black hat®
EUROPE 2023

Collide+Power

The Evolution of Software-based Power Side-Channels Attacks

Andreas Kogler
Graz University of Technology

6th December 2023

Whoami

- **Andreas Kogler**

Whoami

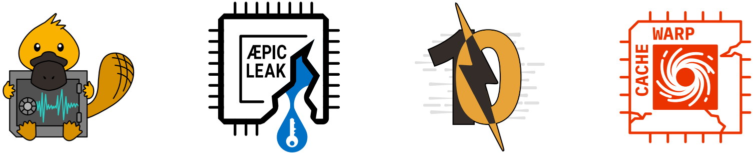
- **Andreas Kogler**
- PhD-Candidate - Graz University of Technology

Whoami

- **Andreas Kogler**
- PhD-Candidate - Graz University of Technology
  - Software-based power side channels
  - Software-based fault attacks
  - Trusted execution environments

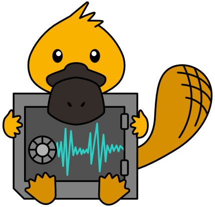
Whoami

- **Andreas Kogler**
- PhD-Candidate - Graz University of Technology
  - Software-based power side channels
  - Software-based fault attacks
  - Trusted execution environments



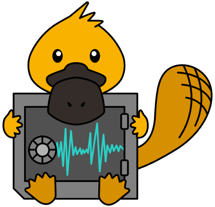
Motivation

Motivation



Software-based Power Side Channels

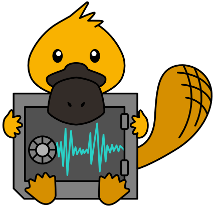
Motivation



Software-based Power Side Channels

- **Specific** targets: Algorithms

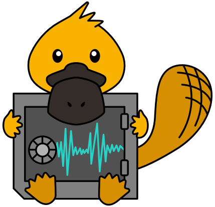
Motivation



Software-based Power Side Channels

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

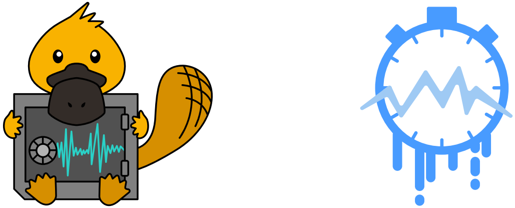
Motivation



Software-based Power Side Channels

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

Motivation

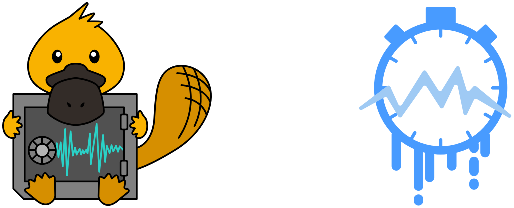


Software-based Power Side Channels

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

Transient Execution Attacks

Motivation



Software-based Power Side Channels

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

Transient Execution Attacks

- **Generic** targets: CPU components

Motivation

Software-based Power Side Channels

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

Transient Execution Attacks

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

Motivation

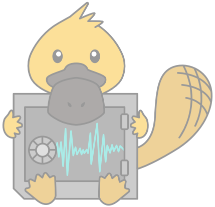
Software-based Power Side Channels

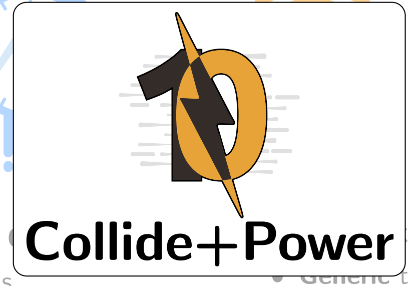
- **Specific** targets: Algorithms
- Leak edge cases
- **Limited** to a side channels

Transient Execution Attacks

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

Motivation





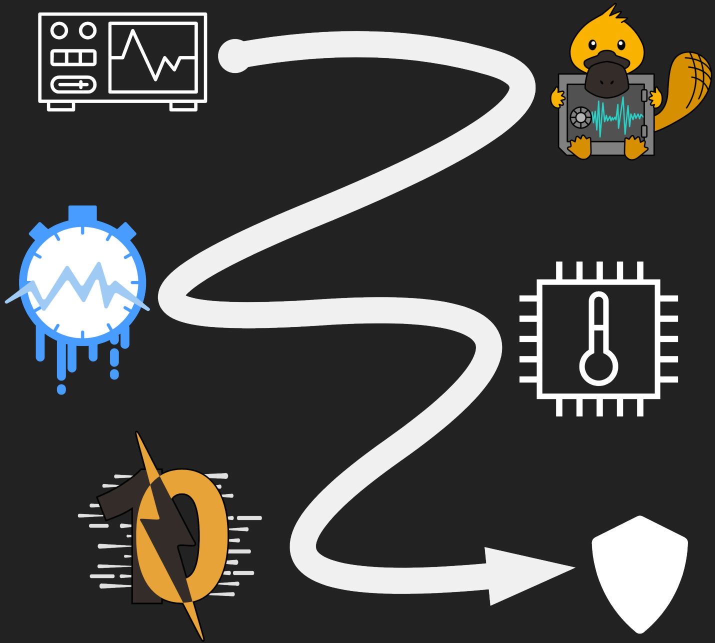
Software-based Power Side C

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

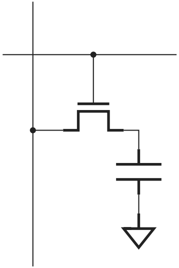
ecution Attacks

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

Can we build a generic software-based power side-channel attack independent of the targeted application?

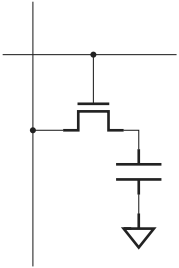

10

Power Leakage - Source



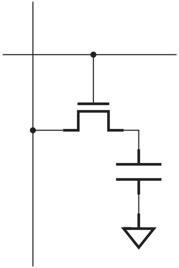
- **C**omplementary **M**etal **O**xide **S**emiconductor

Power Leakage - Source



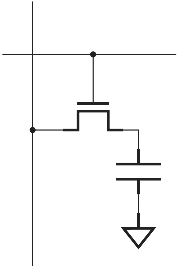
- **C**omplementary **M**etal **O**xide **S**emiconductor
- Low power consumption

Power Leakage - Source



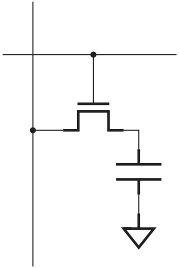
- **C**omplementary **M**etal **O**xide **S**emiconductor
- Low power consumption
- Depends on:

Power Leakage - Source



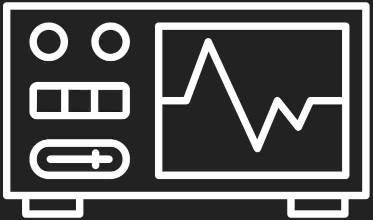
- **C**omplementary **M**etal **O**xide **S**emiconductor
- Low power consumption
- Depends on:
  - **Instruction** that is executed

Power Leakage - Source

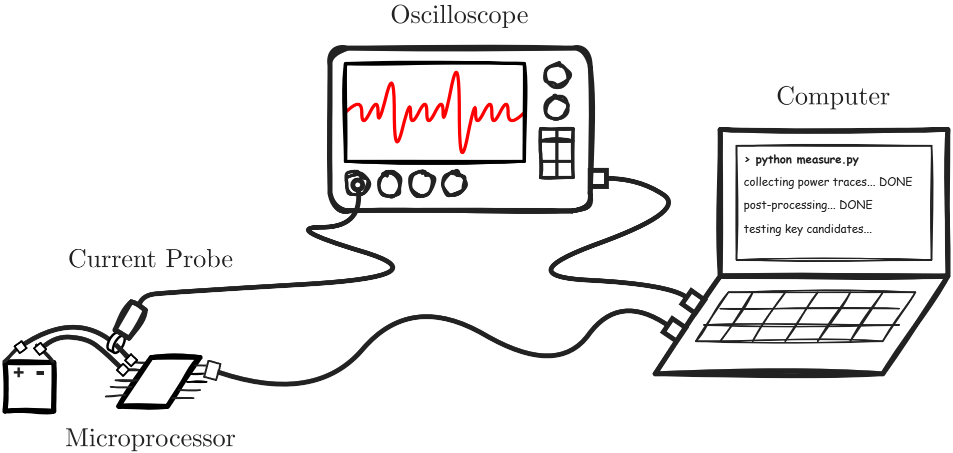


- **C**omplementary **M**etal **O**xide **S**emiconductor
- Low power consumption
- Depends on:
  - **Instruction** that is executed
  - **Data** that is being processed

Traditional Power Side Channels



Power Side Channel - Setup



Oscilloscope

Computer

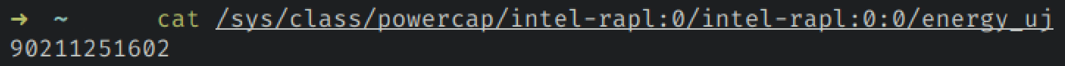
```
> python measure.py
collecting power traces... DONE
post-processing... DONE
testing key candidates...
```

Current Probe

Microprocessor

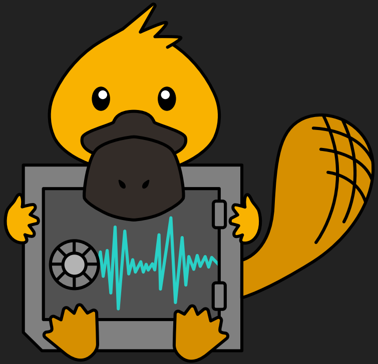
How can we measure the power consumption of a modern CPU?

