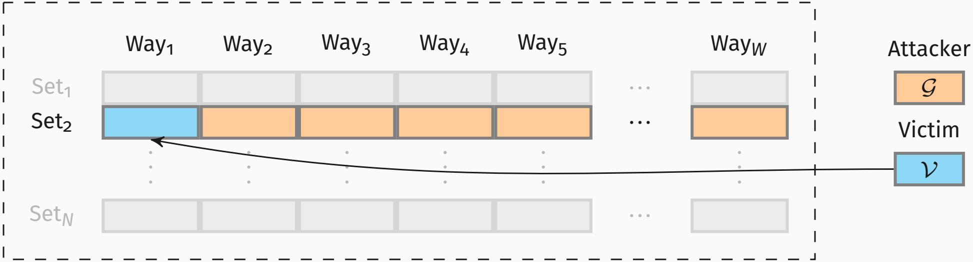


Mathias Oberhuber

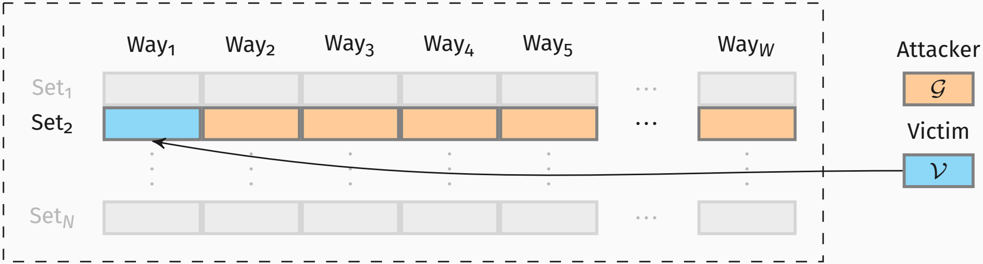
Collide+Power - Memory Subsystem

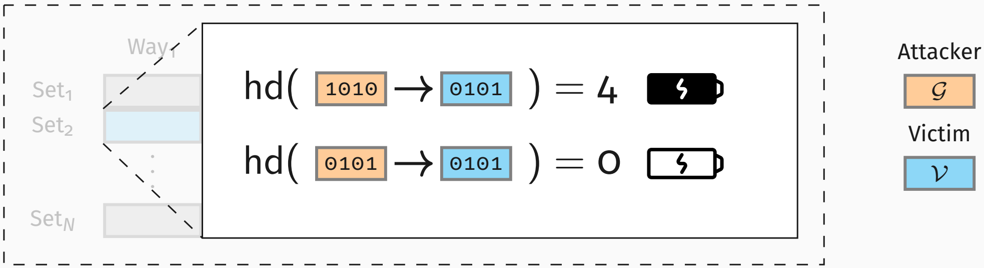


Collide+Power - Memory Subsystem



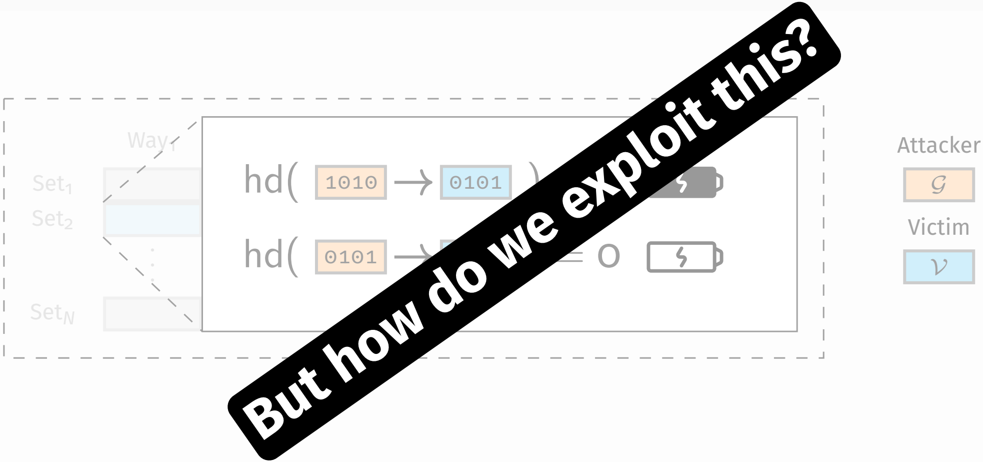
Collide+Power - Memory Subsystem





Way₁
Set₁
Set₂
Set_N
hd(1010 → 0101) = 4
hd(0101 → 0101) = 0
Attacker
Victim

Collide+Power - Memory Subsystem



Collide+Power - Intuition

$$\mathcal{P}(\mathcal{G}, \mathcal{V}) \approx \ldots$$

Collide+Power - Intuition

$$\mathcal{P}(\mathcal{G}, \mathcal{V}) \approx \text{hd}(\mathcal{G}, \mathcal{V})$$

