The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications

Yossef Oren, Vasileios P. Kemerlis,

Simha Sethumadhavan, Angelos D. Keromytis

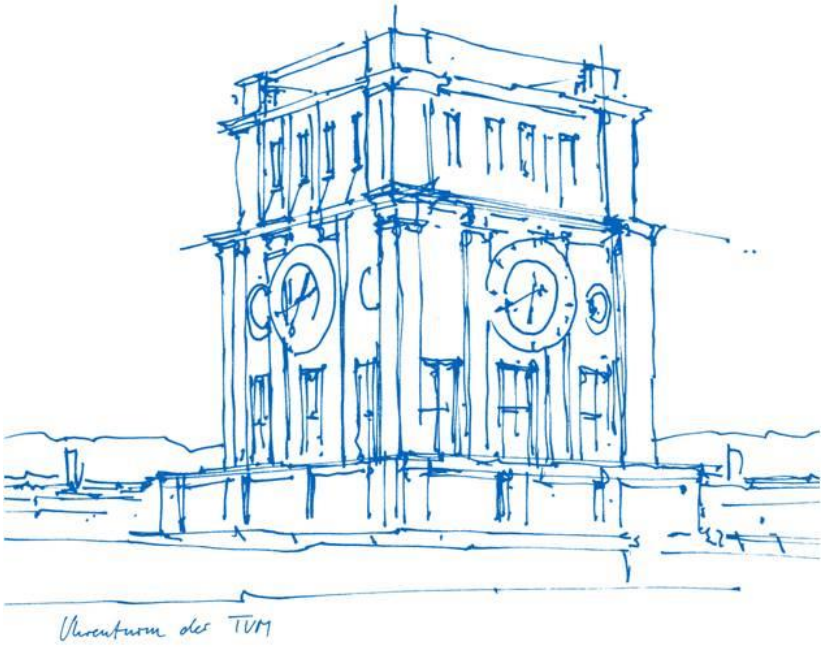
Columbia University

Floris Westermann

Technical University Munich

Chair for IT Security

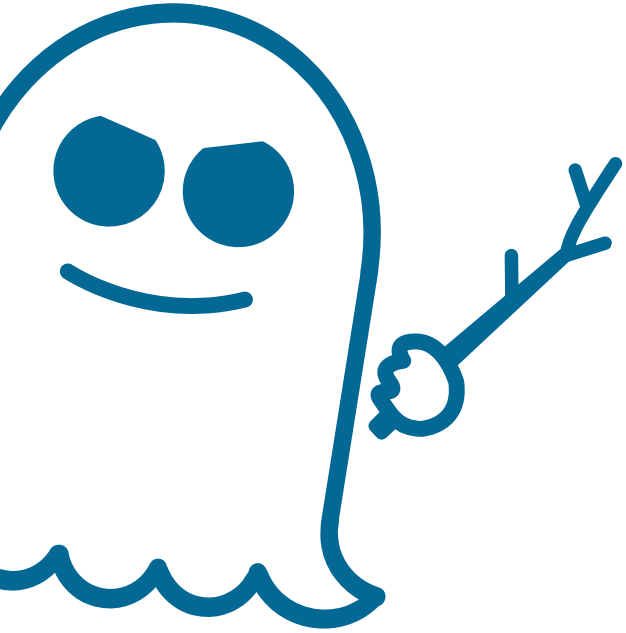
Garching, 14. May 2018



Overview

- Side Channel Attacks
- Cache Hierarchy
- Cache Attacks
- … in JavaScript
- Attacking Privacy
- Discussion