How would we ever do this remotely?



```
➜  ~      cat /sys/class/powercap/intel-rapl:0/intel-rapl:0:0/energy_uj
90211251602
```

PLATYPUS[1]



[1] Moritz Lipp, Andreas Kogler, David Oswald, Michael Schwarz, Catherine Easdon, Claudio Canella, and Daniel Gruss. PLATYPUS: Software-based Power Side-Channel Attacks on x86. In: S&P. 2021.

Running Average Power Limit (RAPL)

Unprivileged power meter

Running Average Power Limit (RAPL)

Unprivileged power meter

No physical access

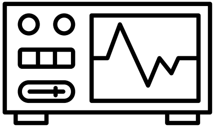
Running Average Power Limit (RAPL)

Unprivileged power meter

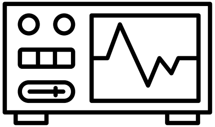
No physical access

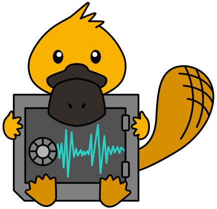
Low refresh rate

External Measurement Equipment vs RAPL

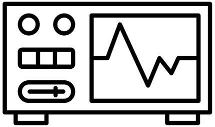


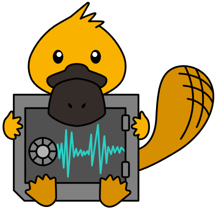
External Measurement Equipment vs RAPL





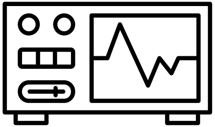
External Measurement Equipment vs RAPL

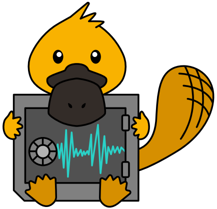




- **Full** Control

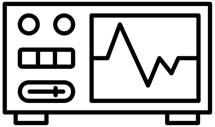
External Measurement Equipment vs RAPL

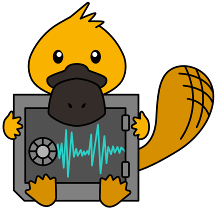




- **Full** Control
- **High** timing resolution

External Measurement Equipment vs RAPL

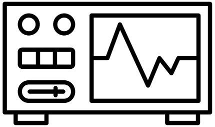




- **Full** Control
- **High** timing resolution

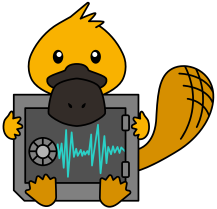
→ Multiple samples per instruction

External Measurement Equipment vs RAPL



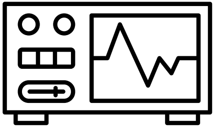
- **Full** Control
- **High** timing resolution

→ Multiple samples per instruction



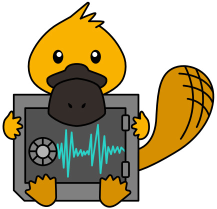
- **No** control, just a register

External Measurement Equipment vs RAPL



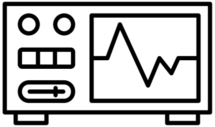
- **Full** Control
- **High** timing resolution

→ Multiple samples per instruction



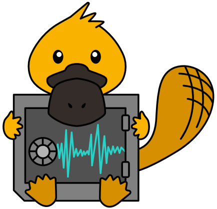
- **No** control, just a register
- **Low** timing resolution