$$\mathcal{P}(\mathcal{G}, \mathcal{V}) \approx \mathrm{hd}(\mathcal{G}, \mathcal{V})$$

Collide+Power - Intuition

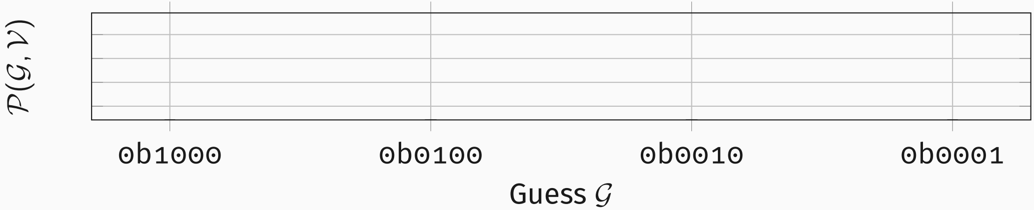
$$\underbrace{\mathcal{P}(\mathcal{G}, \mathcal{V})}_{\text{model}} \approx \text{hd}(\mathcal{G}, \mathcal{V})$$

Collide+Power - Intuition

$$\underbrace{\mathcal{P}(\mathcal{G}, \mathcal{V})}_{\text{model}} \approx \underbrace{\text{hd}(\mathcal{G}, \mathcal{V})}_{\text{signal}}$$

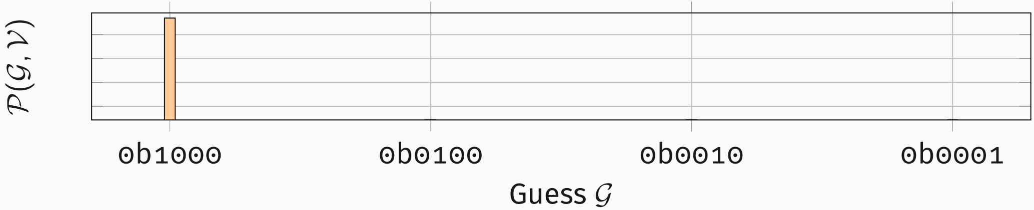
Collide+Power - Example

$$\mathcal{P}(\mathcal{G}, 0101_2) \approx \text{hd}(\mathcal{G}, 0101_2)$$



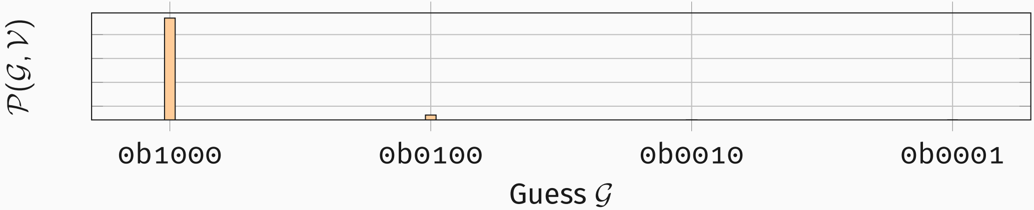
Collide+Power - Example

$$\mathcal{P}(1000_2, 0101_2) \approx \text{hd}(\mathbf{1}000_2, \mathbf{0}101_2) = 3$$