Side Channel Attacks

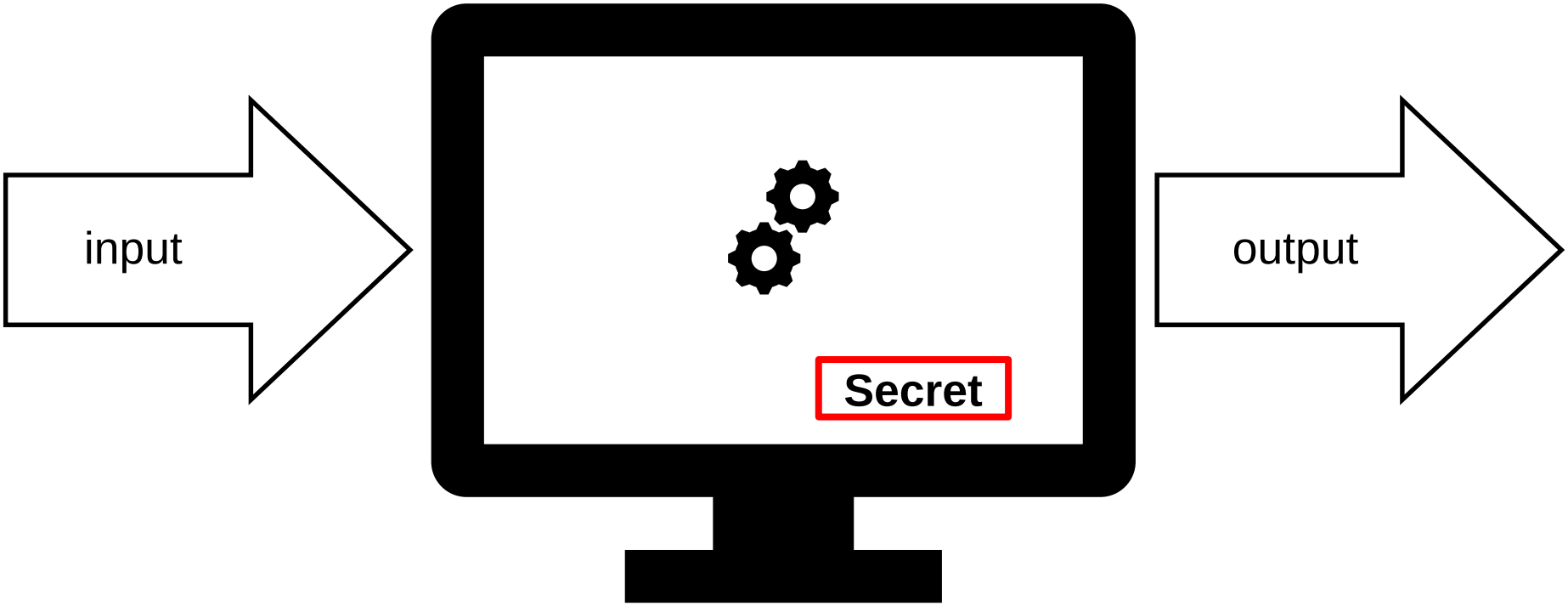


RSA

AES

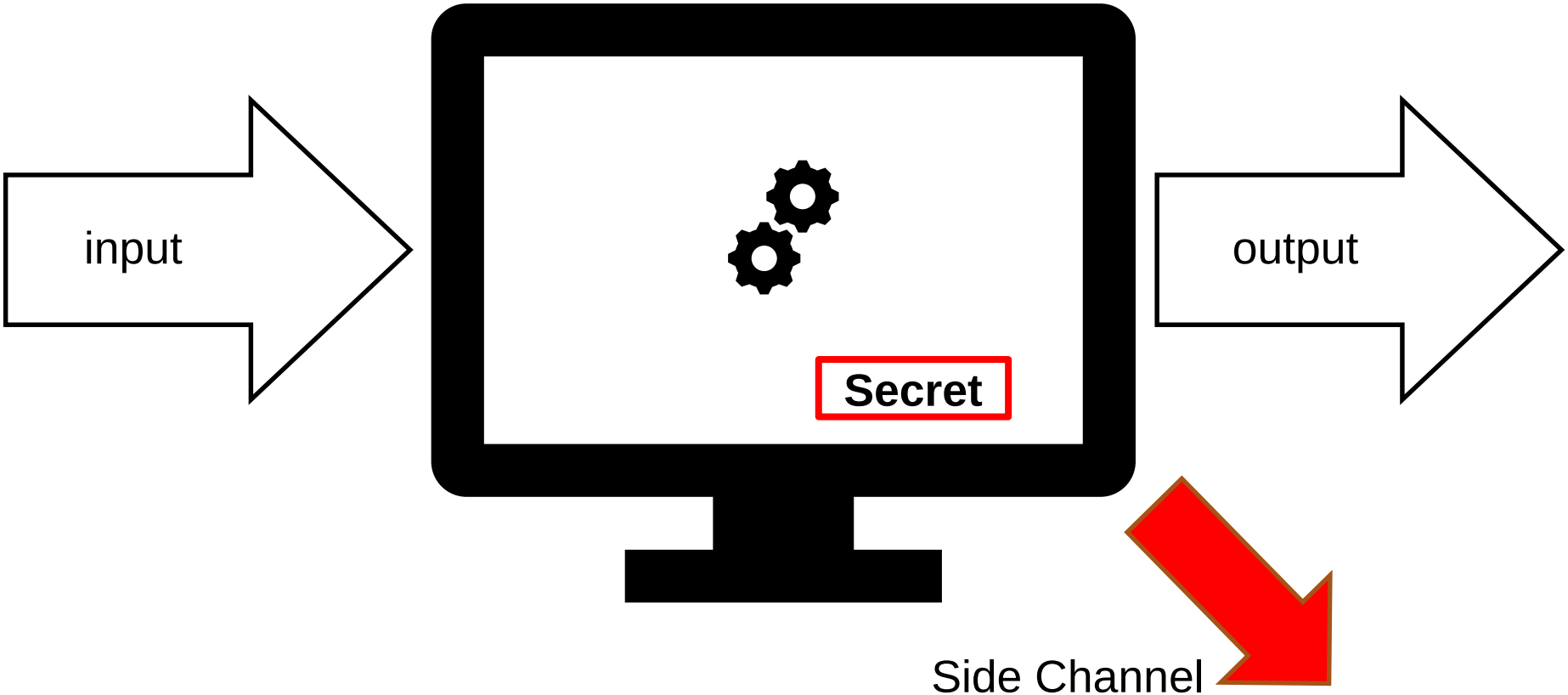
Source: https://meltdownattack.com/

Side Channel Attacks



Inspired by: https://youtu.be/hKw_vxtSNaU?t=2m52s

Side Channel Attacks



Inspired by: https://youtu.be/hKw_vxtSNaU?t=2m52s

Side Channel Attacks – Problem

Physical proximity to victim

- Install hardware device
- Run code on same machine

Possible scenarios?