External Measurement Equipment vs RAPL



- **Full** Control
- **High** timing resolution

→ Multiple samples per instruction

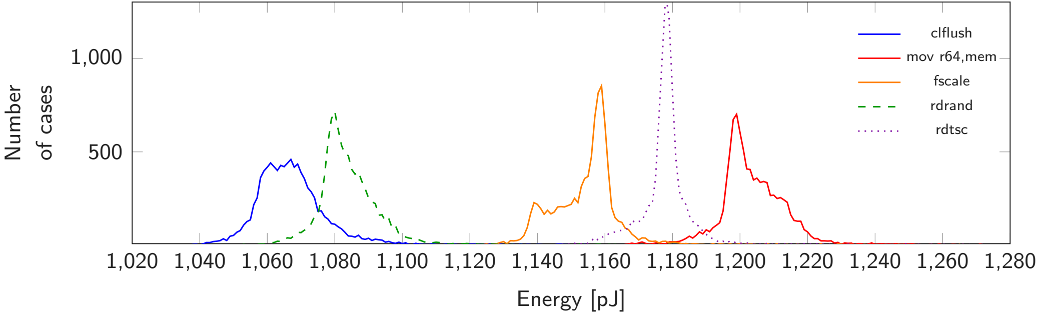


- **No** control, just a register
- **Low** timing resolution

→ Single sample per multiple instructions

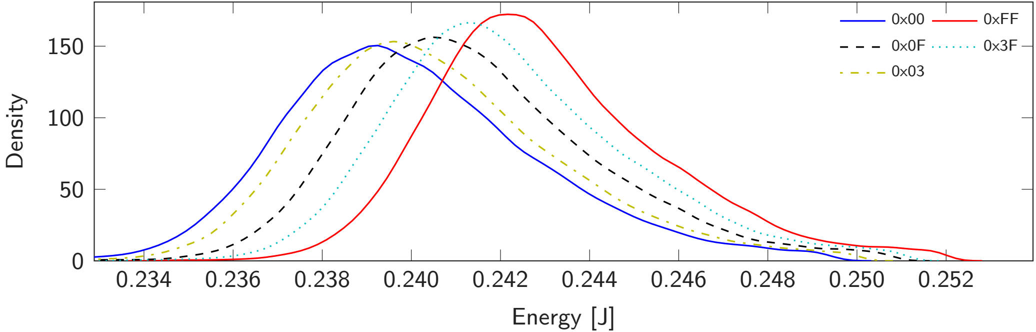
Distinguishing Instructions

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

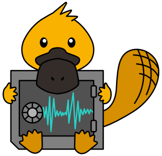


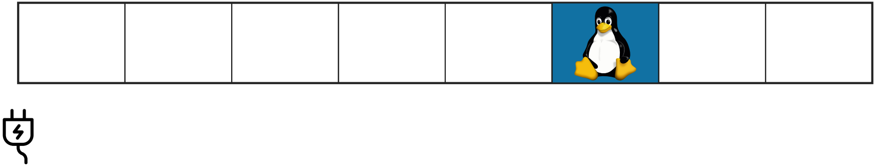
Distinguishing Operands

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

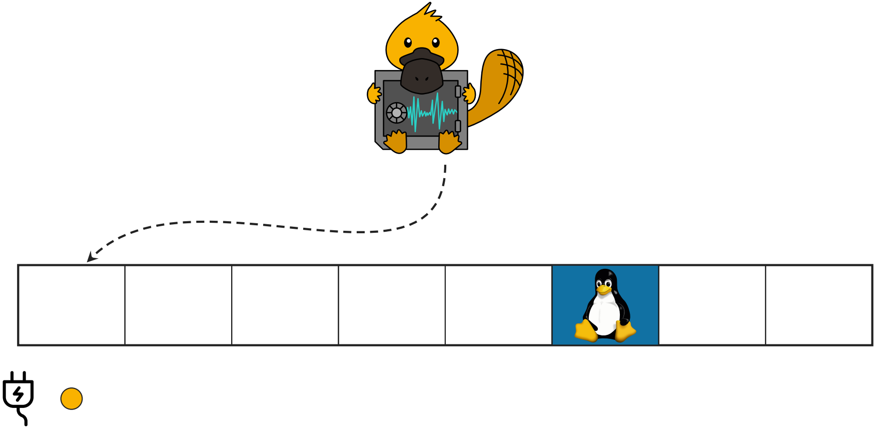


Breaking KASLR

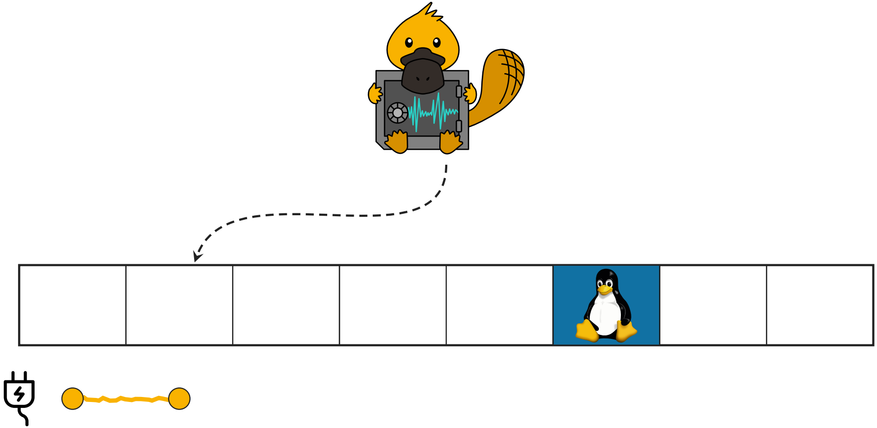




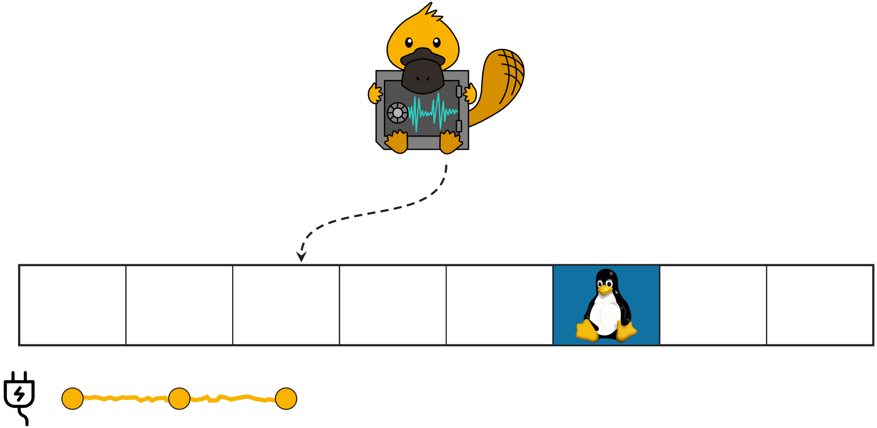
Breaking KASLR



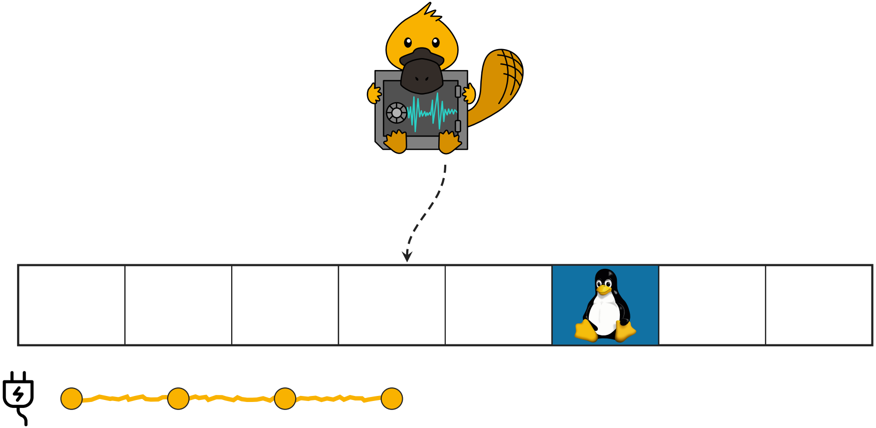
Breaking KASLR



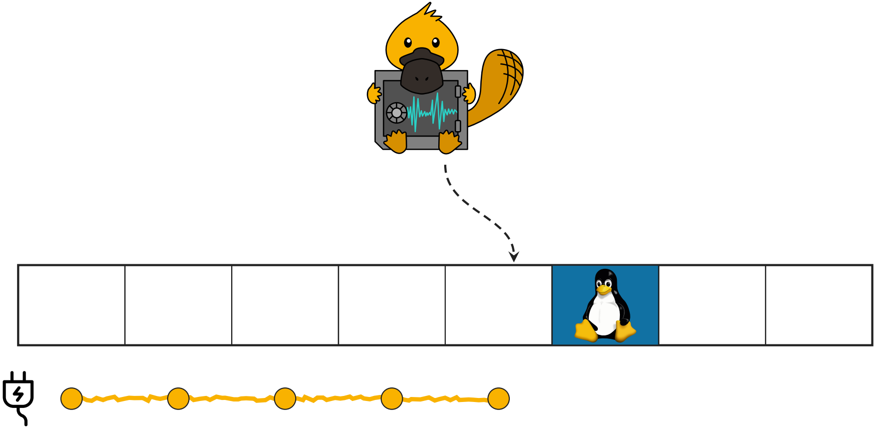
Breaking KASLR



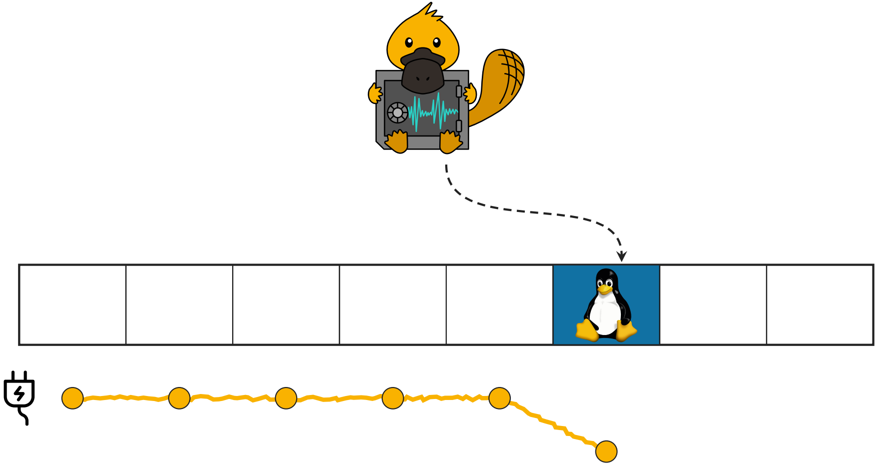
Breaking KASLR



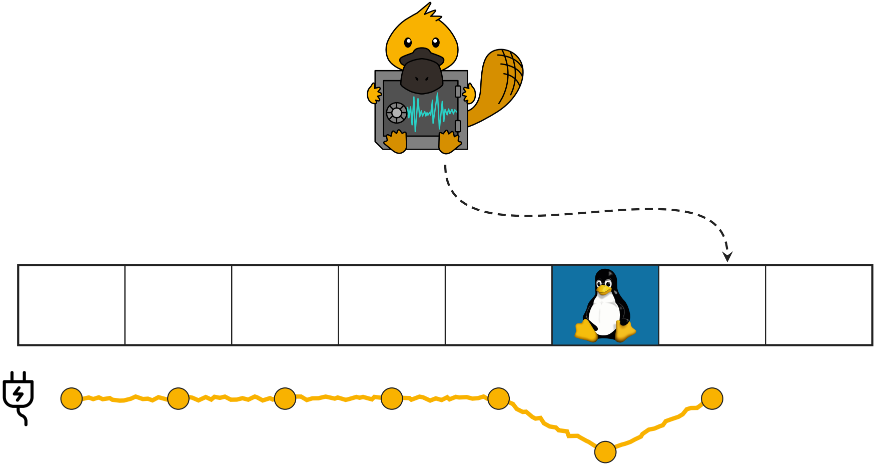
Breaking KASLR



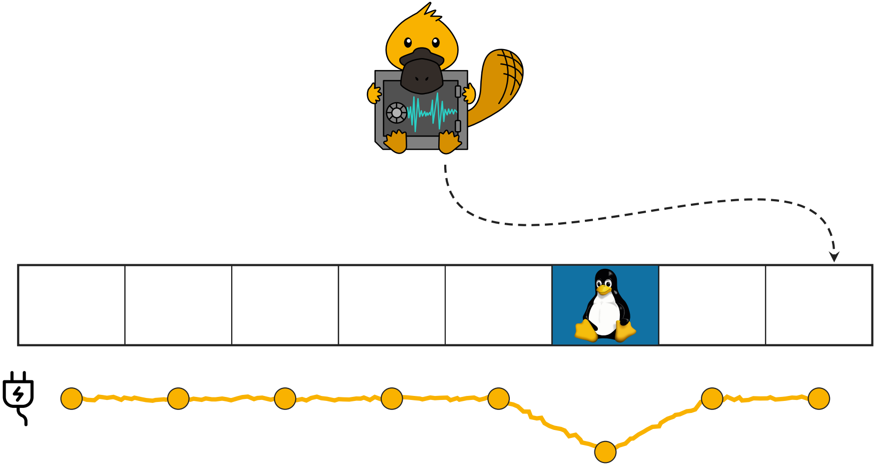
Breaking KASLR

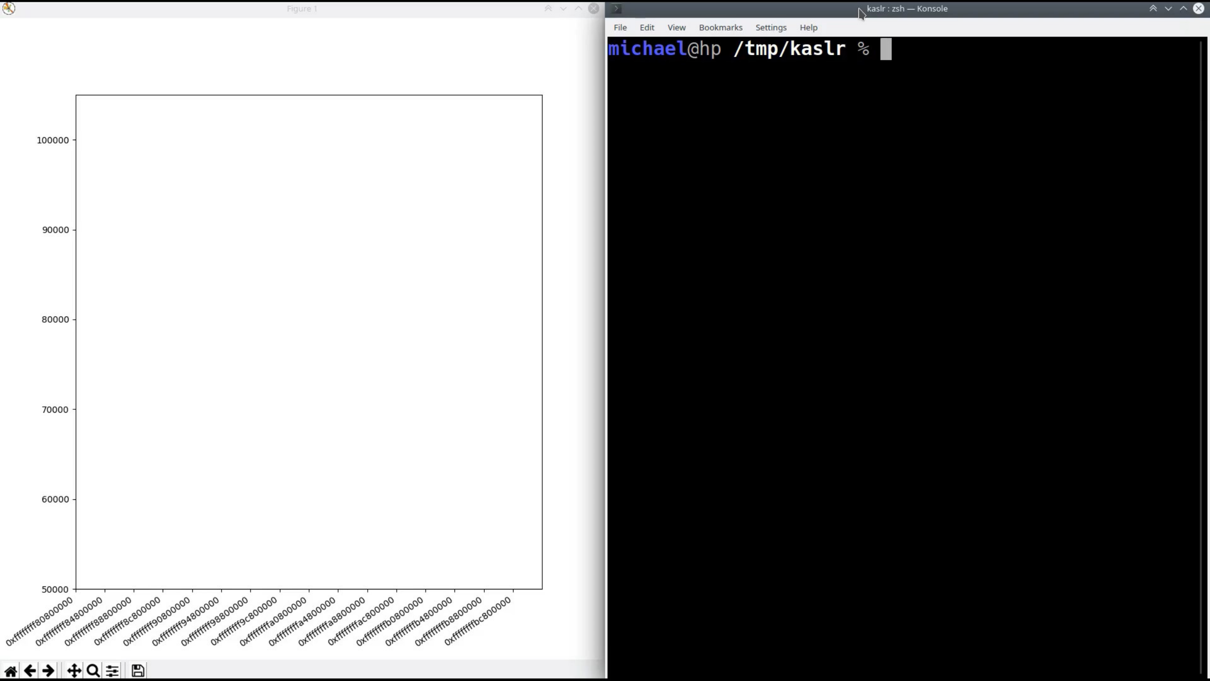


Breaking KASLR

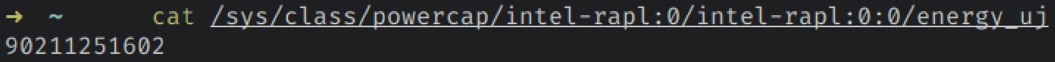