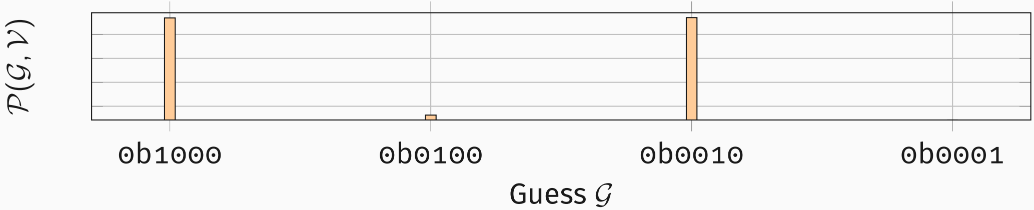
Collide+Power - Example

$$\mathcal{P}(0100_2, 0101_2) \approx \mathrm{hd}(0100_2, 0101_2) = 1$$



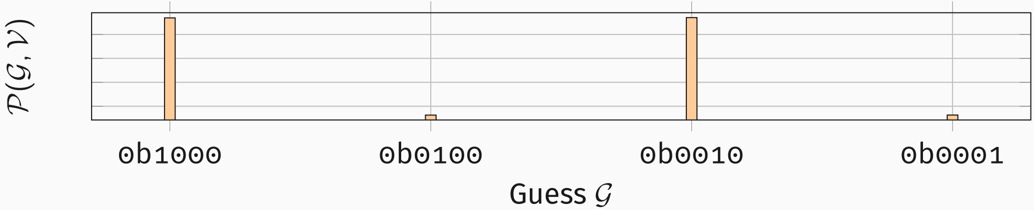
Collide+Power - Example

$$\mathcal{P}(0010_2, 0101_2) \approx \mathrm{hd}(0010_2, 0101_2) = 3$$



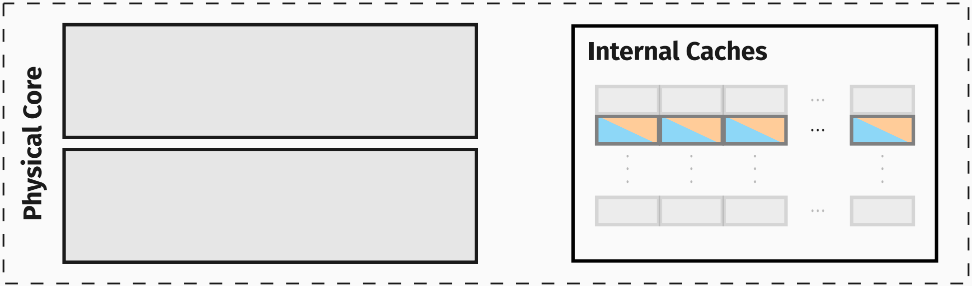
Collide+Power - Example

$$\mathcal{P}(0001_2, 0101_2) \approx \text{hd}(0001_2, 0101_2) = 1$$

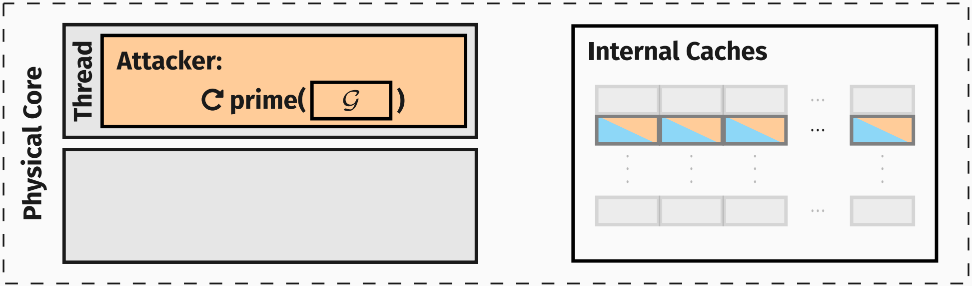


Generic Attacks

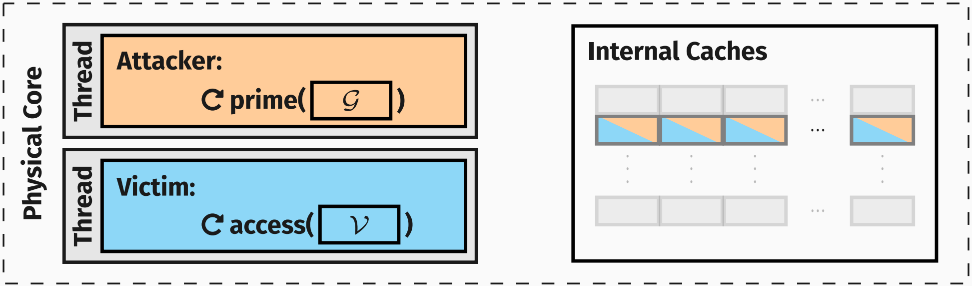
MDS-style Attack



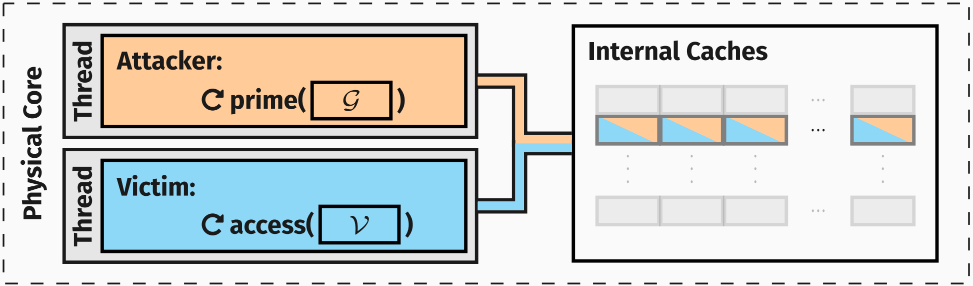
MDS-style Attack



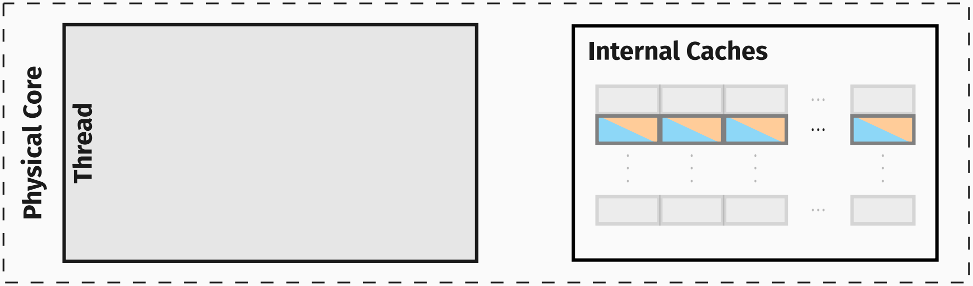
MDS-style Attack



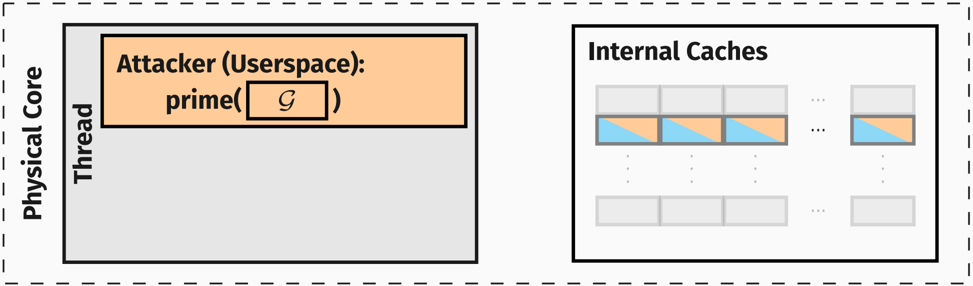
MDS-style Attack



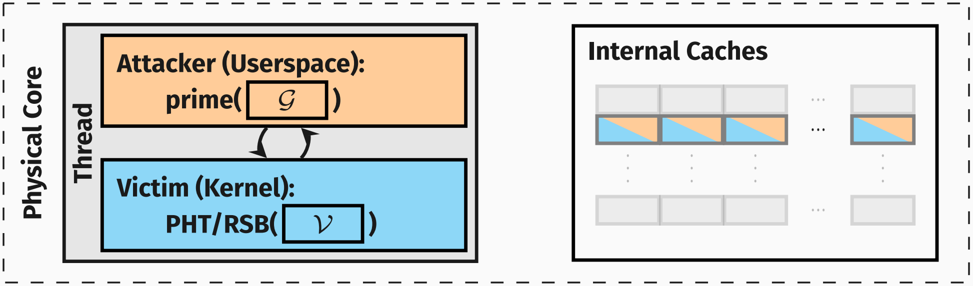
Meltdown-style Attack



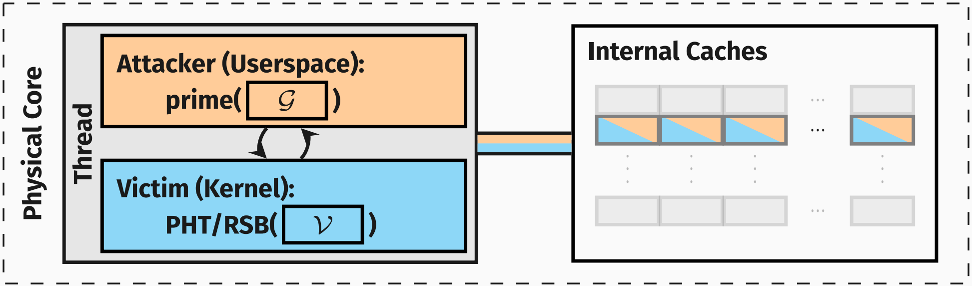
Meltdown-style Attack



Meltdown-style Attack



Meltdown-style Attack



This must be slow?

NO!

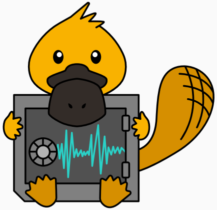
It is EXTREMELY slow![1]

