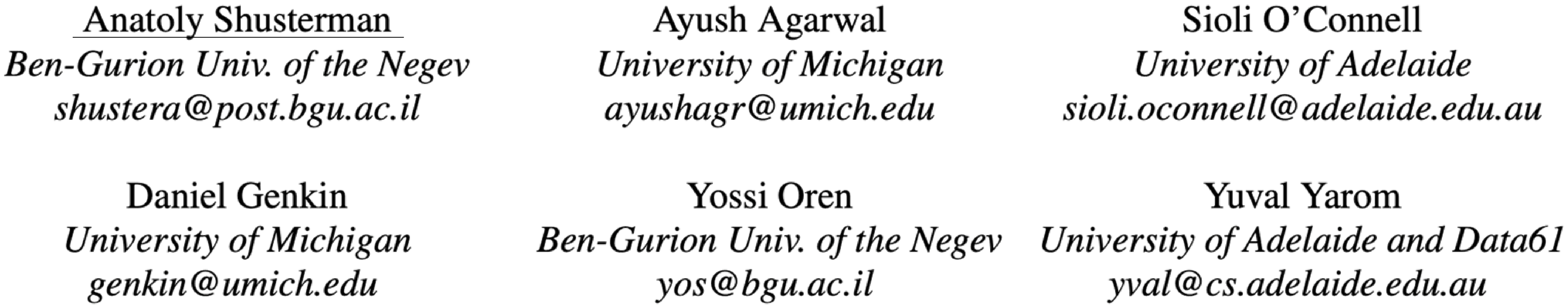
Prime+Probe 1 – JavaScript 0

Overcoming Browser-based Side-Channel Defenses



Anatoly Shusterman
Ben-Gurion Univ. of the Negev
shustera@post.bgu.ac.il

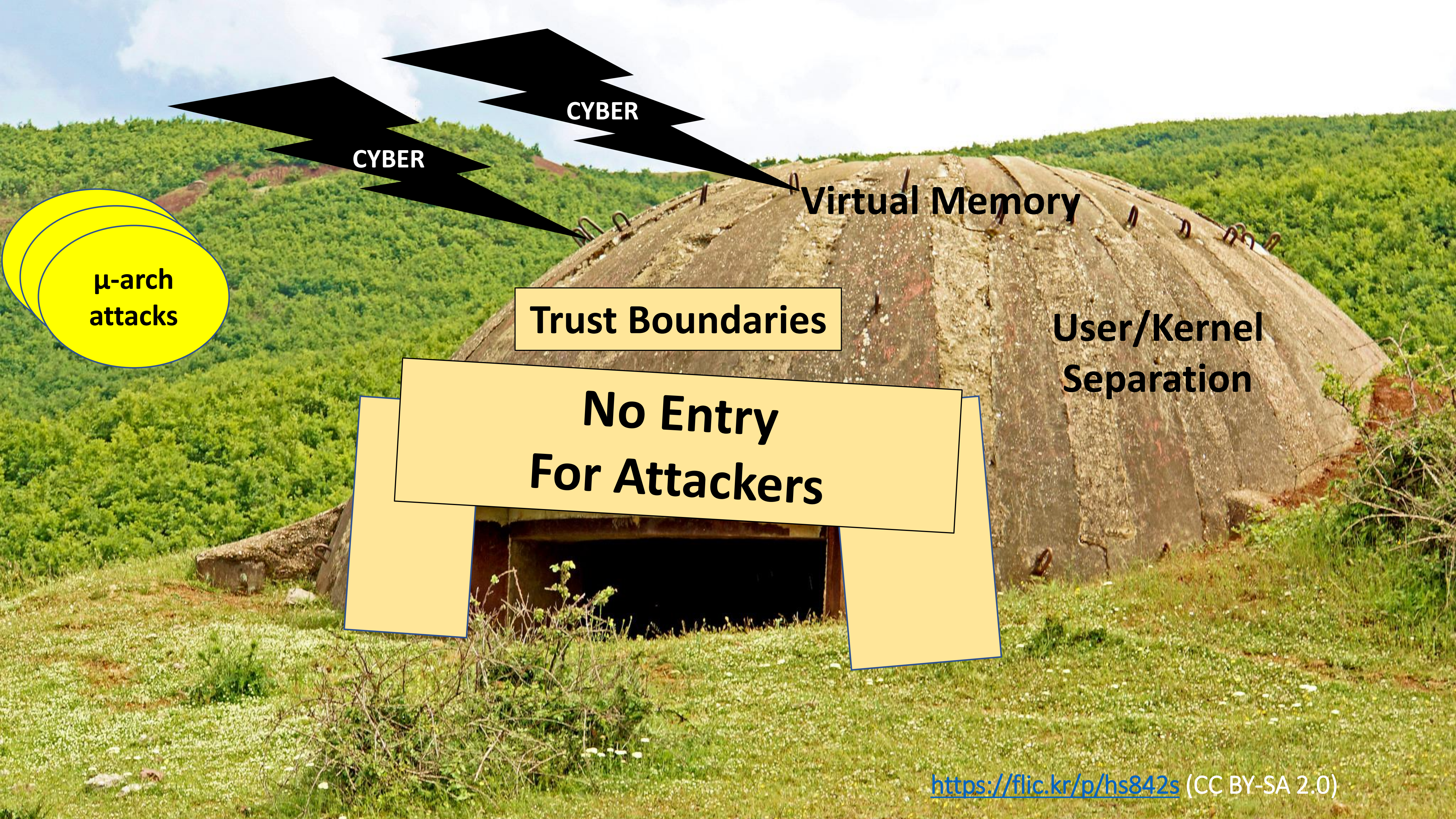
Ayush Agarwal
University of Michigan
ayushagr@umich.edu