Breaking KASLR

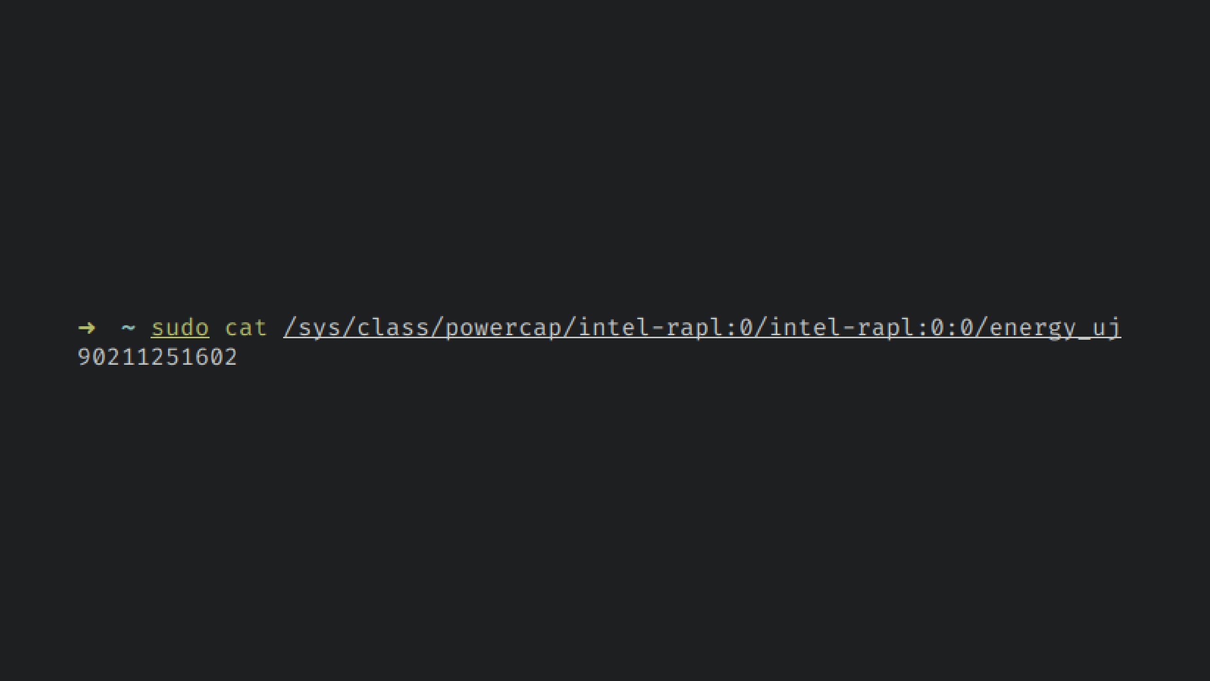




```
michael@hp /tmp/kaslr %
```



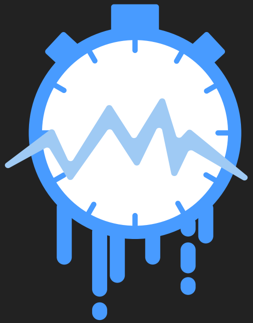
```
➜  ~      cat /sys/class/powercap/intel-rapl:0/intel-rapl:0:0/energy_uj
90211251602
```



```
➜  ~ sudo cat /sys/class/powercap/intel-rapl:0/intel-rapl:0:0/energy_uj
90211251602
```

The end?

Hertzbleed[2][3]



[2] Yingchen Wang, Riccardo Paccagnella, Elizabeth He, Hovav Shacham, Christopher W. Fletcher, and David Kohlbrenner. Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86. In: USENIX Security. 2022.

[3] Chen Liu, Abhishek Chakraborty, Nikhil Chawla, and Neer Roggel. Frequency throttling side-channel attack. In: CCS. 2022.

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

CPU Power Management

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

Shut down resources

Reduce voltage

CPU Power Management

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

Shut down resources

Reduce voltage

Reduce frequency

CPU Power Management

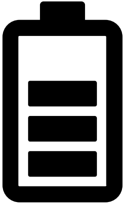
- CPU power management is complex
- In order to save power, you can . . .