[1] With the current state-of-the-art.

- **MDS-style:**
  4.82 bit/h

- **MDS-style:**
  4.82 bit/h
- **Meltdown-style (RSB):**
  0.84 bit/h

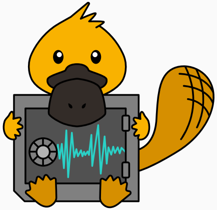
Software-based Power Side Channels



- **MDS-style:**
  4.82 bit/h
- **Meltdown-style (RSB):**
  0.84 bit/h

- **MDS-style:**
  0.065 to 0.68 bit/h

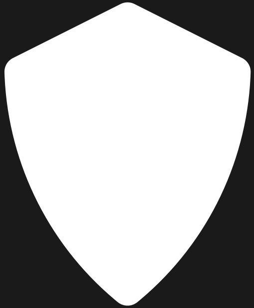
Software-based Power Side Channels



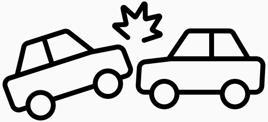
- **MDS-style:**
  4.82 bit/h
- **Meltdown-style (RSB):**
  0.84 bit/h

- **MDS-style:**
  0.065 to 0.68 bit/h
- **Meltdown-style estimate (PHT):**
  99.95 days/bit to 2.86 years/bit

Mitigations

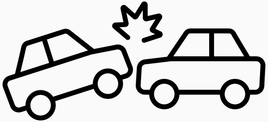


Mitigations



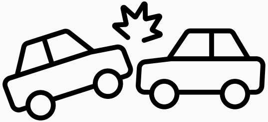
- **Preventing data collisions:**

Mitigations



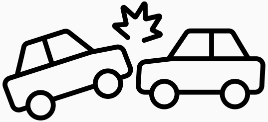
- **Preventing data collisions:**
  - **Redesign** of the complete shared data path

Mitigations



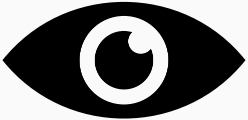