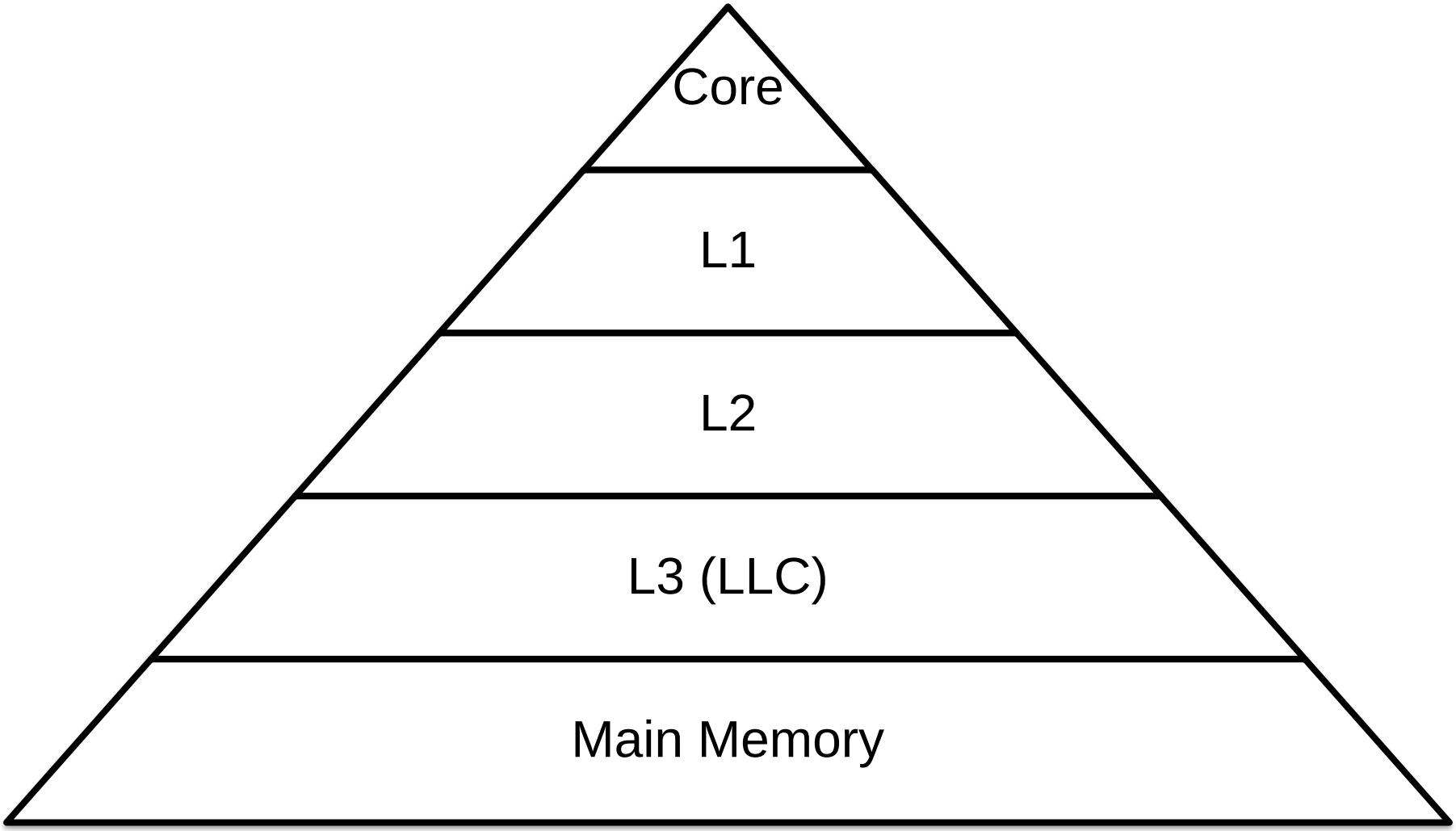
➢ Cloud computing

Side Channel Attacks – Problem

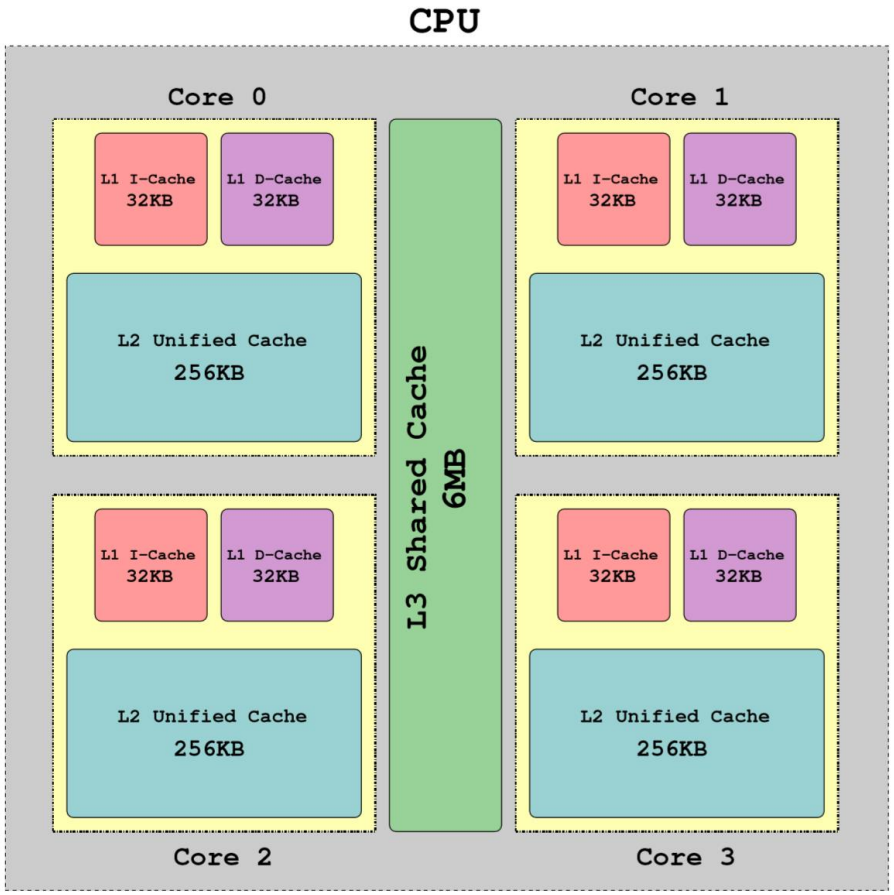
We want more!