Shut down resources

Reduce voltage

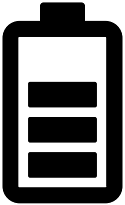
Reduce frequency

Hertzbleed Effect

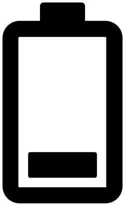


Hertzbleed Effect

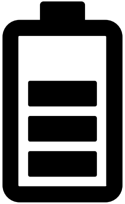
Hertzbleed Effect



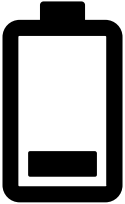
- Consumes **more** energy



Hertzbleed Effect

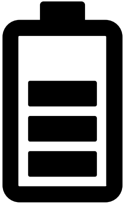


- Consumes **more** energy

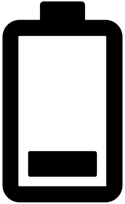


- Consumes **less** energy

Hertzbleed Effect

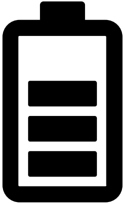


- Consumes **more** energy
- **Reaches** power limit after some time

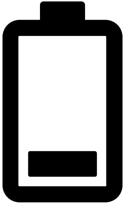


- Consumes **less** energy

Hertzbleed Effect

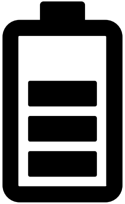


- Consumes **more** energy
- **Reaches** power limit after some time

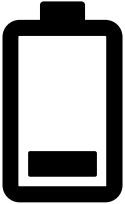


- Consumes **less** energy
- **Never reaches** power limit

Hertzbleed Effect

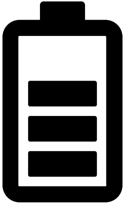


- Consumes **more** energy
- **Reaches** power limit after some time
- Throttling occurs

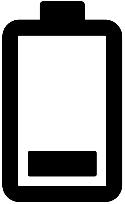


- Consumes **less** energy
- **Never reaches** power limit

Hertzbleed Effect

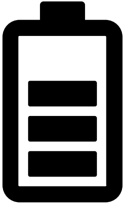


- Consumes **more** energy
- **Reaches** power limit after some time
- Throttling occurs



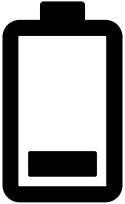
- Consumes **less** energy
- **Never reaches** power limit
- No throttling

Hertzbleed Effect



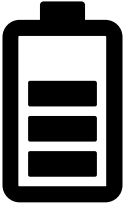
- Consumes **more** energy
- **Reaches** power limit after some time
- Throttling occurs

→ Slowdown



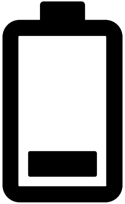
- Consumes **less** energy
- **Never reaches** power limit
- No throttling

Hertzbleed Effect



- Consumes **more** energy
- **Reaches** power limit after some time
- Throttling occurs

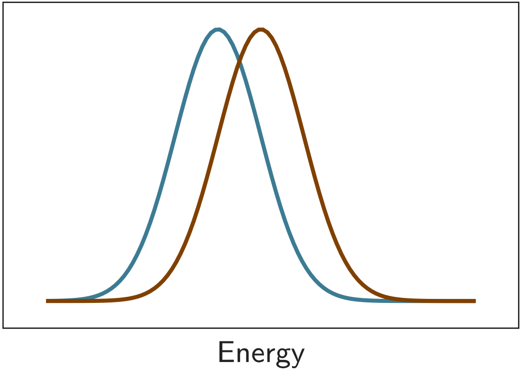
→ Slowdown

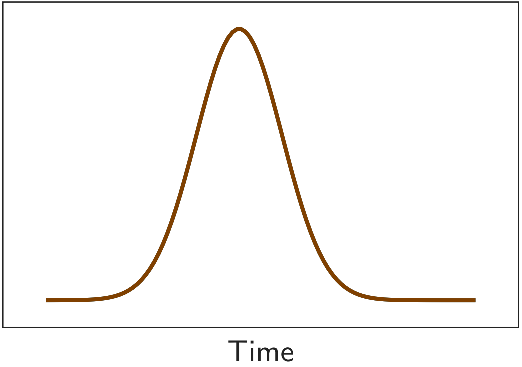


- Consumes **less** energy
- **Never reaches** power limit
- No throttling

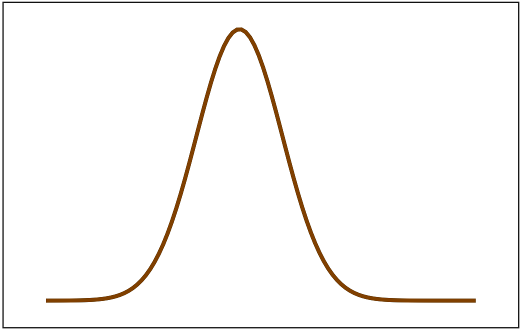
→ No slowdown

Hertzbleed Effect - Without Power Limit

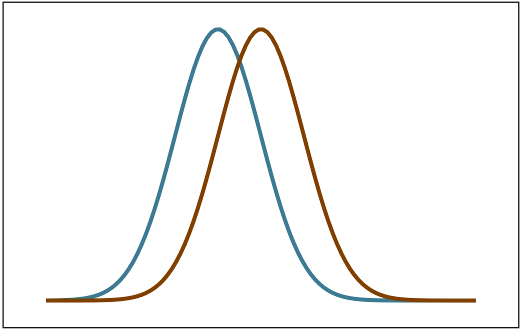




Hertzbleed Effect - With Power Limit



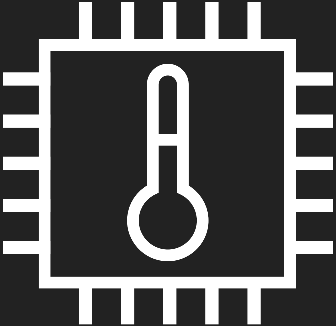
Energy



Time

NYPD
POWER
TIME
imgflip.com

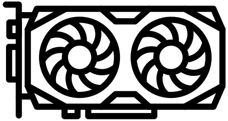
GPU Throttling[4][5]



[4] Yingchen Wang, Riccardo Paccagnella, Alan Wandke, Zhao Gang, Grant Garrett-Grossman, Christopher W Fletcher, David Kohlbrenner, and Hovav Shacham. DVFS frequently leaks secrets: Hertzbleed attacks beyond SIKE, cryptography, and CPU-only data. In: S&P. 2023.

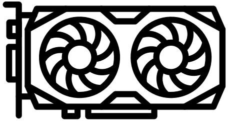
[5] Hritvik Taneja, Jason Kim, Jie Jeff Xu, Stephan van Schaik, Daniel Genkin, and Yuval Yarom. Hot Pixels: Frequency, Power, and Temperature Attacks on GPUs and ARM SoCs. In: USENIX Security.

GPU Throttling



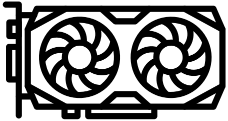
- **Integrated** GPUs share power limits with the CPU

GPU Throttling



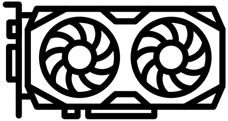
- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption

GPU Throttling



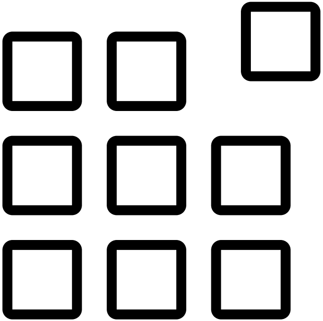
- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption
- **Dedicated** GPUs have power limits too

GPU Throttling



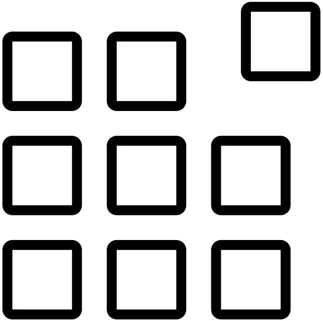
- **Integrated** GPUs share power limits with the CPU
  → **CPU throttling** indicates high GPU consumption
- **Dedicated** GPUs have power limits too
  → **Observable** by timing a GPU workload

Pixel Stealing



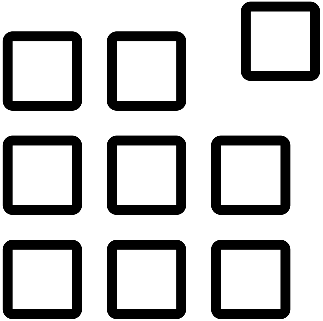
- What secrets are *“inside”* a GPU?

