Microarchitectural Attacks and Heterogenous Cloud Computing

By
Daniel Moghimi
PhD Candidate
Worcester Polytechnic Institute (WPI)
@danielmgmi

Outline

- Data Dependency
- SPOILER: Speculative Load Hazards Boost Rowhammer and Cache Attacks
- Intel SCAP: Protecting Accelerators in the Cloud

Data Dependency

```
add %ebx, %eax   1
sub %eax, %edx   2
xor %ecx, %ecx   3
add %eax, %edi   4
sub %ecx, %edi   5
```

Data Dependency - Pipelined Execution

```
add %ebx, %eax   1
sub %eax, %edx   2
xor %ecx, %ecx   3
add %eax, %edi   4
sub %ecx, %edi   5
```

IF	ID
	IF

IF Instruction Fetch

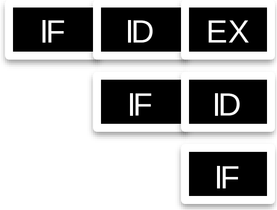
ID Instruction Decode

EX Execute

WB Write Back

Data Dependency - Pipelined Execucution

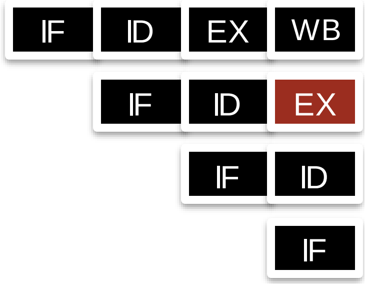
```
add %ebx, %eax   1
sub %eax, %edx   2
xor %ecx, %ecx   3
add %eax, %edi   4
sub %ecx, %edi   5
```



- IF Instruction Fetch
- ID Instruction Decode
- EX Execute
- WB Write Back

Data Dependency - Pipelined Execuction

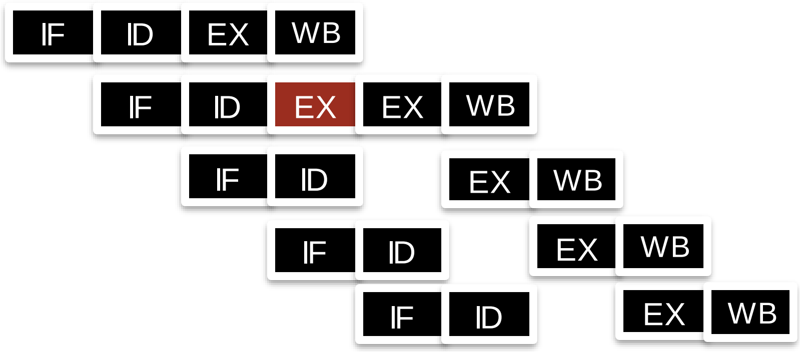
```
add %ebx, %eax   1
sub %eax, %edx   2
xor %ecx, %ecx   3
add %eax, %edi   4
sub %ecx, %edi   5
```



IF	Instruction Fetch
ID	Instruction Decode
EX	Execute
WB	Write Back

Data Dependency - Pipelined Execution

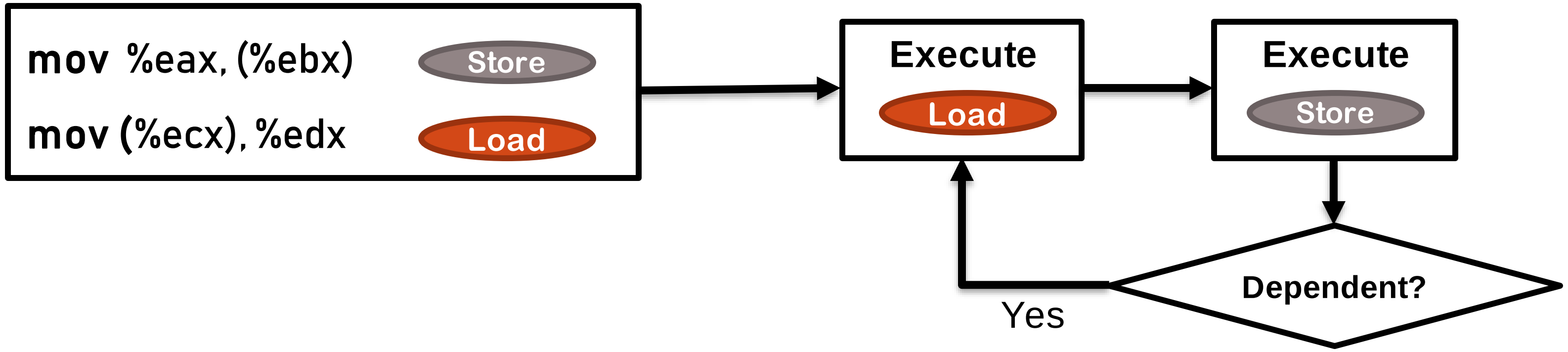
```
add %ebx, %eax   1
sub %eax, %edx   2
xor %ecx, %ecx   3
add %eax, %edi   4
sub %ecx, %edi   5
```



- IF Instruction Fetch
- ID Instruction Decode
- EX Execute
- WB Write Back

4K Aliasing False Dependency

- Memory loads/stores are executed out of order and speculatively
- The dependency is verified after the execution!