Sioli O'Connell
University of Adelaide
sioli.oconnell@adelaide.edu.au

Daniel Genkin
University of Michigan
genkin@umich.edu

Yossi Oren
Ben-Gurion Univ. of the Negev
yos@bgu.ac.il

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au



CYBER

CYBER

μ-arch
attacks

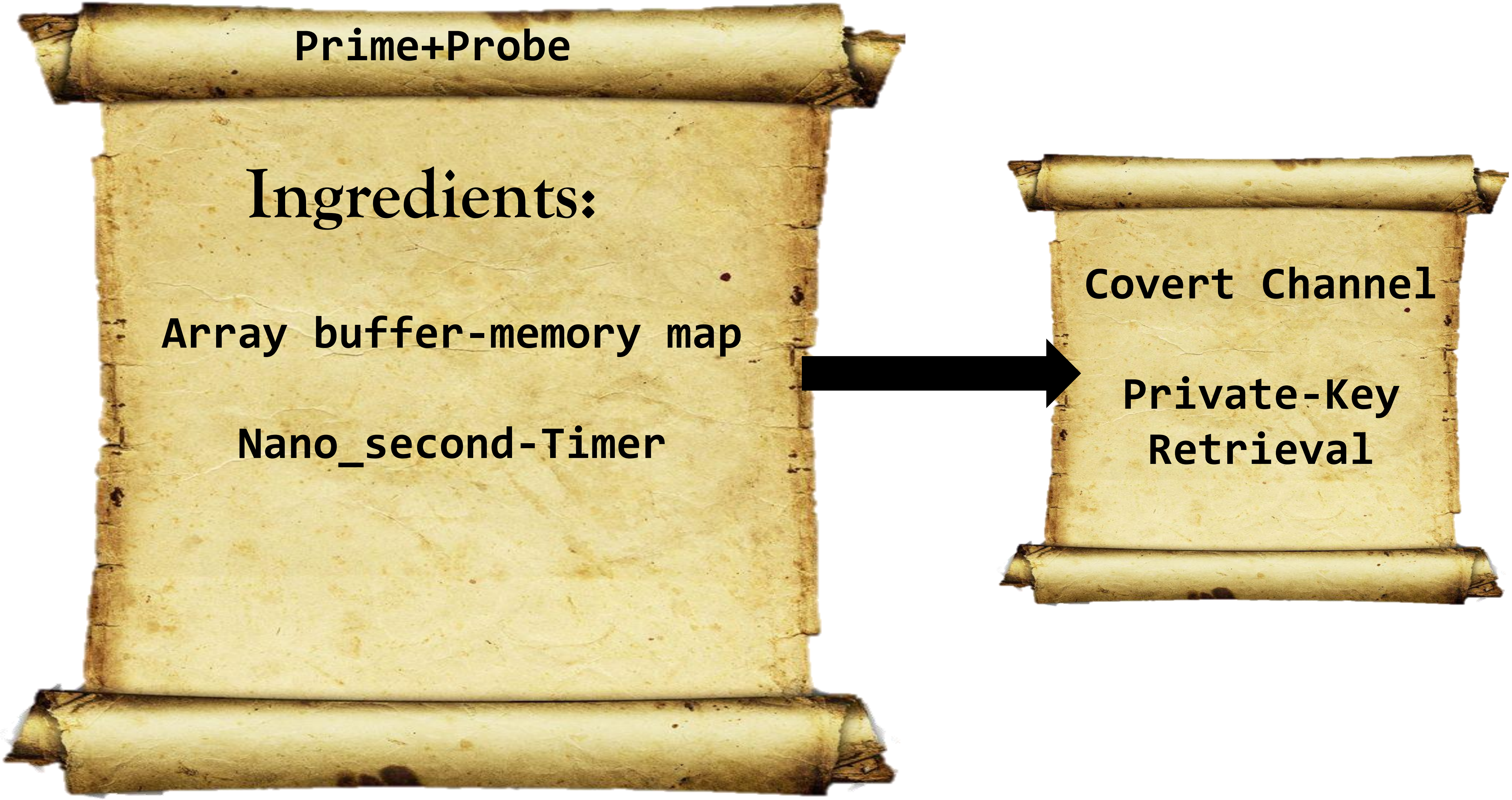
Virtual Memory

Trust Boundaries

User/Kernel
Separation

No Entry
For Attackers

https://flic.kr/p/hs842s (CC BY-SA 2.0)



Prime+Probe

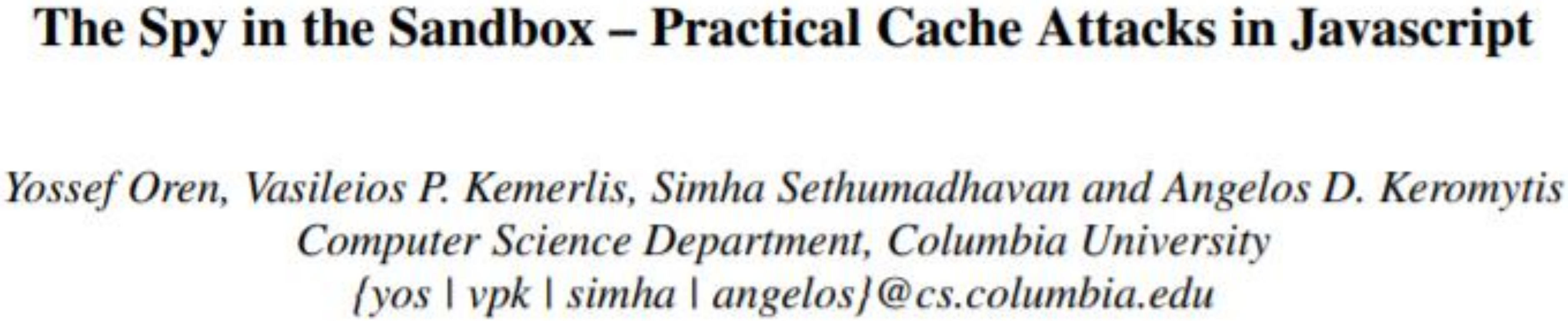
Ingredients:

Array buffer-memory map

Nano_second-Timer

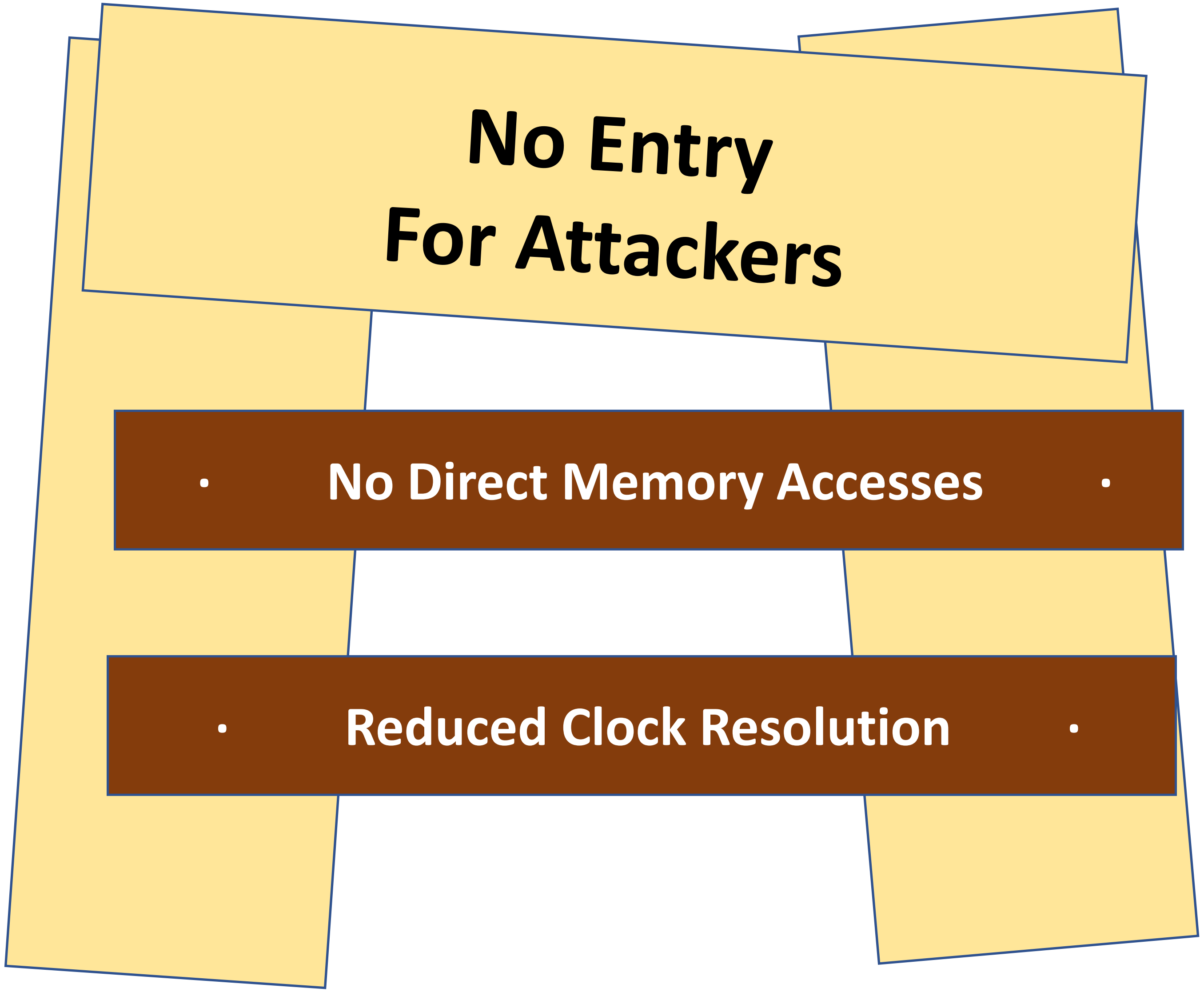
Covert Channel

Private-Key Retrieval



The Spy in the Sandbox – Practical Cache Attacks in Javascript

Yossef Oren, Vasileios P. Kemerlis, Simha Sethumadhavan and Angelos D. Keromytis
Computer Science Department, Columbia University
{yos | vpk | simha | angelos}@cs.columbia.edu