Pixel Stealing



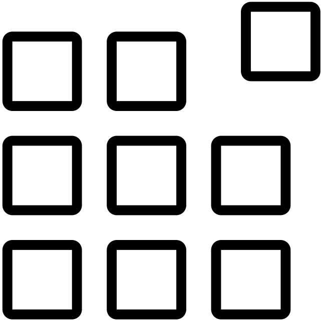
- What secrets are *“inside”* a GPU?
    - GPU renders windows and screen

Pixel Stealing



- What secrets are *"inside"* a GPU?
  - GPU renders windows and screen
  - → **Privacy** related information

Pixel Stealing



- 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

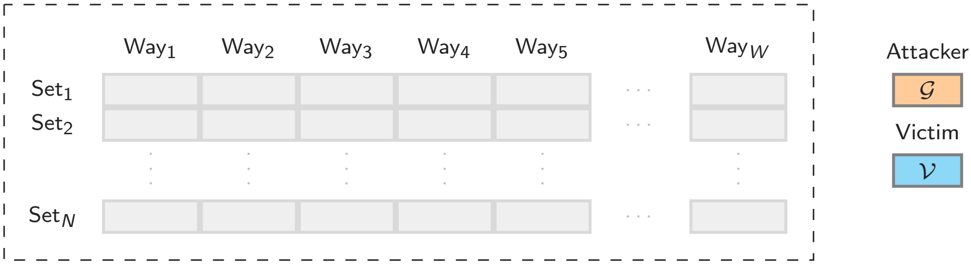
→ Measure timing and infer pixel value

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

Collide+Power[6]

[6] Andreas Kogler, Jonas Juffinger, Lukas Giner, Lukas Gerlach, Martin Schwarzl, Michael Schwarz, Daniel Gruss, and Stefan Mangard. Collide+Power: Leaking Inaccessible Data with Software-based Power Side Channels. In: USENIX Security. 2023.

Collide+Power - Memory Subsystem



Way$_1$ Way$_2$ Way$_3$ Way$_4$ Way$_5$ ... Way$_W$

Set$_1$

Set$_2$

Set$_N$

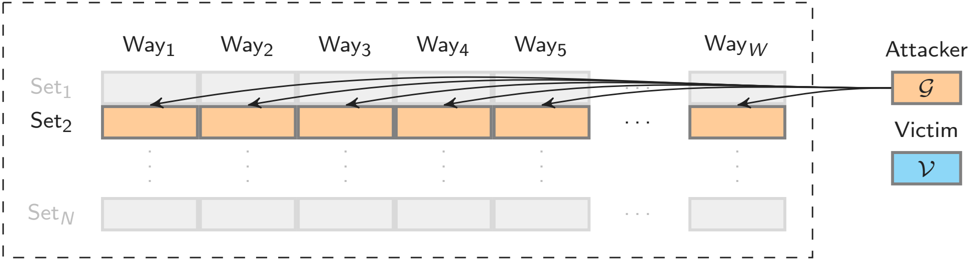
Attacker

$\mathcal{G}$

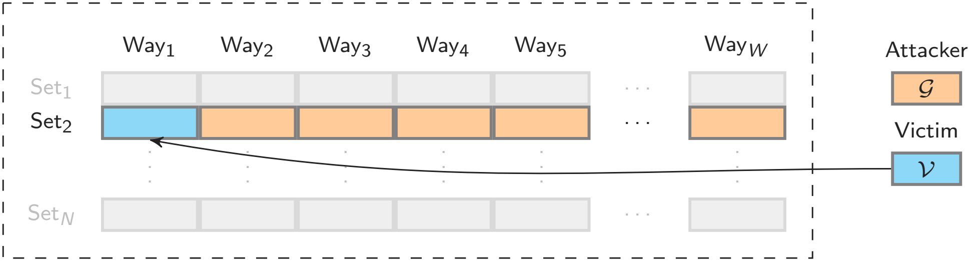
Victim

$\mathcal{V}$

Collide+Power - Memory Subsystem



Collide+Power - Memory Subsystem



Way_1 Way_2 Way_3 Way_4 Way_5 Way_W

Set_1

Set_2

Set_N

Attacker

$\mathcal{G}$

Victim

$\mathcal{V}$

Power Leakage - Model Components

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Number of set bits

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Number of set bits

$\mathrm{hw}(11_2) = 2$

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Number of set bits

$\mathrm{hw}(11_2) = 2$

Hamming Distance: $\mathrm{hd}(x, y)$

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Number of set bits

$\mathrm{hw}(11_2) = 2$

Hamming Distance: $\mathrm{hd}(x, y)$

Number of different bits

Power Leakage - Model Components

Hamming Weight: $\mathrm{hw}(x)$

Number of set bits

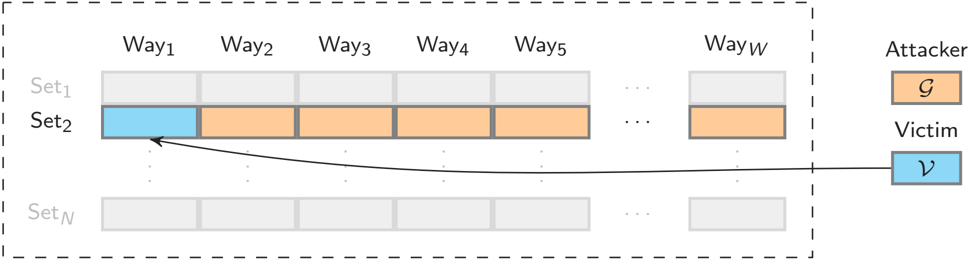
$\mathrm{hw}(11_2) = 2$

Hamming Distance: $\mathrm{hd}(x, y)$

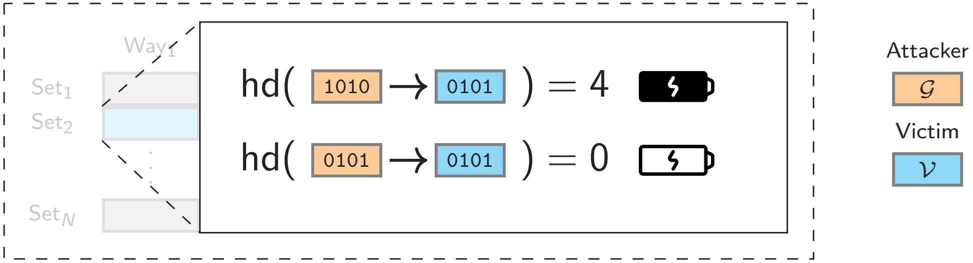
Number of different bits

$\mathrm{hd}(11_2, 01_2) = 1$

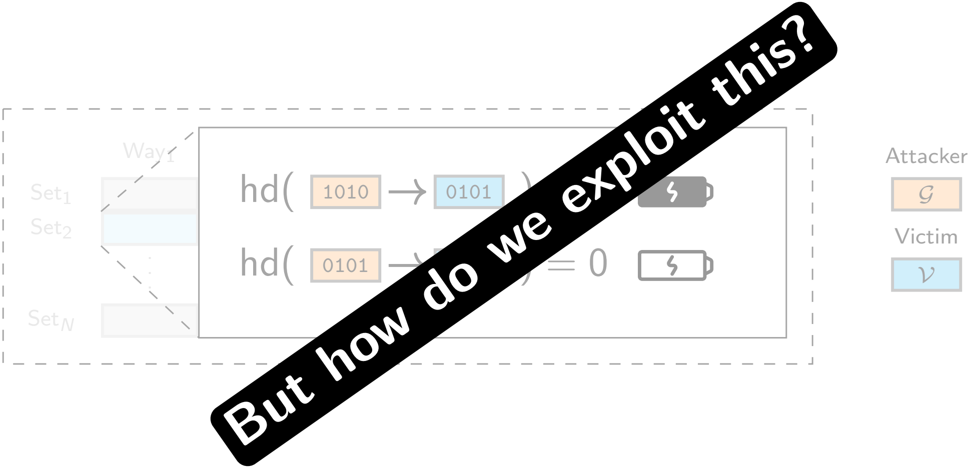
Collide+Power - Memory Subsystem



Collide+Power - Memory Subsystem



Collide+Power - Memory Subsystem



But how do we exploit this?