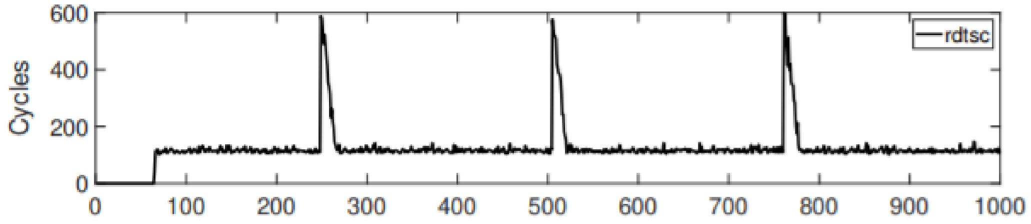
- 4K Aliasing: Addresses that are 4K apart are assumed dependent
- Re-execute the **load** and corresponding instructions due to false dependency
- Virtual-to-physical address translation → Memory disambiguation

SPOILER

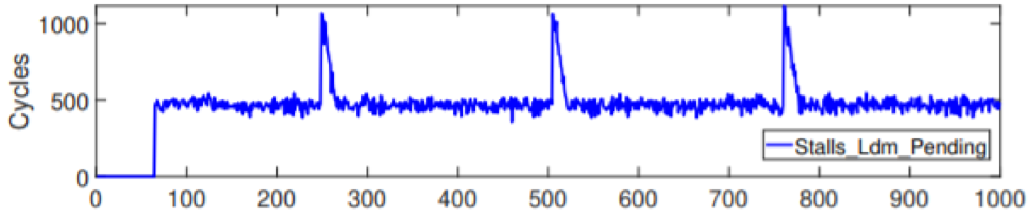
1 MB Aliasing False Dependency

Algorithm 1 Address Aliasing

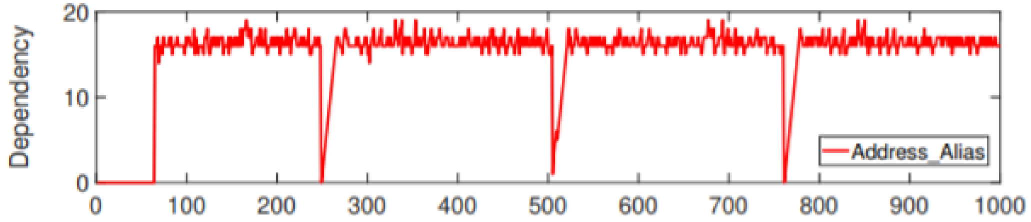
for p **from** w **to** PAGE_COUNT **do**
  for i **from** w **to** 0 **do**
    $data \xrightarrow{store} buffer[(p-i)\times PAGE\_SIZE]$
  end for
  $t_1 = rdtscp()$
  $data \xleftarrow{load} buffer[x \times PAGE\_SIZE]$
  $t_2 = rdtscp()$
  $measure[p] \leftarrow t_2 - t_1$
end for
return *measure*



(a) Step-wise peaks with a very high latency can be observed on some of the virtual pages