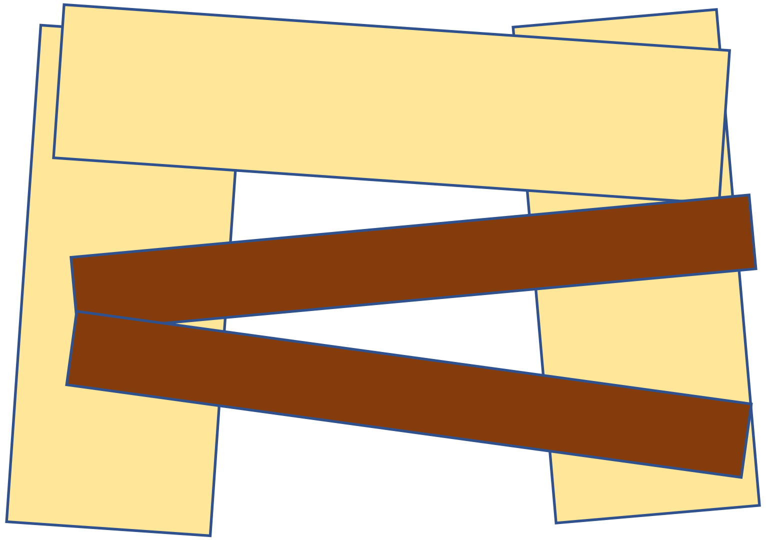


No Entry
For Attackers

- No Direct Memory Accesses
- Reduced Clock Resolution

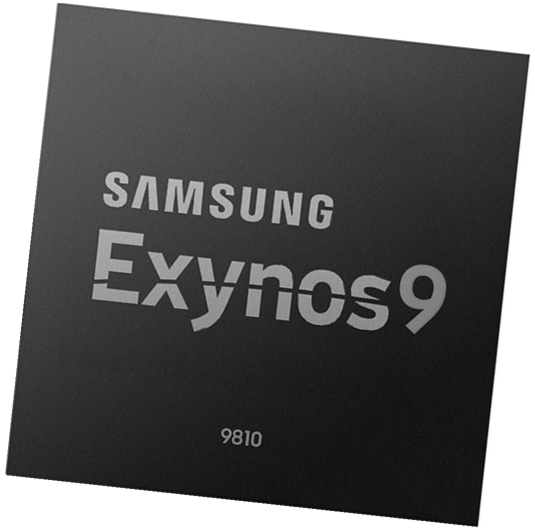
Our Research Questions

- RQ1: What are the minimal requirements for μ-architectural side-channel attacks in browsers?



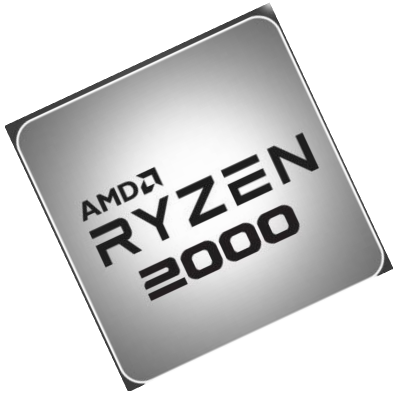
Our Research Questions

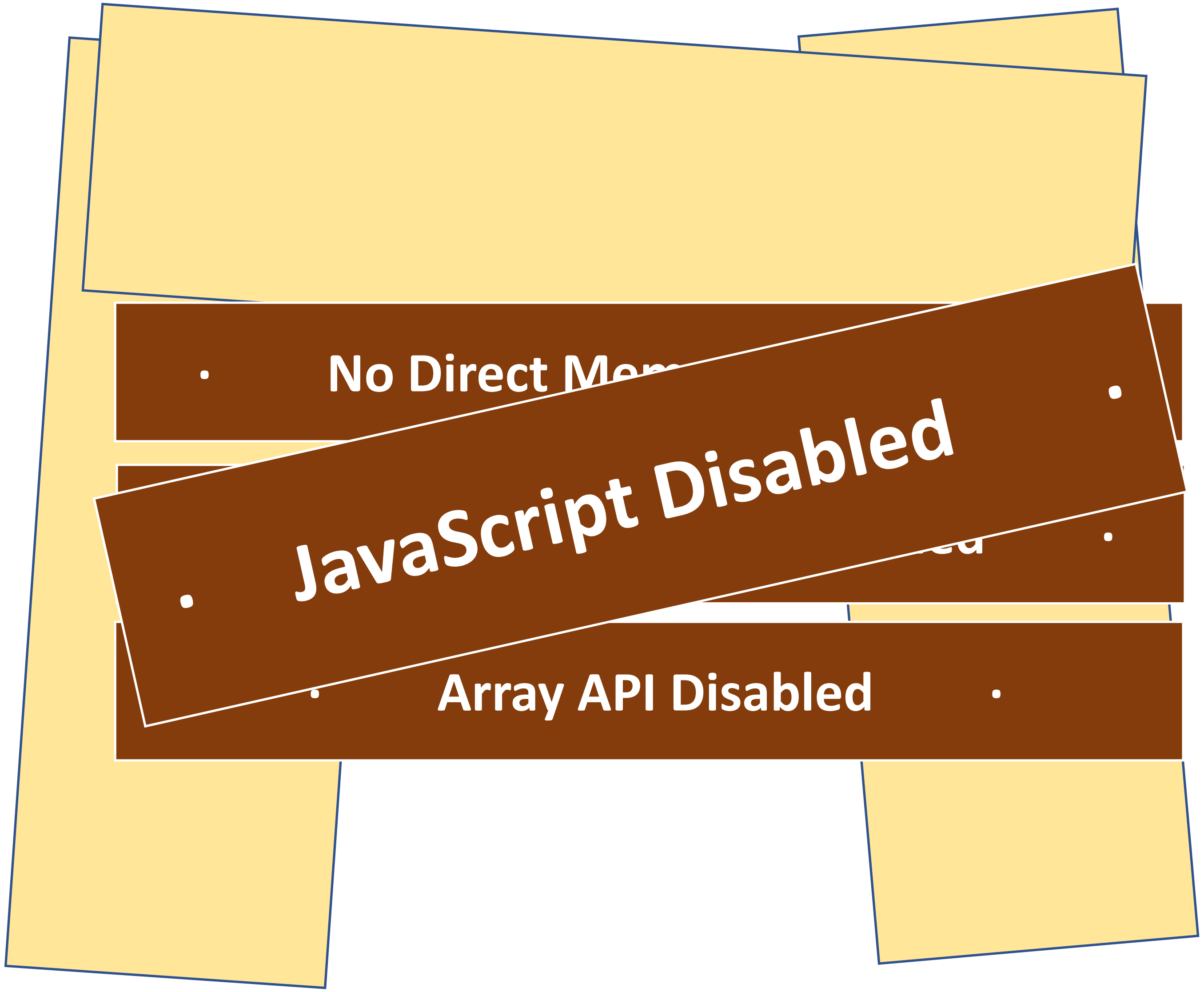
- RQ2: Can processor diversity prevent side-channel attacks?