- Reach millions of people over the Internet

Cache Hierarchy



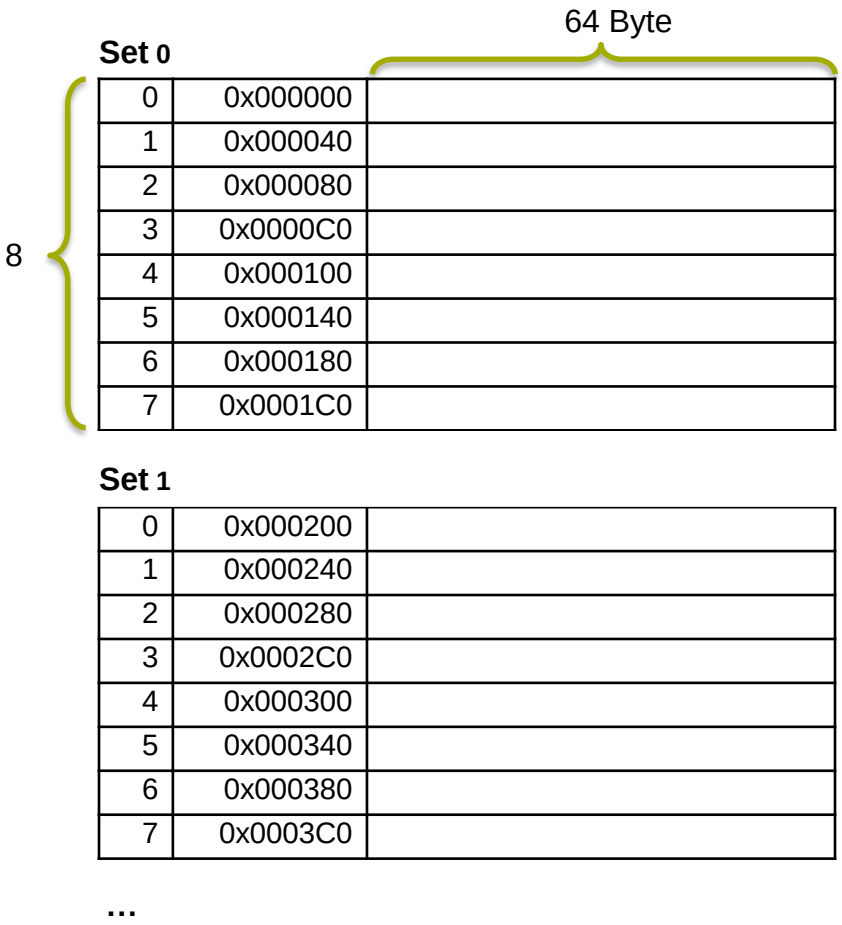
Cache Hierarchy



- Inclusive / exclusive
- Cache miss
- Cache line replacement

Source: “The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications”, Oren et al. 2015

Cache Hierarchy – Cache Set

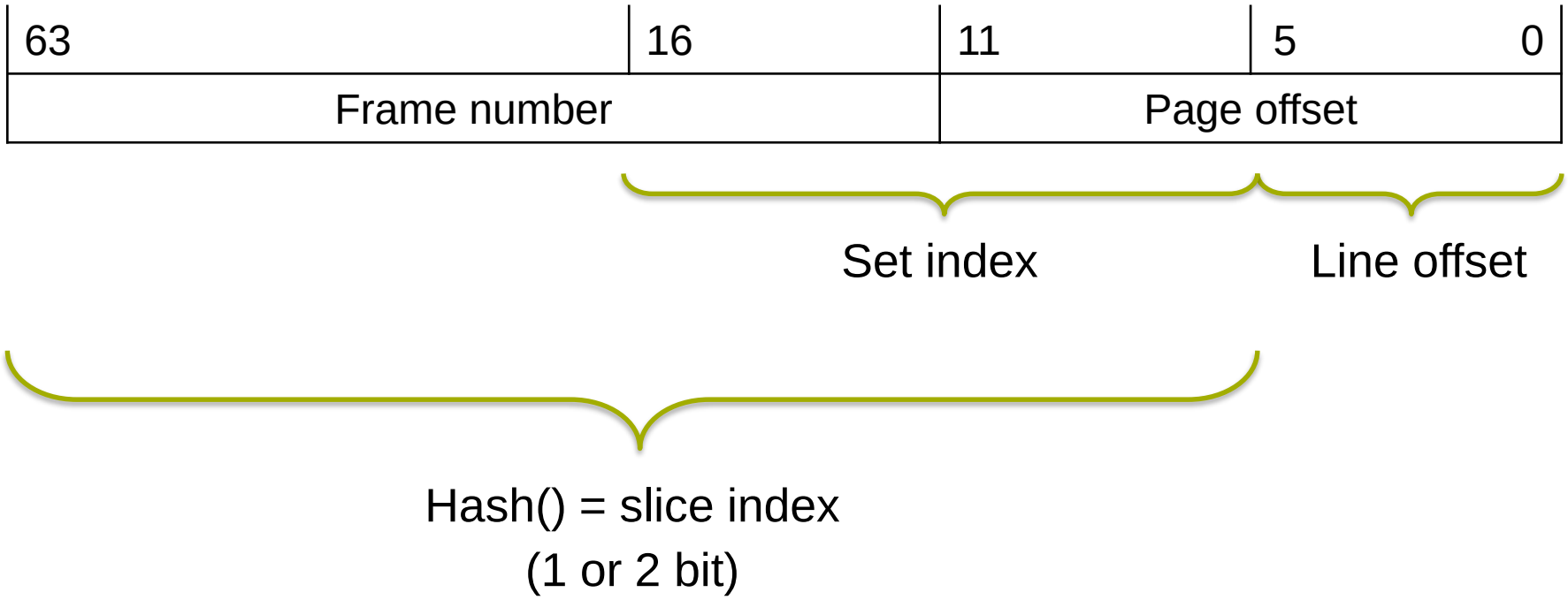


- Line 64B
- Set n-way associative
- Slice 1 per core

Question:
How do you map the addresses to a cache set?