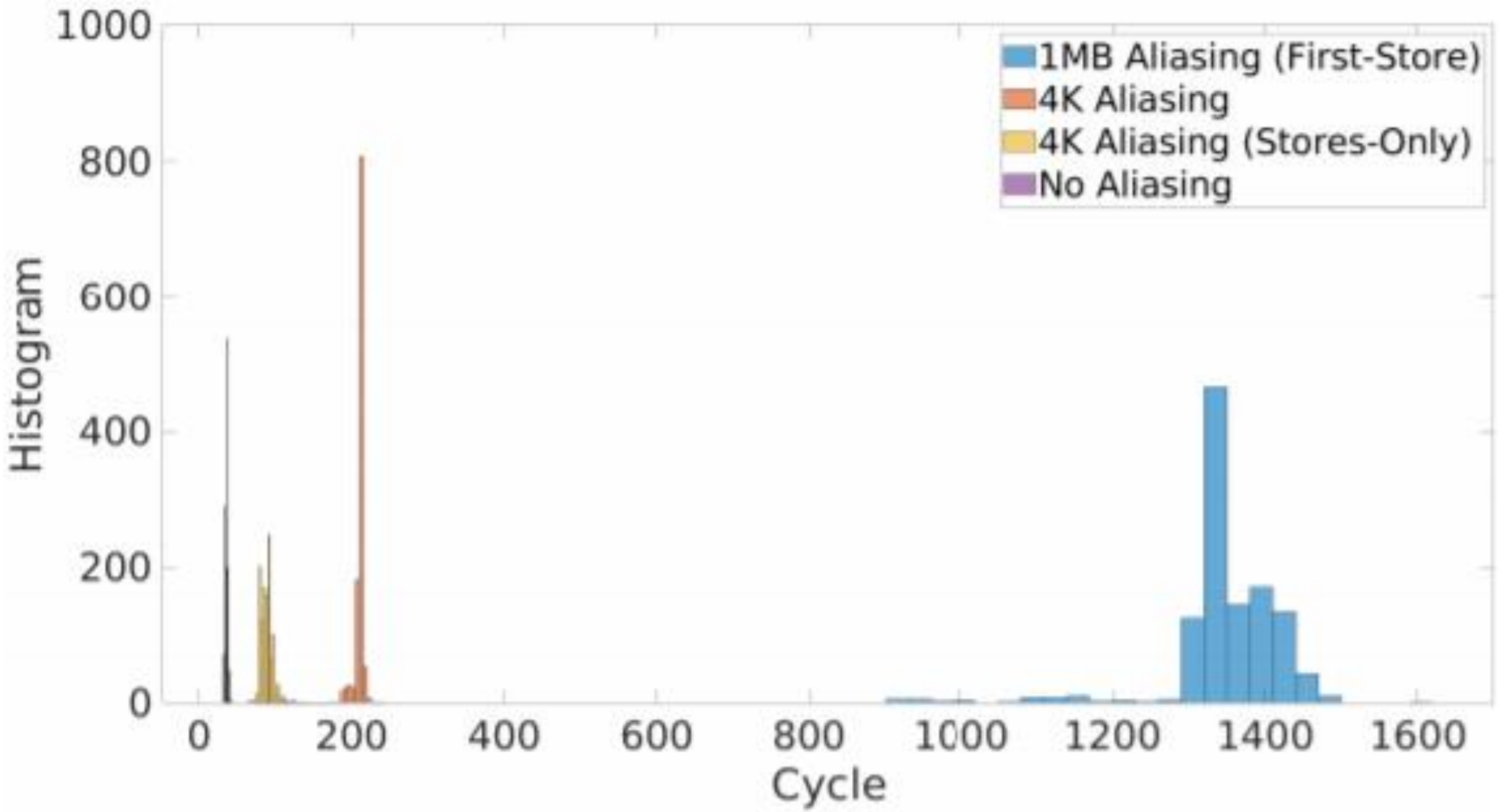


(b) Affected HPC event: `Cycle_Activity:Stalls_Ldm_Pending`

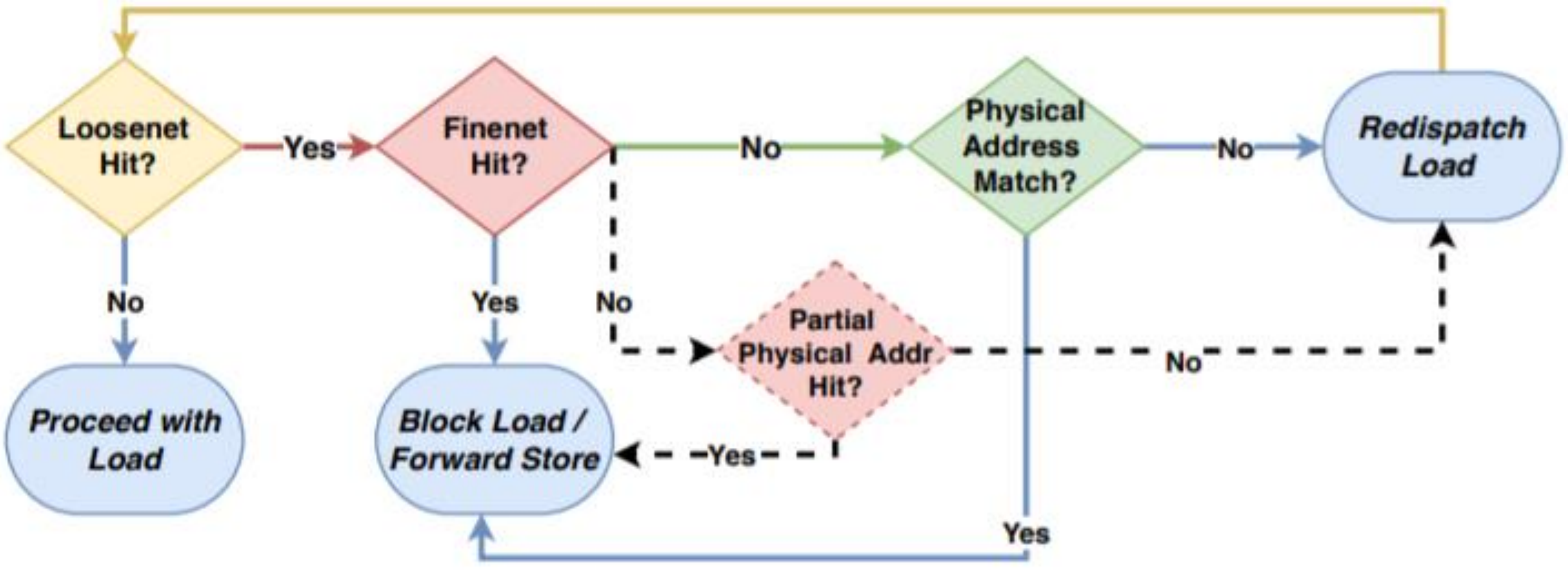


(c) Affected HPC event: `Ld_Blocks_Partial:Address_Alias`

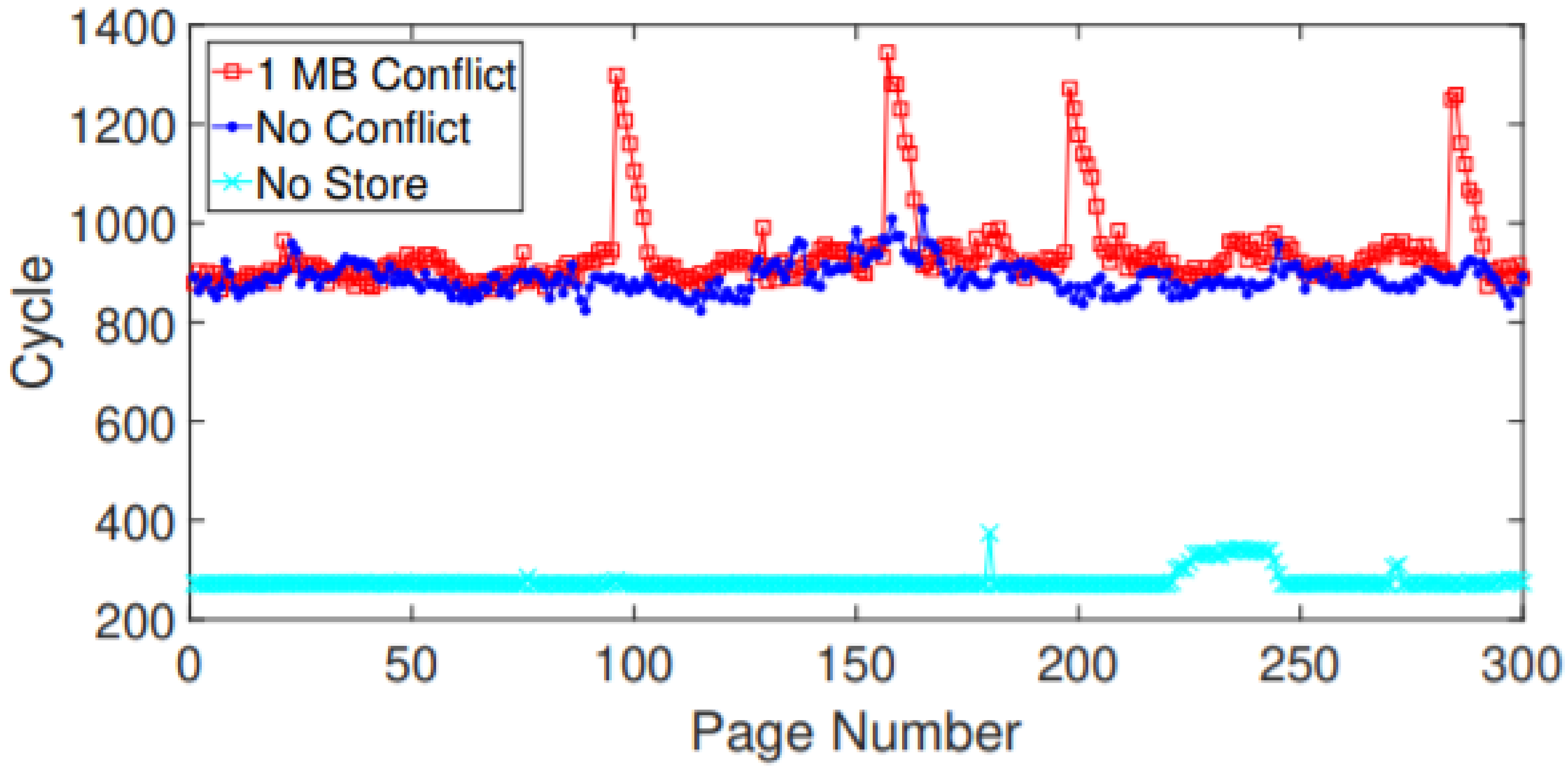
1 MB Aliasing False Dependency



1 MB Aliasing False Dependency



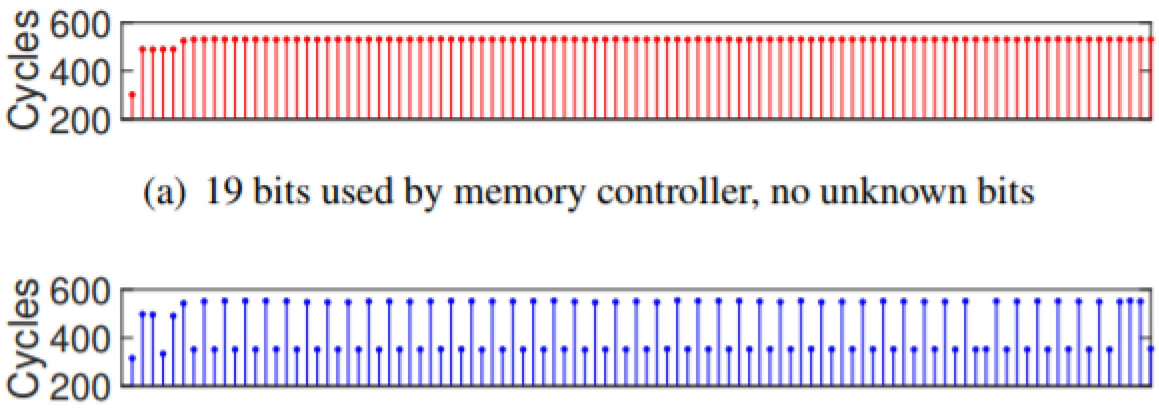
Cross-Context Address Leakage?



Rowhammer – Bank Colocation

- DRAM Banks are mapped based on the physical address

System Model	DRAM Configuration	# of Bits
Dell XPS-L702x (Sandy Bridge)	1 x (4GB 2Rx8)	21
	2 x (4GB 2Rx8)	22
Dell Inspiron-580 (Nehalem)	1 x (2GB 2Rx8) **(b)**	21
	2 x (2GB 2Rx8) **(c)**	22
	4 x (2GB 2Rx8) **(d)**	23
Dell Optiplex-7010 (Ivy Bridge)	1 x (2GB 1Rx8) **(a)**	19
	2 x (2GB 1Rx8)	20
	1 x (4GB 2Rx8) **(e)**	21
	2 x (4GB 2Rx8)	22