Contributions

- RQ1: End-to-end of remote cache attacks with no timers , no arrays, and no JavaScript
- RQ2: An architecturally-agnostic attack that works on ARM, AMD, Intel and Apple M1





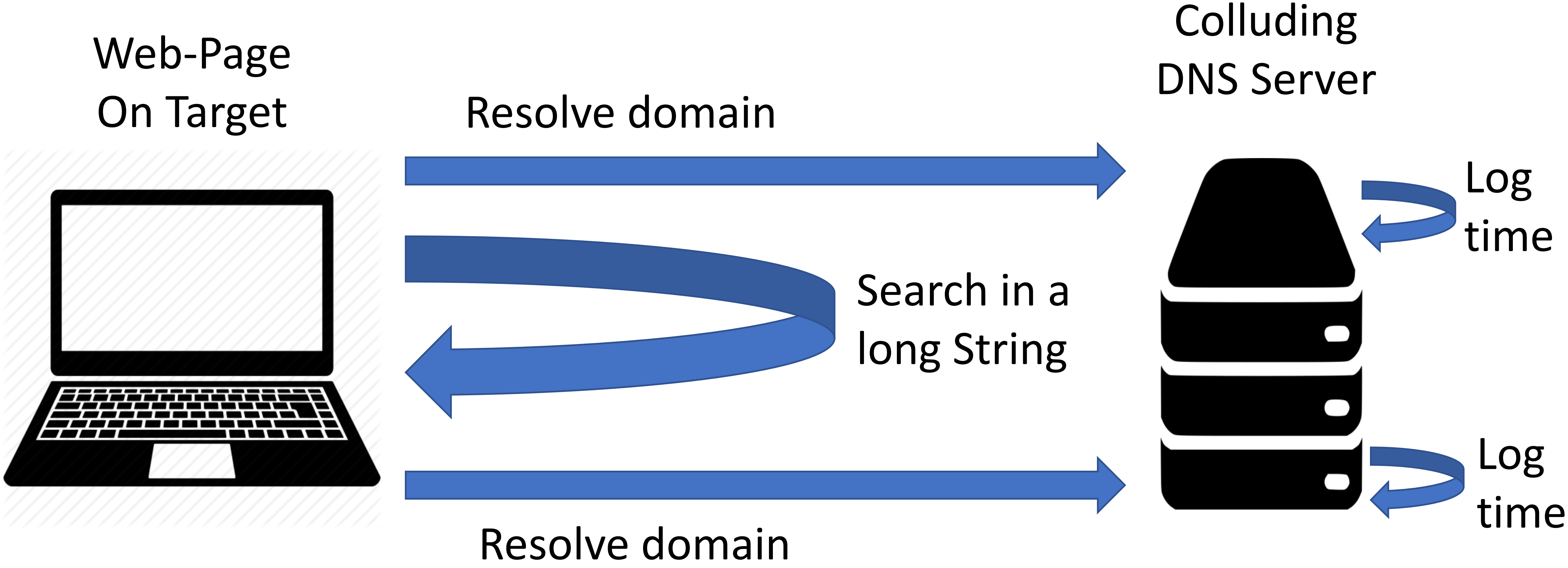
No Direct Mem

JavaScript Disabled

ed

Array API Disabled

Attack : CSS Prime+Probe [New!]



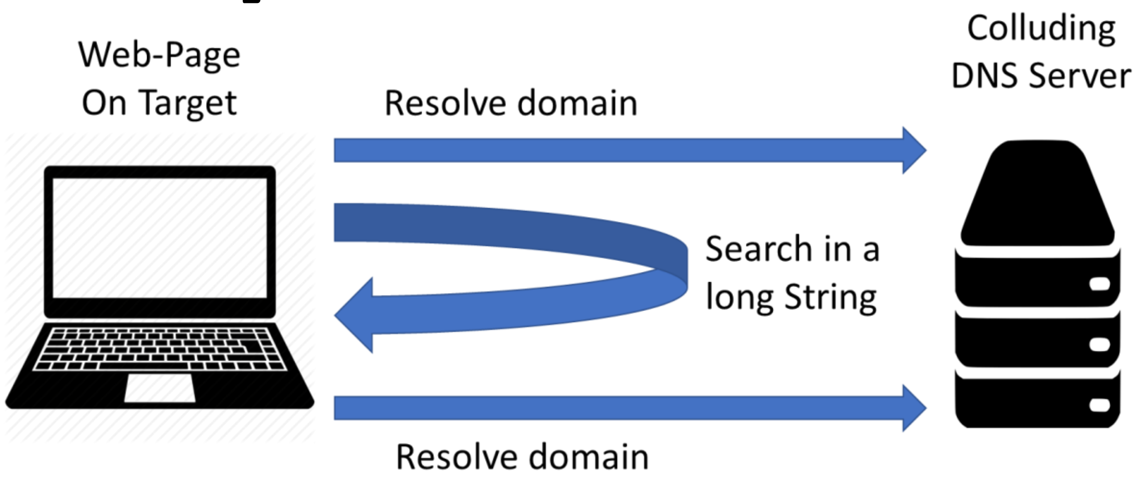
Attack : CSS Prime+Probe [New!]

```
<div id="pp" class="AAA…AAA">
  <div id="s1">X</div>
  <div id="s2">X</div>
  <div id="s3">X</div>
 .
 .
 .
</div>
```