Collide+Power - Intuition

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

Collide+Power - Intuition

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

Collide+Power - Intuition

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

Collide+Power - Intuition

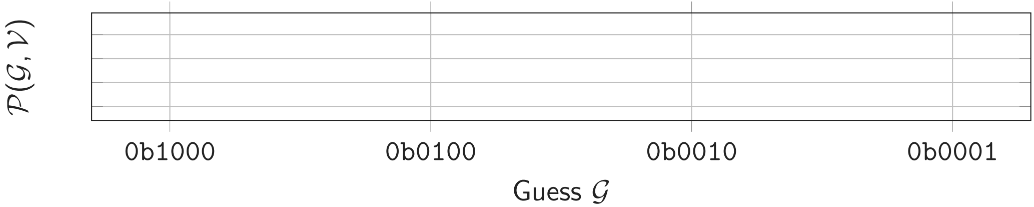
$$\underbrace{\mathcal{P}(\mathcal{G}, \mathcal{V})}_{\text{model}} \approx \mathrm{hd}(\mathcal{G}, \mathcal{V})$$

Collide+Power - Intuition

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

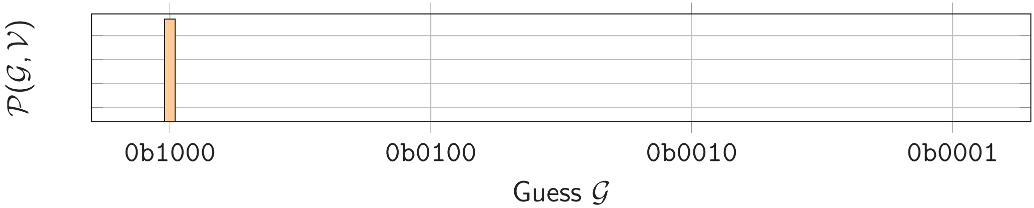
Collide+Power - Example

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



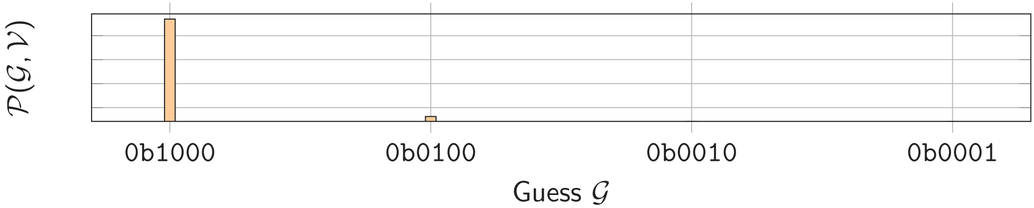
Collide+Power - Example

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



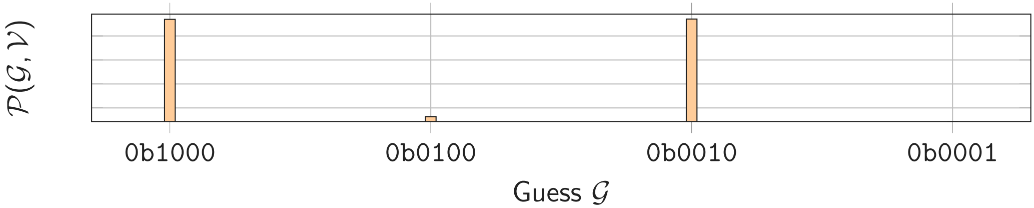
Collide+Power - Example

$$\mathcal{P}(0100_2, 0101_2) \approx \mathrm{hd}(0\mathbf{1}00_2, 0\mathbf{1}01_2) = 1$$



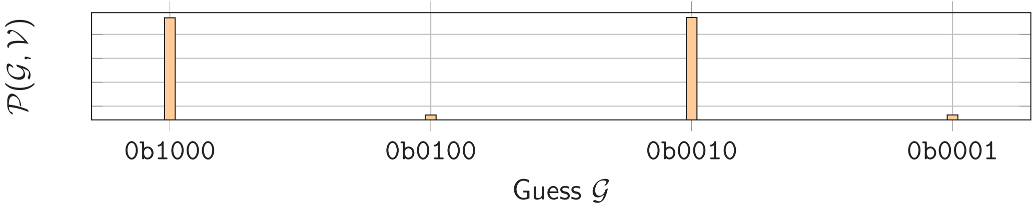
Collide+Power - Example

$$\mathcal{P}(0010_2, 0101_2) \approx \mathrm{hd}(00\mathbf{1}0_2, 01\mathbf{0}1_2) = 3$$



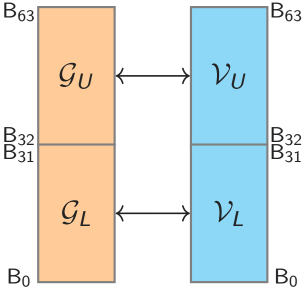
Collide+Power - Example

$$\mathcal{P}(0001_2, 0101_2) \approx \text{hd}(000\mathbf{1}_2, 010\mathbf{1}_2) = 1$$

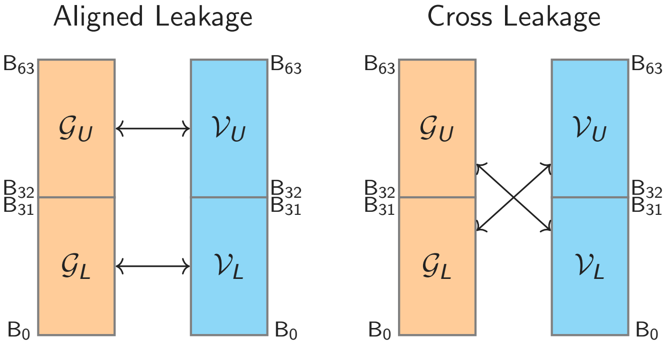


Leakage Analysis - Generalization

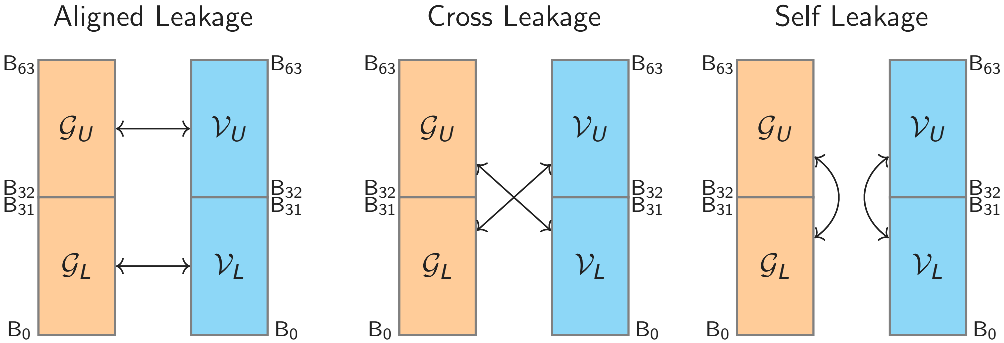
Aligned Leakage



Leakage Analysis - Generalization



Leakage Analysis - Generalization



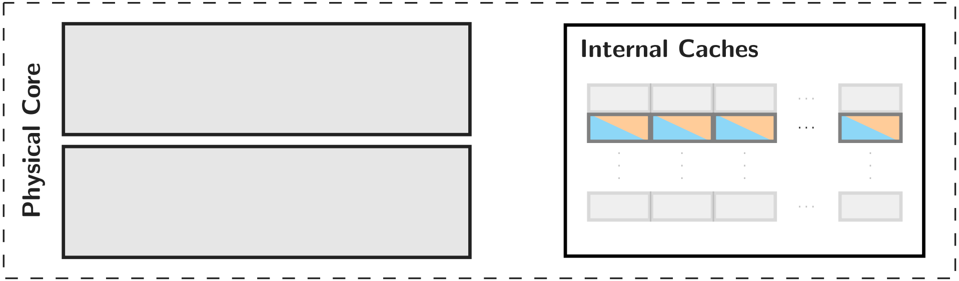
Leakage Analysis: Results

Inst.	Evict.	Effectiveness $\hat{\rho}$ $\cdot 1$	 SNR$_A$ $\cdot 10^{-3}$	Aligned Leakage hd(v_L, g_L) a_0 in μW	 hd(v_U, g_U) a_1 in μW	Cross Leakage hd(v_L, g_U) c_0 in μW	 hd(v_U, g_L) c_1 in μW	Self Leakage hd(v_L, v_U) s_0 in μW	 hd(g_L, g_U) s_1 in μW	Weights hw(v_L) [illegible]	 [illegible](v_U) [illegible]W	 hw(g_L) w_2 in μW	 hw(g_U) w_3 in μW
Load	None	0.311	72.004	544.5	4.2	1.1	0.5	0.0	[illegible]	[illegible]	0.0	362.6	0.0
	L1	0.907	7.873	598.3	278.8	0.0	0.0	[illegible]	[illegible]	0.0	0.0	6124.4	2696.9
	L1+L2	0.822	5.632	339.3	141.7	106.6	[illegible]	[illegible]	0.0	0.0	0.0	3750.7	1435.0
Prefetch	None	0.003	0.000	0.0	[illegible]	[illegible]	[illegible]	0.0	0.0	0.0	0.0	1.7	2.8
	L1	0.370	11.365	[illegible]	[illegible]	[illegible]	0.1	0.0	0.0	0.0	0.0	454.1	455.5
	L1+L2	0.300	5.294	[illegible]	[illegible]	40.9	43.0	0.0	0.0	0.0	0.0	334.0	332.5
Store	None	0.003	0.000	[illegible]	0.0	0.0	3.1	0.0	0.0	0.0	0.0	7.0	0.0
	L1	0.241	3.876	63.3	74.5	4.9	9.6	0.0	0.0	0.0	0.0	204.6	303.2
	L1+L2	0.450	6.457	133.7	169.0	84.7	86.2	0.0	0.0	0.0	0.0	347.1	1130.5

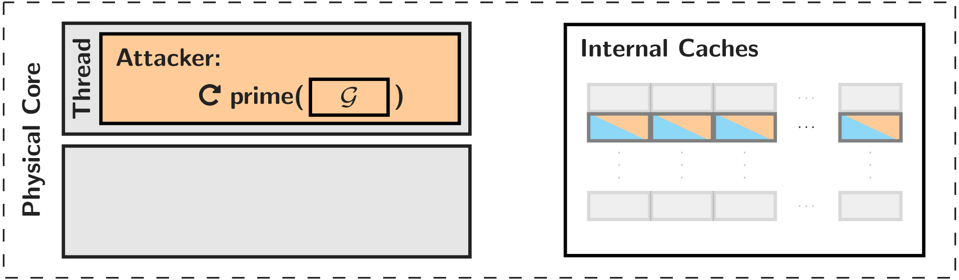
Do not start reading this!