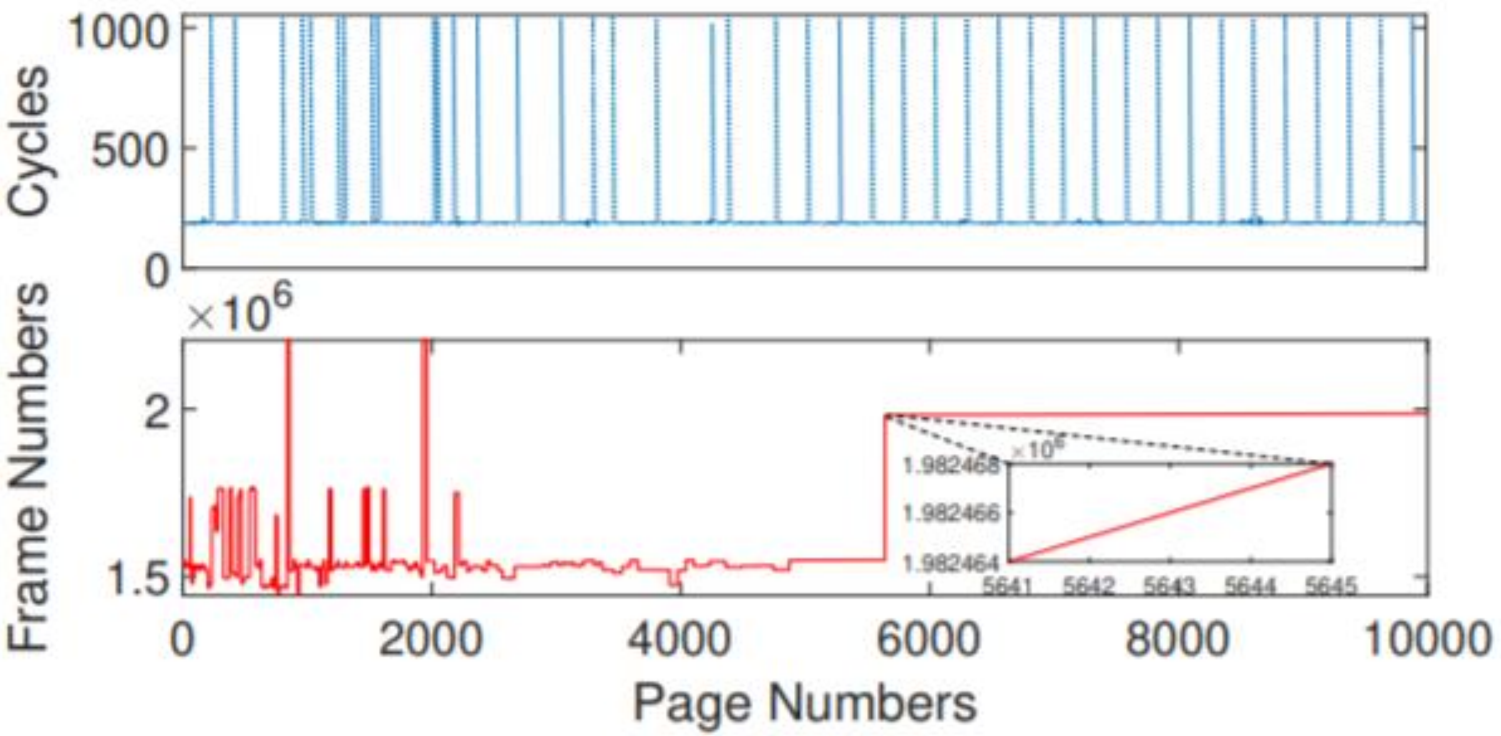


(a) 19 bits used by memory controller, no unknown bits

(b) 21 bits used by memory controller, 1 unknown bit

Rowhammer – Detecting Contiguous Memory

- Memory is contiguous when the peaks 256 apart



Cache Attacks

- Cache sets are mapped based on the physical address.
- https://github.com/UzL-ITS/Spoiler

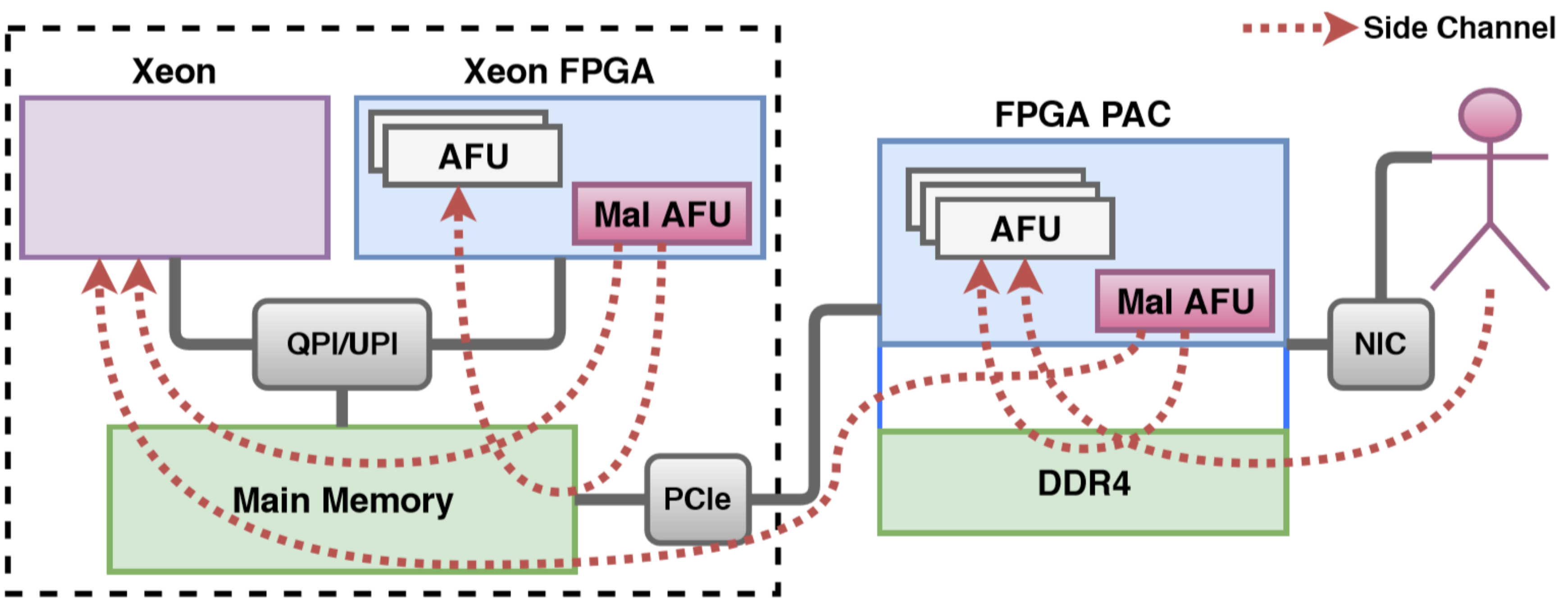
Accelerators in the Cloud

- Optimized Application-specific Hardware Configuration
- e.g. Real-time Artificial Intelligence