Search non existing string

==

Probe the LLC



Resolve non existing image

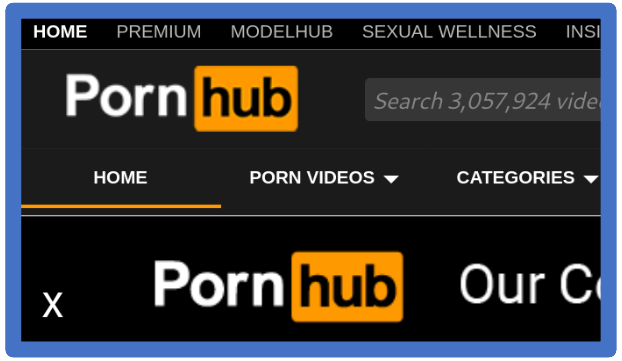
==

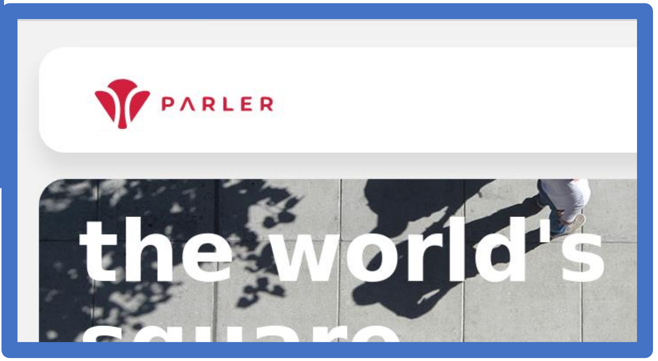
TIMER

```
#pp:not([class*= 'jigbaa']) #s1 {
  background-image: url('https://knbdsd.badserver.com');
}
#pp:not([class*= 'akhevn']) #s2 {
  background-image: url('https://pjemh7.badserver.com');
}
```

Evaluation

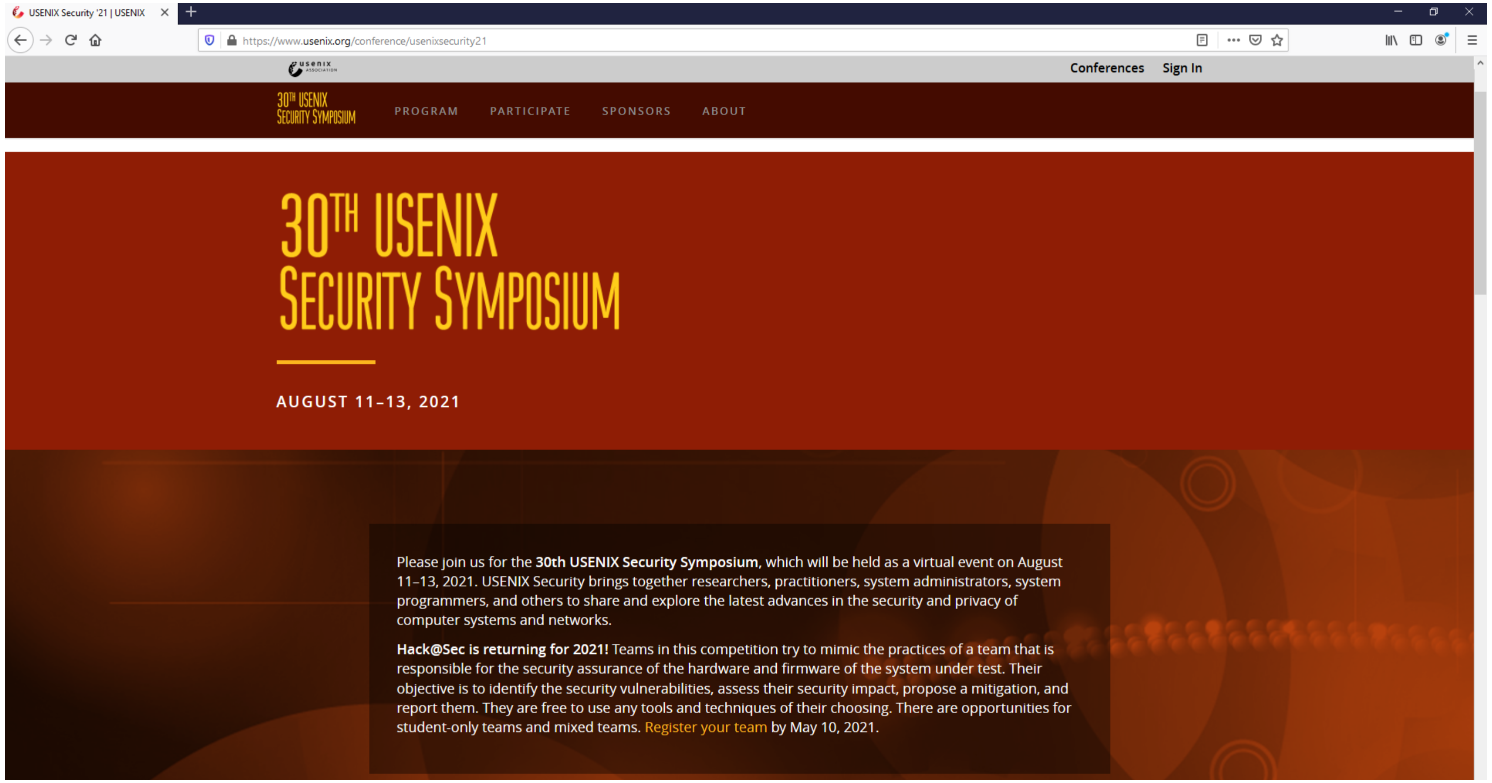
- Our method is probably not effective for cryptanalysis
- So, what is it good for?