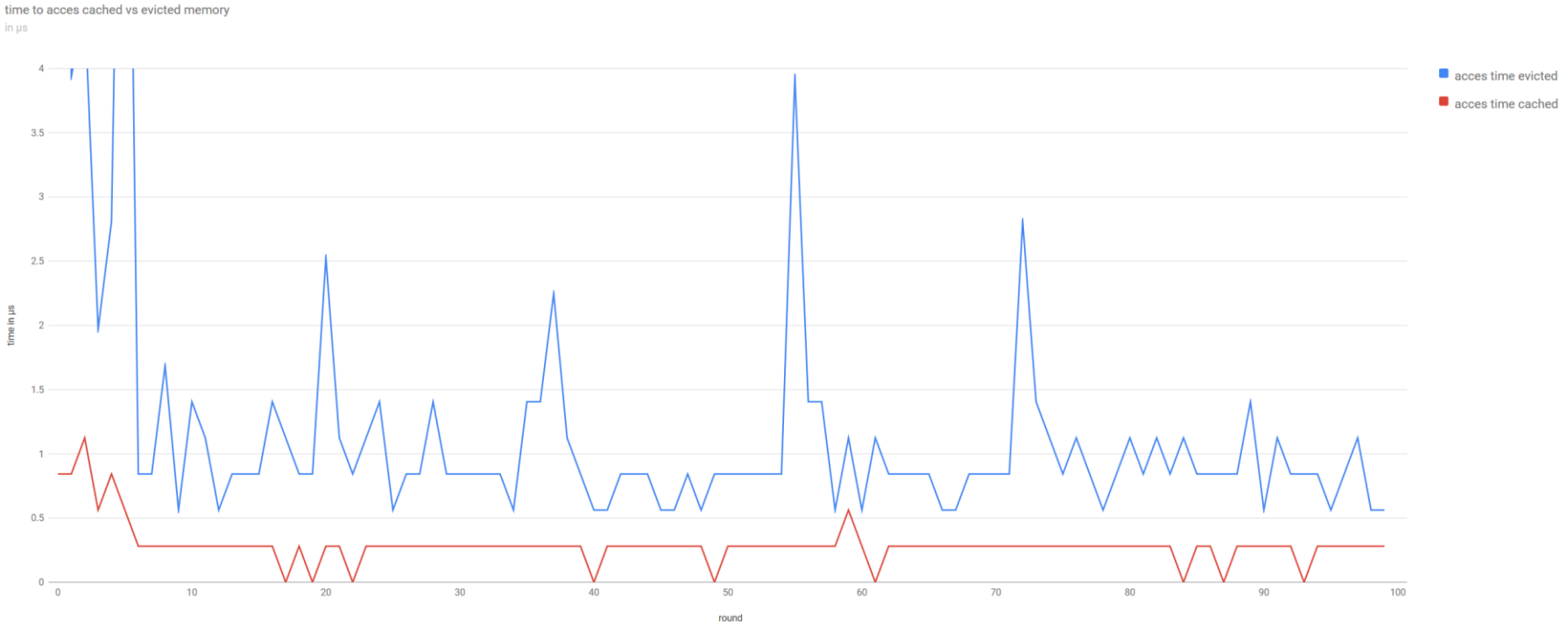
Example – Sandy Bridge

64b Physical address



Source: “Practical Timing Side Channel Attacks Against Kernel Space ASLR”, Hund et al. 2013

Cache Hierarchy



Cache Attacks

- EVICT+TIME
- FLUSH+RELOAD
- PRIME+PROBE

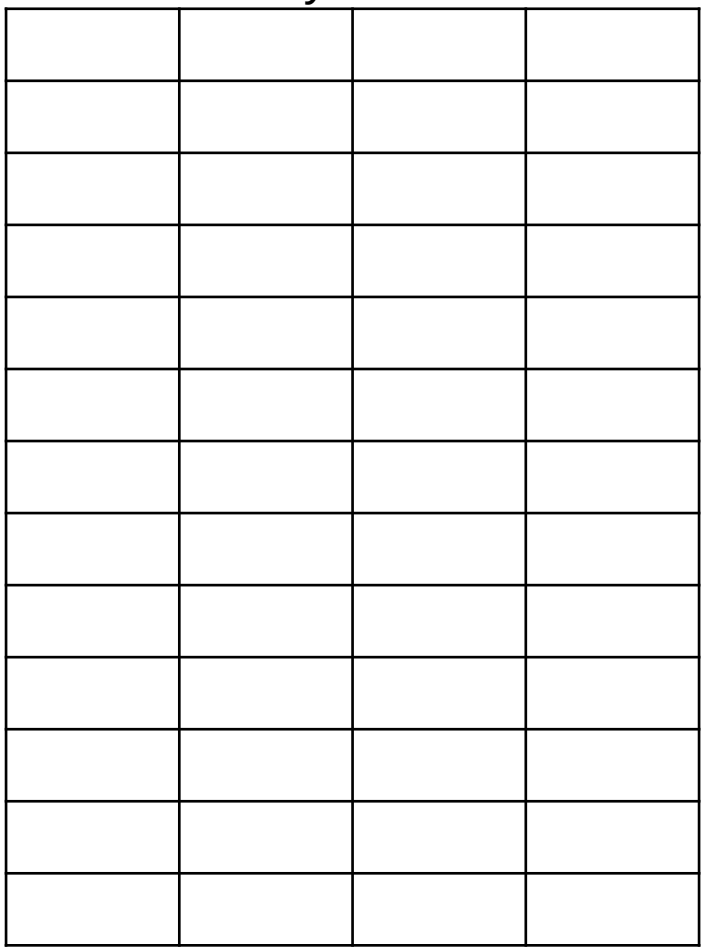
Cache Attacks – PRIME+PROBE

- Eviction set
- 4 steps
  - Create eviction set
  - Prime cache set
  - Trigger victim op
  - Probe cache set