Side channels on Heterogeneous Accelerators

- New Attack Surface:
  - Accelerator Function Units (AFUs) placed on the FPGA can be used to interact with the CPU or other AFUs for malicious purpose.
    - AFU to AFU Attack
    - AFU to HPS Attack
    - AFU to CPU Attack
    - CPU to AFU Attack
    - Across VMS ?
- Customizable Hardware → More Devastating Attacks
  - E.g. Design your own timers, Direct access to memory interface, etc.
- Complex Threat Model

Integrated FPGA-CPU Platforms



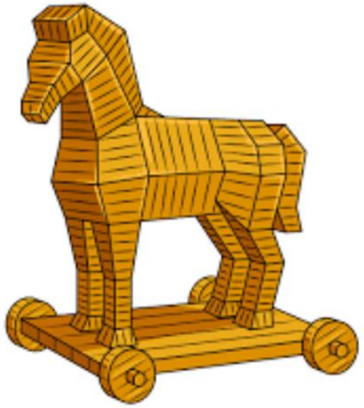
Attack Vectors

- Rowhammer

- Cache Attacks

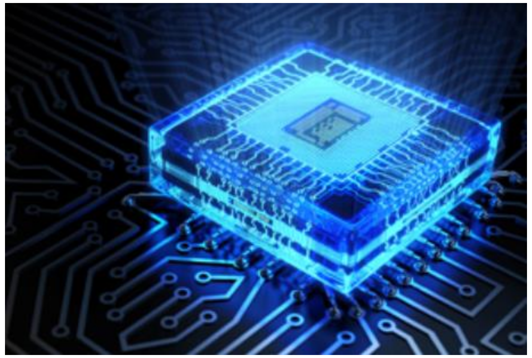
- DMA/IOMMU

- Trojan Bitstreams



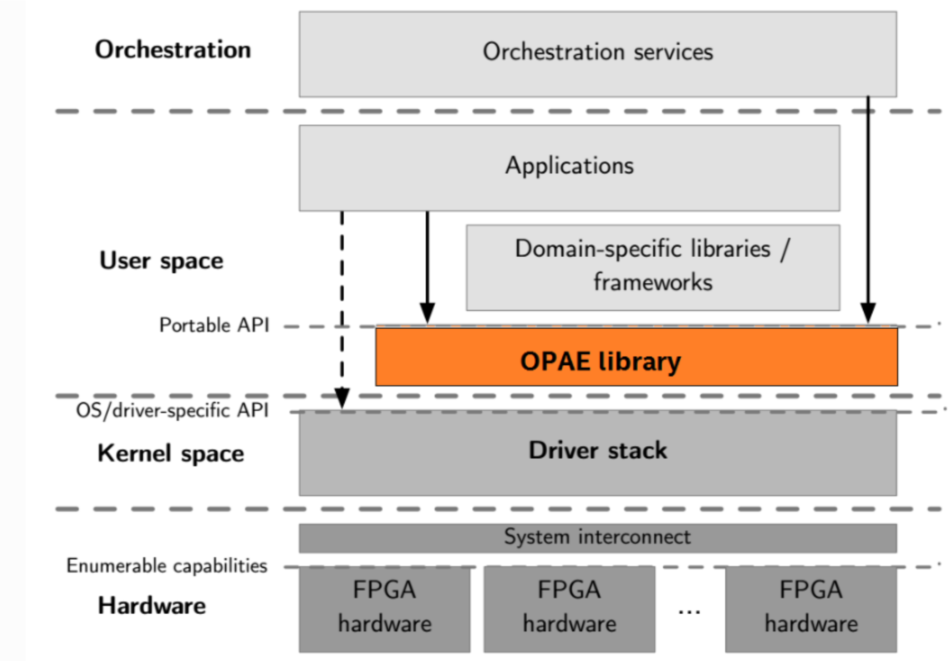
- Cold Boot

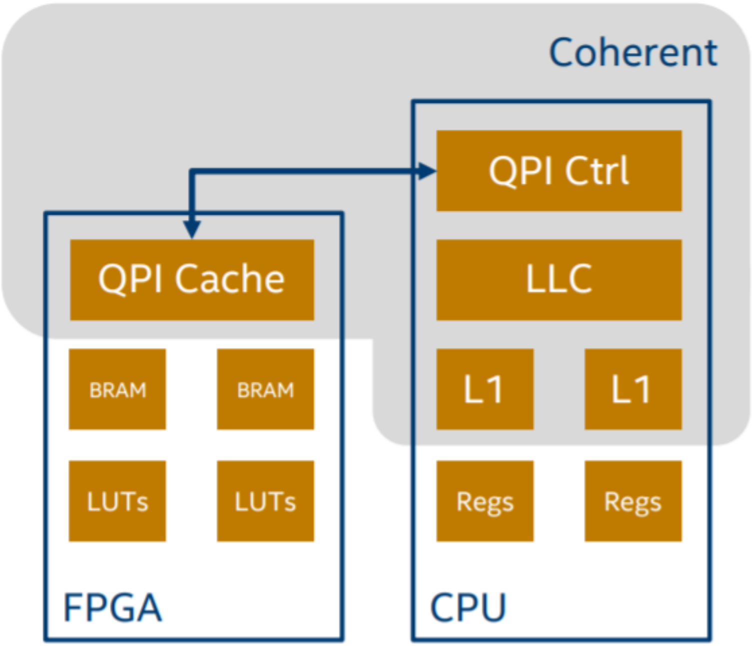
- FPGA-centric Attacks



Replicating µArch Attacks on FPGA-CPU Interface

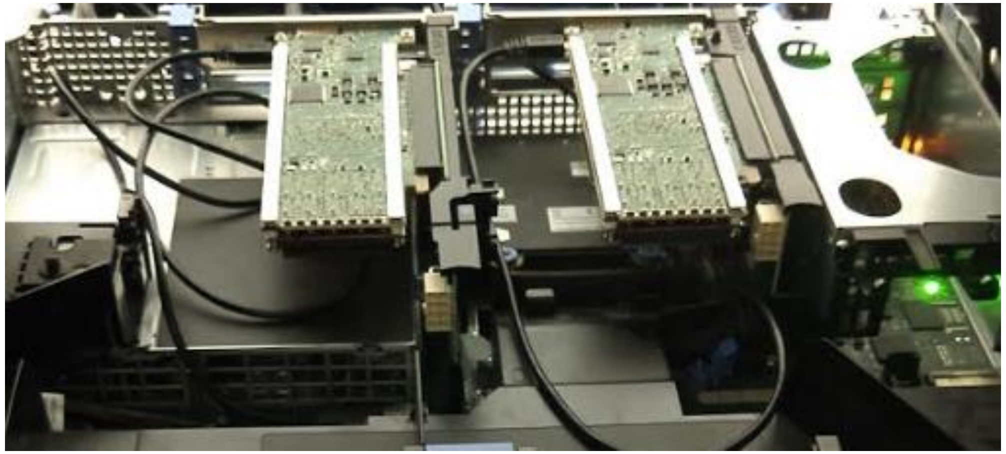
- Memory Interface and the Cache Coherency Protocol
- Side-channel Analysis of Memory Operations