Generic Attacks

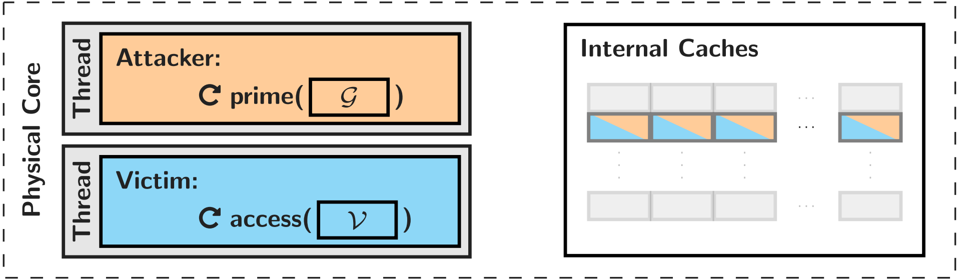
MDS-style Attack



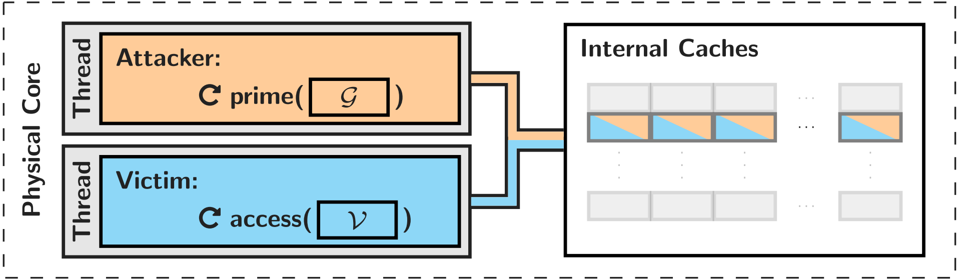
MDS-style Attack



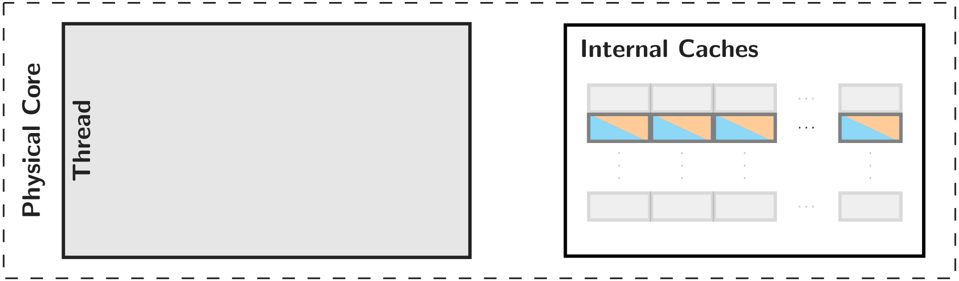
MDS-style Attack



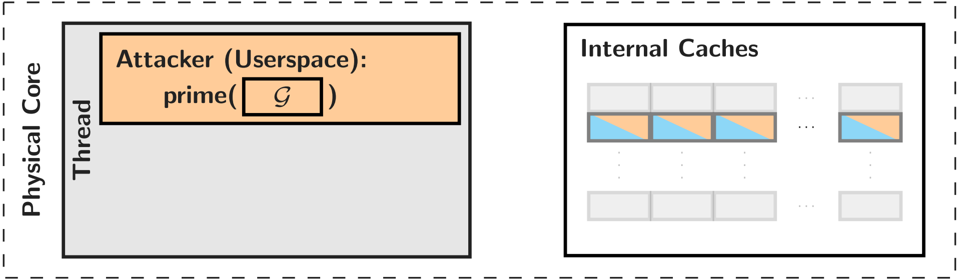
MDS-style Attack



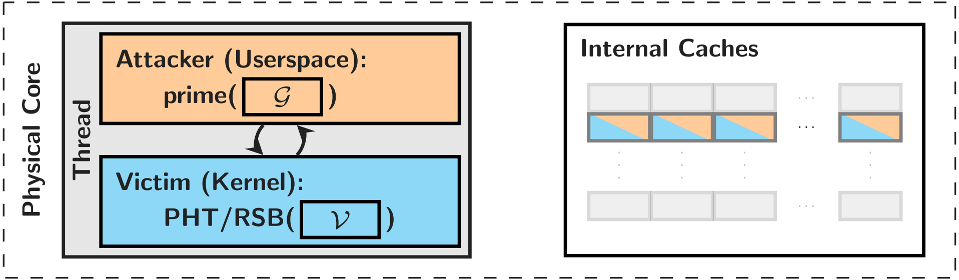
Meltdown-style Attack



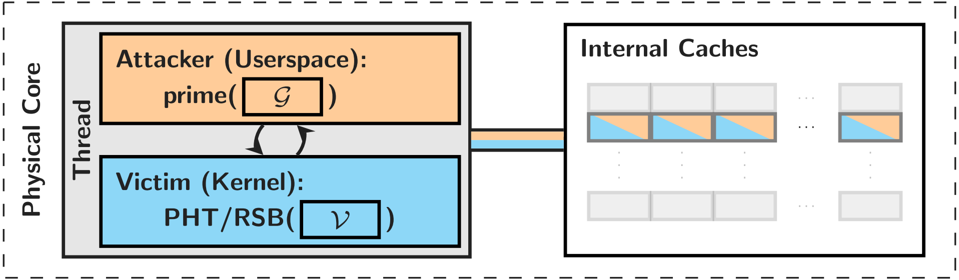
Meltdown-style Attack



Meltdown-style Attack



Meltdown-style Attack



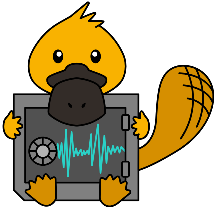
This must be slow?

NO!

It is EXTREMELY slow![7]

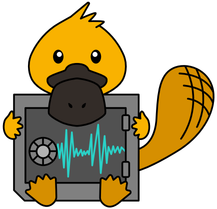
[7] With the current state-of-the-art.

Software-based Power Side Channels



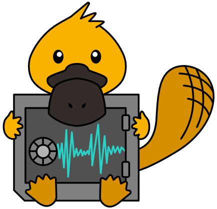
- **MDS-style:**
  4.82 bit/h

Software-based Power Side Channels



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

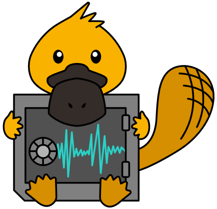
Software-based Power Side Channels



- **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



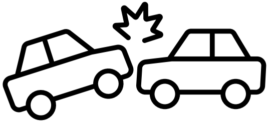
- **MDS-style:**
  4.82 bit/h
- **Meltdown-style (RSB):**
  0.84 bit/h

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

DEMO

Mitigations

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
- **Mitigating** Hertzbleed is challenging
  - Thermal and power management is required

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

Black Hat Sound Bytes

- **Unrestricted** power interfaces are a threat for system security

Black Hat Sound Bytes

black hat EUROPE 2023

- **Unrestricted** power interfaces are a threat for system security
- **Indirect interfaces** still expose exploitable information

Black Hat Sound Bytes

- **Unrestricted** power interfaces are a threat for system security
- **Indirect interfaces** still expose exploitable information
- **Software-based power side channels** can leak arbitrary data

Black Hat Sound Bytes

black hat®
EUROPE 2023

- **Unrestricted** power interfaces are a threat for system security
- **Indirect interfaces** still expose exploitable information
- **Software-based power side channels** can leak arbitrary data
- **Many more details** in the papers

  `https://collidepower.com`

  `https://hertzbleed.com`

  `https://platypusattack.com/`