Cache Attacks – PRIME+PROBE

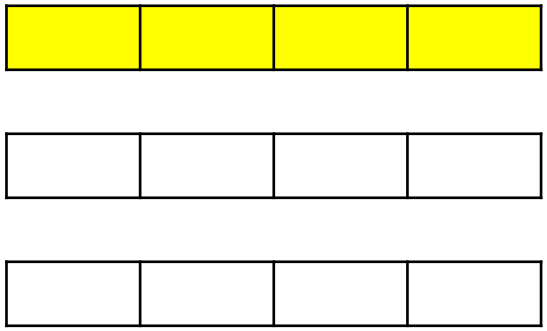
Cache Sets

Virtual Memory

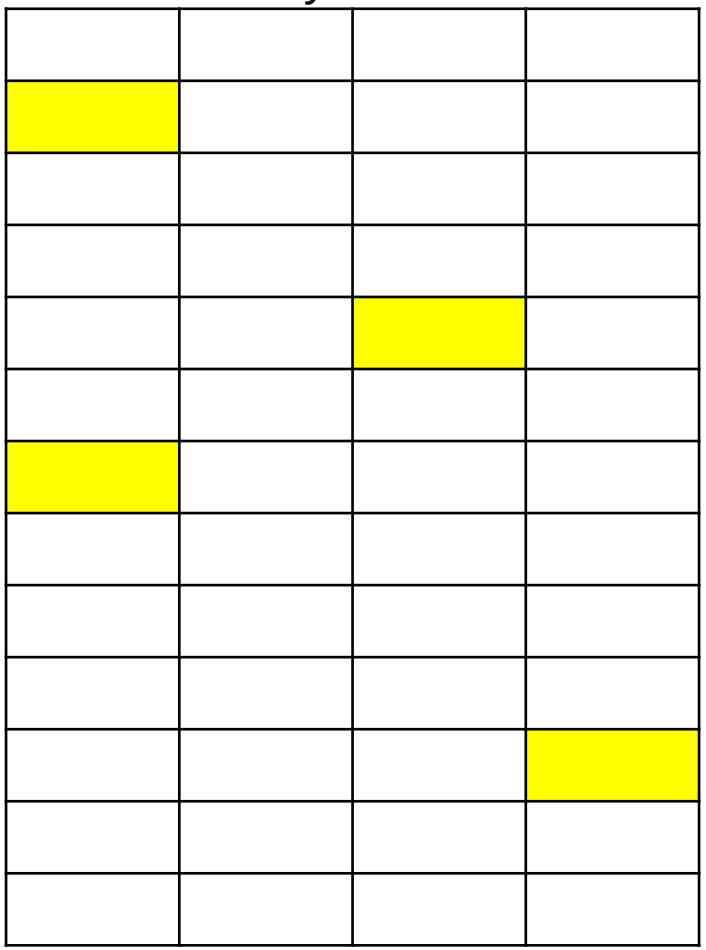


Cache Attacks – PRIME+PROBE

Cache Sets

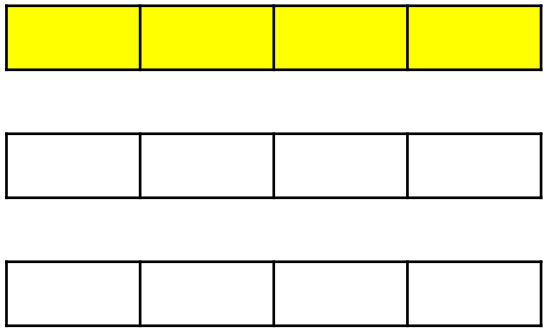


Virtual Memory

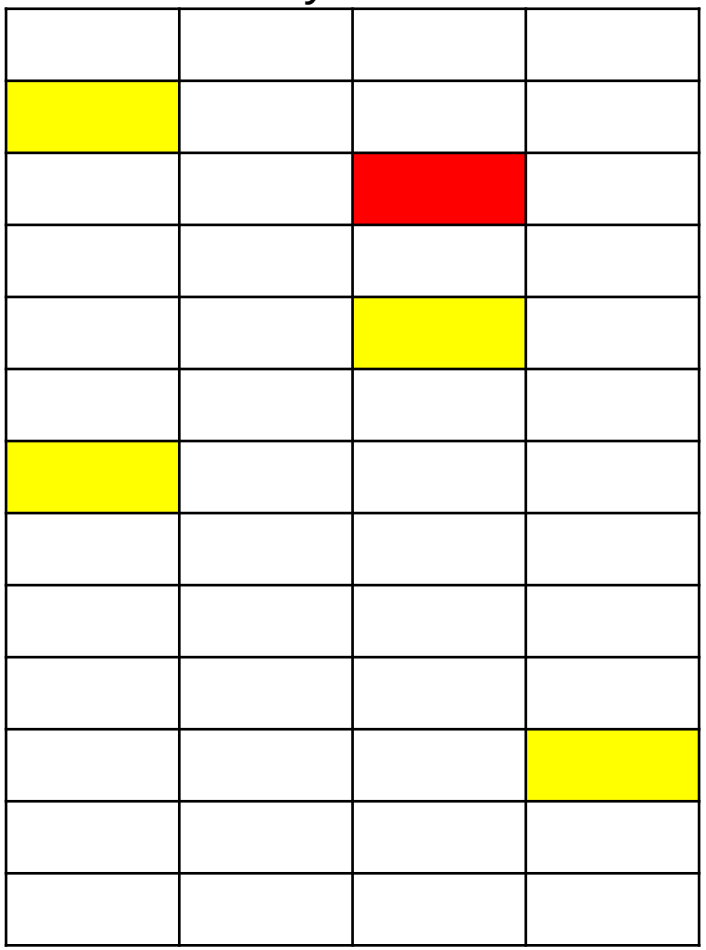


Cache Attacks – PRIME+PROBE

Cache Sets

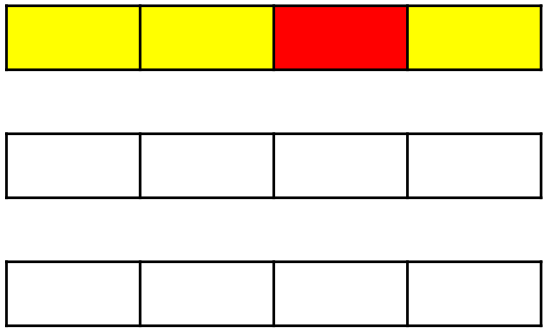


Virtual Memory

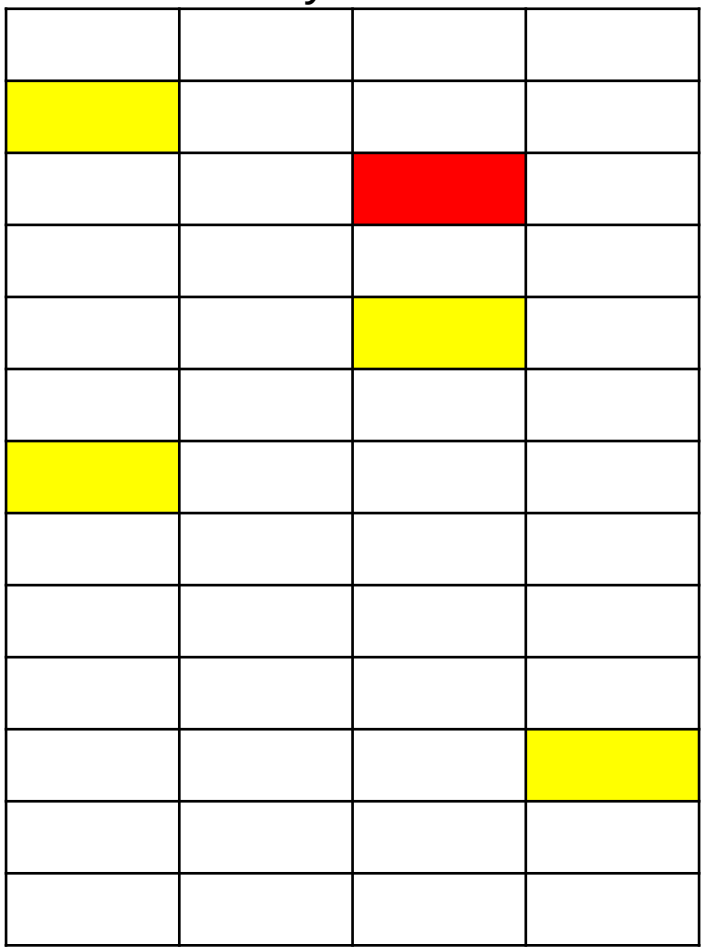


Cache Attacks – PRIME+PROBE

Cache Sets

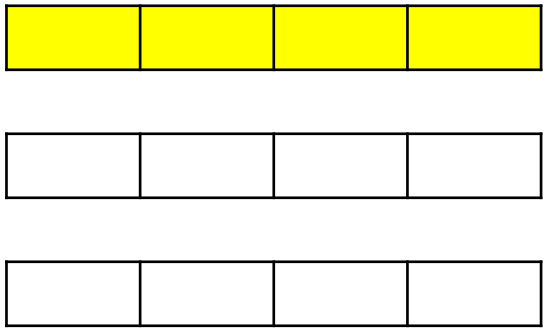


Virtual Memory

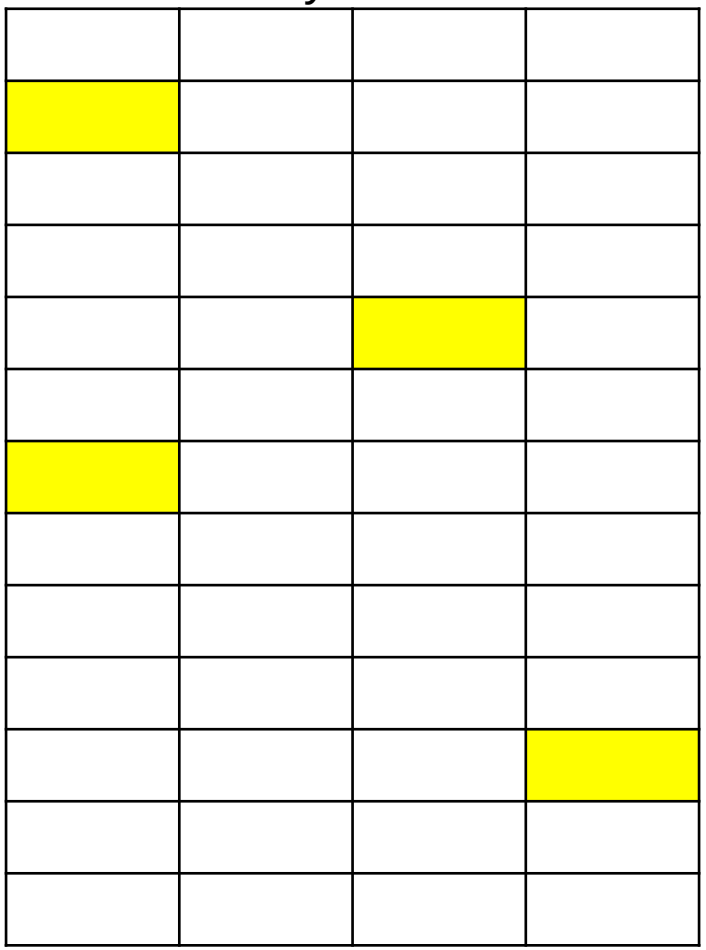


Cache Attacks – PRIME+PROBE

Cache Sets



Virtual Memory



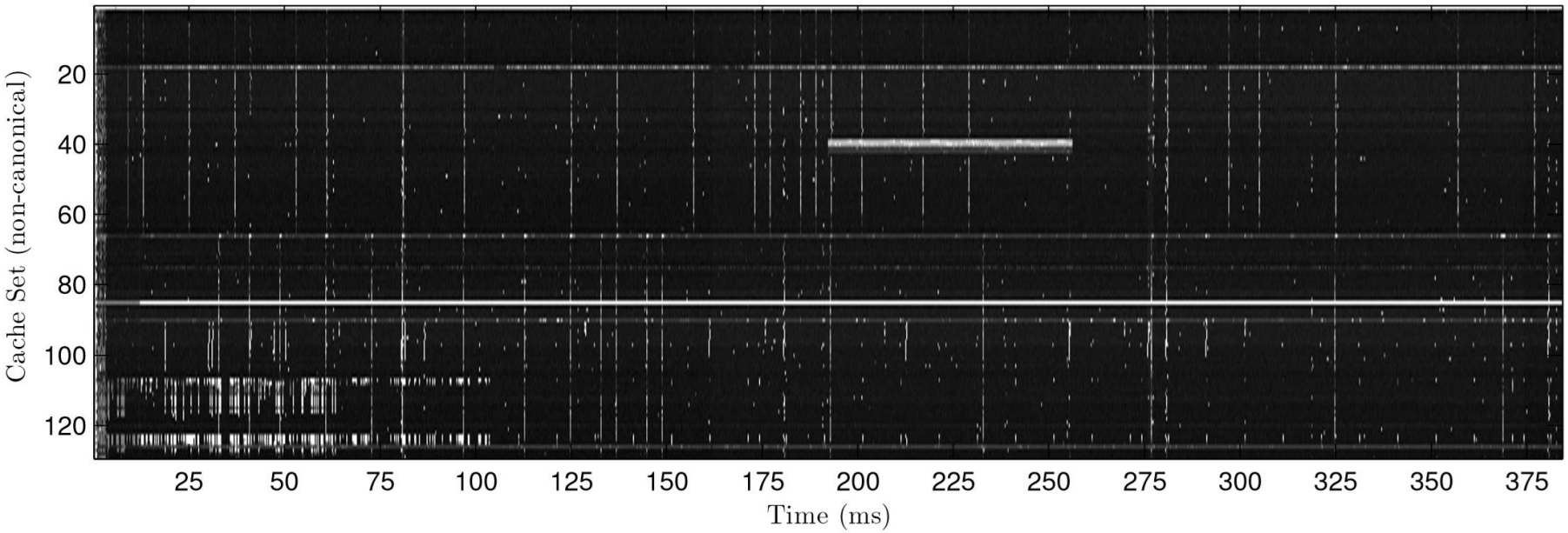
… in JavaScript