Website Fingerprinting

Webpage Rendering



USENIX Security '21 | USENIX
https://www.usenix.org/conference/usenixsecurity21
usenix ASSOCIATION
Conferences
Sign In
30TH USENIX SECURITY SYMPOSIUM
PROGRAM
PARTICIPATE
SPONSORS
ABOUT

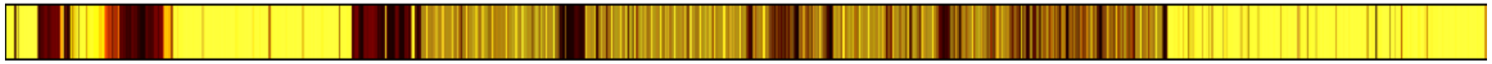
30TH USENIX SECURITY SYMPOSIUM

AUGUST 11–13, 2021

Please join us for the **30th USENIX Security Symposium**, which will be held as a virtual event on August 11–13, 2021. USENIX Security brings together researchers, practitioners, system administrators, system programmers, and others to share and explore the latest advances in the security and privacy of computer systems and networks.

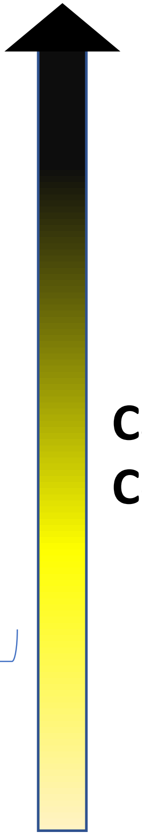
Hack@Sec is returning for 2021! Teams in this competition try to mimic the practices of a team that is responsible for the security assurance of the hardware and firmware of the system under test. Their objective is to identify the security vulnerabilities, assess their security impact, propose a mitigation, and report them. They are free to use any tools and techniques of their choosing. There are opportunities for student-only teams and mixed teams. Register your team by May 10, 2021.

https://privateurl.com

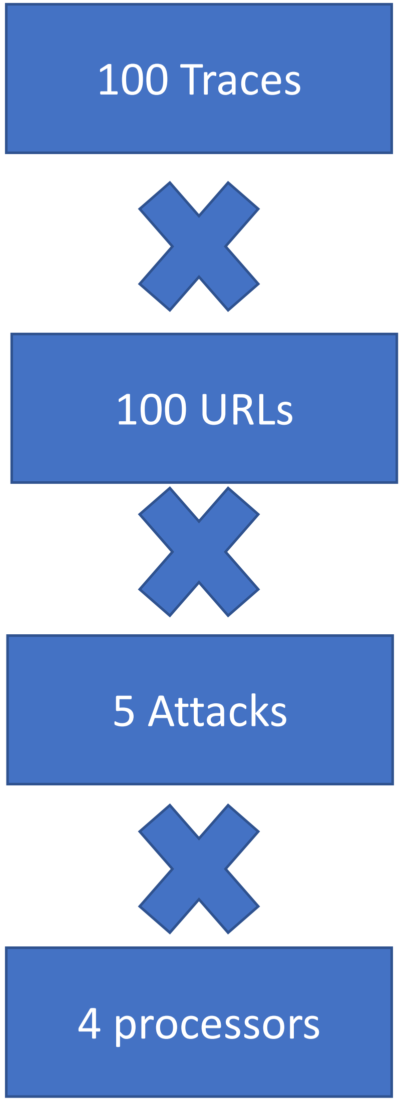


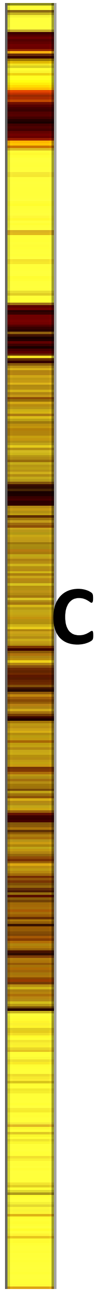
Cache Contention Measurement

Time (msec)