- **Preventing data collisions:**
  - **Redesign** of the complete shared data path
  - **Costly** to deploy

Mitigations

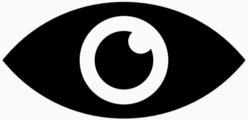


- **Preventing data collisions:**
  - **Redesign** of the complete shared data path
  - **Costly** to deploy
  - **Missed** components re-enable Collide+Power

Mitigations

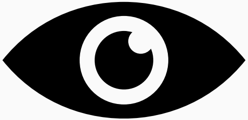


- **Preventing observable power consumption:**
  - **Restricting** all direct power interfaces



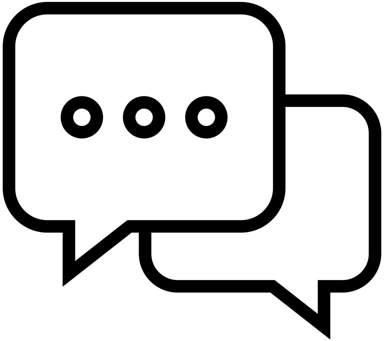
- **Preventing observable power consumption:**
  - **Restricting** all direct power interfaces
- **Mitigating** Hertzbleed is challenging
  - Thermal and power management is required

Mitigations



- **Preventing observable power consumption:**
  - **Restricting** all direct power interfaces
- **Mitigating** Hertzbleed is challenging
  - Thermal and power management is required

→ **Collide+Power** is slow but unmitigated on modern CPUs!



Questions?