- No direct memory access
- No pointers
- No syscalls

… in JavaScript

- But:
  - High resolution Time API (nanoseconds)
  - Typed Arrays

Cache Attacks in JavaScript



Source: “The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications”, Oren et al. 2015

Cache Attacks in JavaScript

Bandwidth

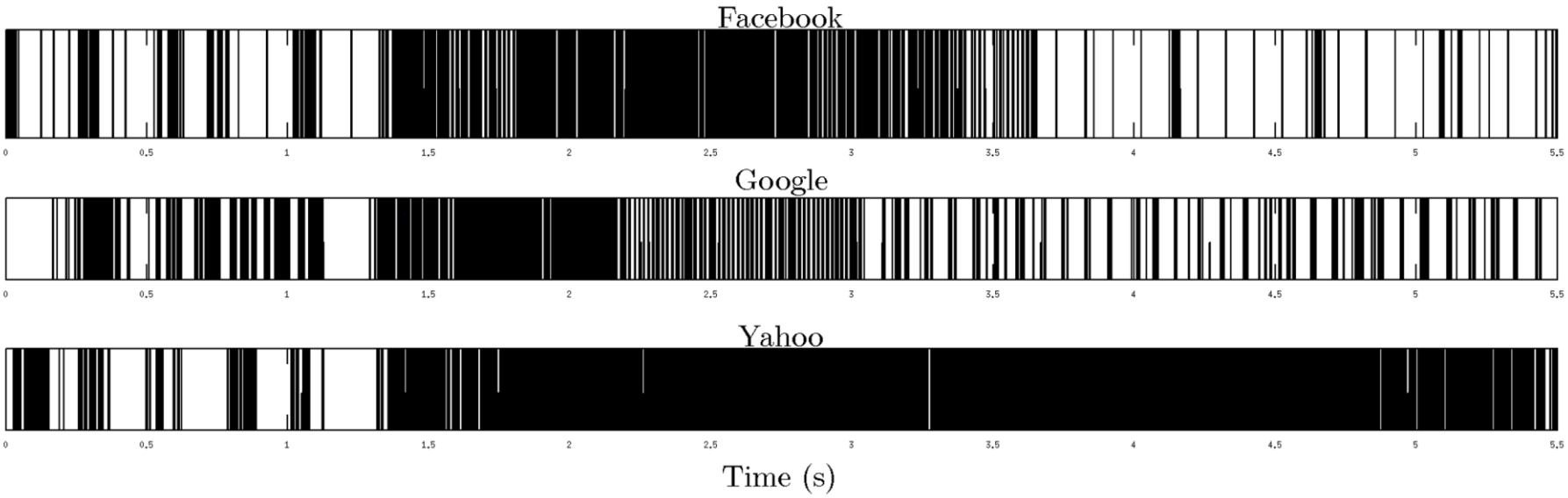
- 4KB array
- 64 cache sets -> 64 bit
- JavaScript: 320 kb/s *
- Native: 1.2 mb/s *

(*) Source: “The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications”, Oren et al. 2015

Attacking Privacy

- Track user browsing behavior
- Across separate browsers (even TOR)
- Use hardware related events (mouse, …)

Attacking Privacy



Source: “The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications”, Oren et al. 2015

Attacking Privacy

1. Find eviction sets
2. Train
3. Measure
4. Classify (mean value of Fourier Transform)

Discussion

- Accuracy:
  - 82.1% for Safari *
  - 88.6% for Tor *
- Noise

(*)Source: “The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications”, Oren et al. 2015

Discussion

- Who is affected?

- Countermeasures
  - Decreasing timer accuracy or restricting access
  - Fuzzing
  - Inclusive to exclusive
  - Changing cache design

Future Work

- Workaround for timer accuracy
- Save cache?

Conclusion

- Keep side channel attacks in mind
- do not leave unused browser tabs open
- disable JavaScript