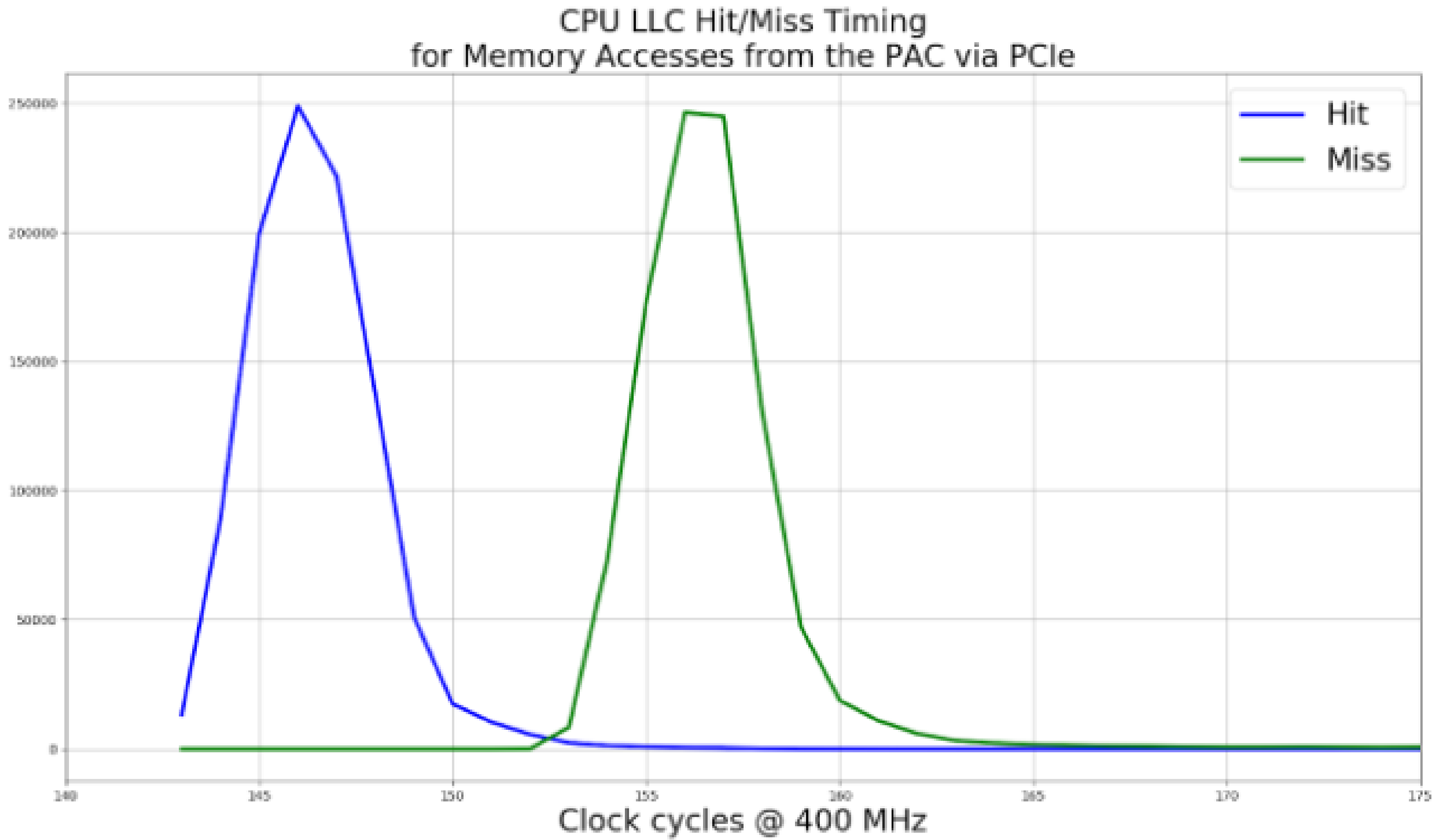


Lab/Collaboration Setup

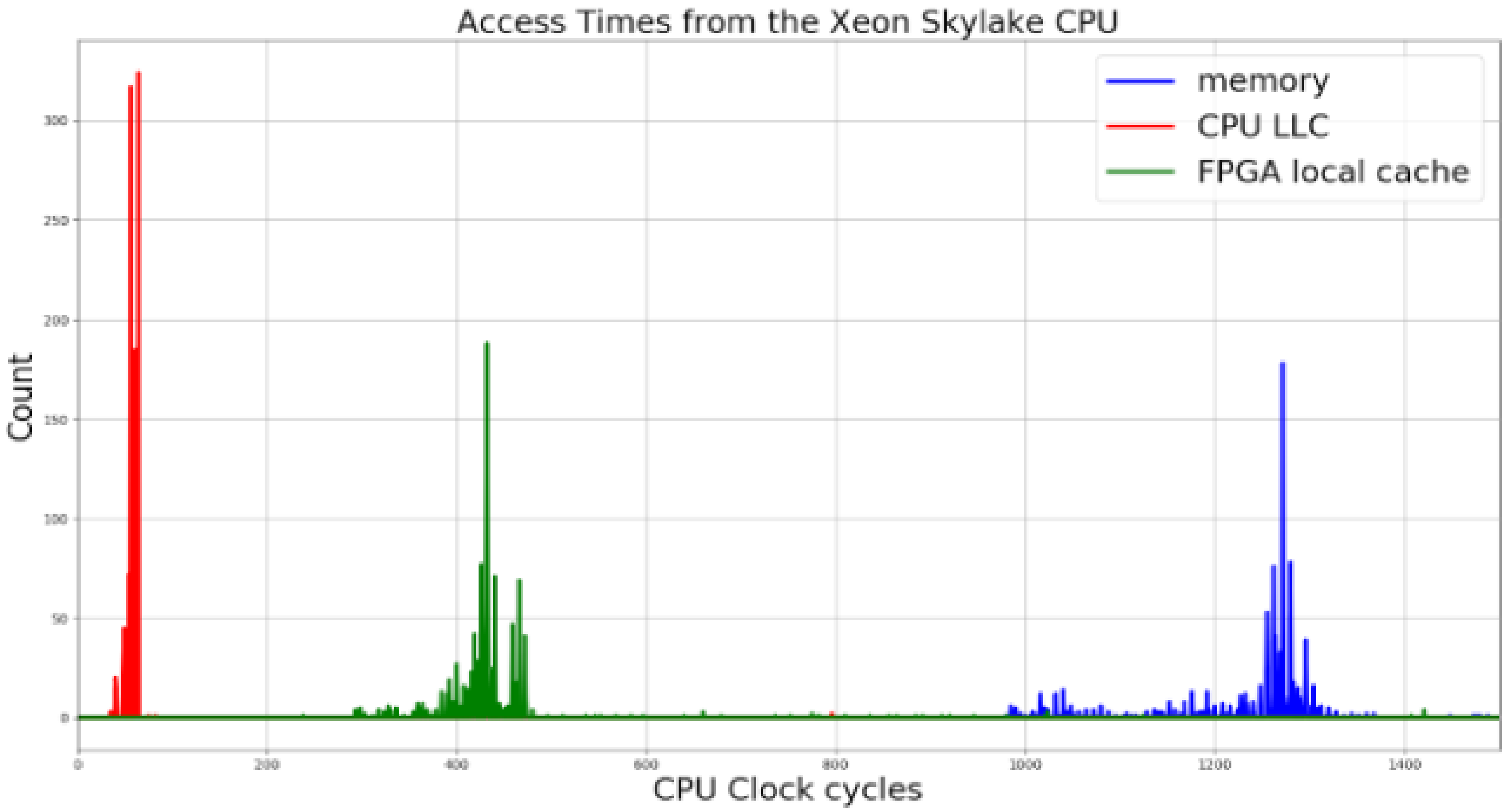
- Weekly Meeting (2 Faculty + 3 Students = 5 people are actively involved.)
- Software
  - OPAE Stack
  - Intel Quartus (Synthesis)
  - KVM (Virtualization Scenario)
- Hardware
  - Remote Access to Intel Labs (Xeon)
  - Local Server including Intel PAC
  - Heavy Load Workstation (Synthesis)



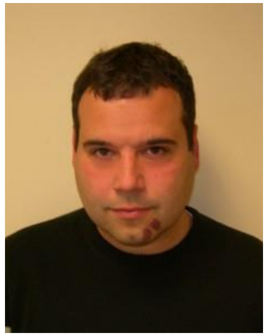
Cache Attack and FPGAs



Cache Attack and FPGAs



WPI + Lubeck Team







Other Works

- Transient Execution Attacks
  - Schwarz et al. “ZombieLoad: Cross-Privilege-Boundary Data Sampling”
  - Minkin et al. “Fallout: Reading Kernel Writes From User Space”
- Microarchitectural Side Channels
  - Islam et al. “SPOILER: Speculative Load Hazards Boost Rowhammer and Cache Attacks”
  - Moghimi et al. “MemJam: A False Dependency Attack against Constant-Time Crypto Implementations”
- Intel SGX / TEE
  - Moghimi et al. “CacheZoom: How SGX Amplifies The Power of Cache Attacks”
- Cryptographic Implementations
  - Wichelmann et al. “MicroWalk: A Framework for Finding Side Channels in Binaries”
  - Dall et al. “CacheQuote: Efficiently Recovering Long-term Secrets of SGX EPID via Cache Attacks”
  - Are remote timing attack being still a thing in 2019 !??!

Acknowledgements

- Thanks to Carlos Rosaz, Matthias Schunter, Anand Rajan, Evan Custodio and Alpa Trivedi from Intel

THANKS

- Questions?

@danielmgmi