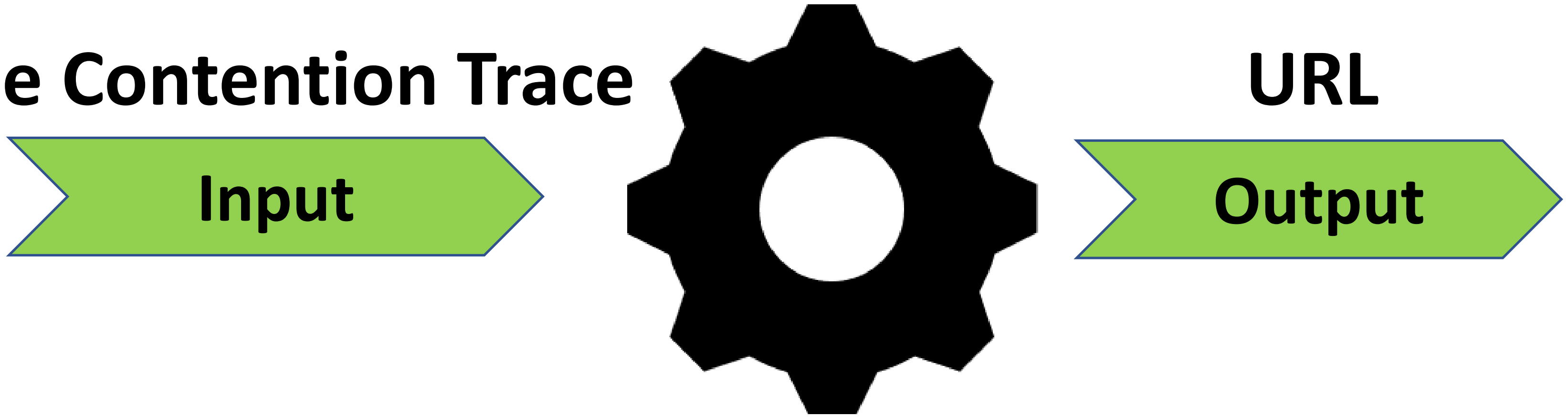


Cache Contention

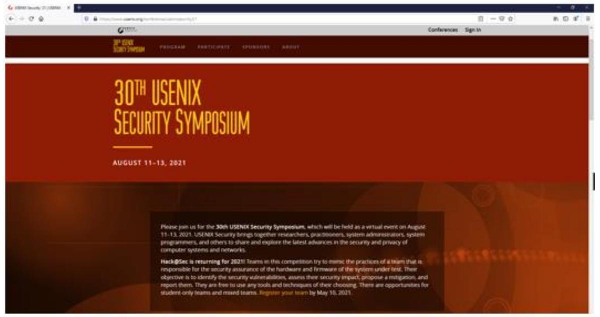




Deep Learning Models


Cache Contention Trace

URL



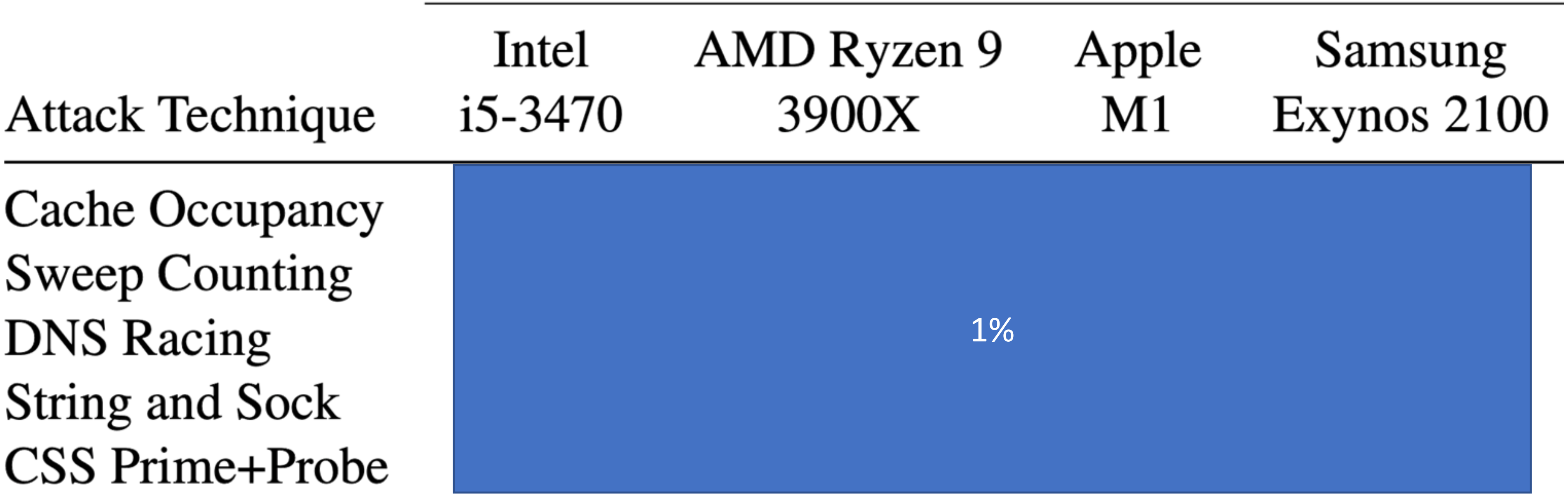
30TH USENIX SECURITY SYMPOSIUM

AUGUST 11–13, 2021

Please join us for the **30th USENIX Security Symposium**, which will be held as a virtual event on August 11–13, 2021. USENIX Security brings together researchers, practitioners, system administrators, system programmers, and others to share and explore the latest advances in the security and privacy of computer systems and networks.

Hack@Sec is returning for 2021! Teams in this competition try to mimic the practices of a team that is responsible for the security assurance of the hardware and firmware of the system under test. Their objective is to identify the security vulnerabilities, assess their security impact, propose a mitigation, and report them. They are free to use any tools and techniques of their choosing. There are opportunities for student-only teams and mixed teams. Register your team by May 10, 2021.

Results



Attack Technique	Intel i5-3470	AMD Ryzen 9 3900X	Apple M1	Samsung Exynos 2100
Cache Occupancy				
Sweep Counting				
DNS Racing				
String and Sock				
CSS Prime+Probe				

1%

Conclusion

- Restricted environments don’t prevent cache contention attacks.
- Lower attack requirements make it architectural agnostic.
- Protection against µ-architectural leaks should be applied at the source, not at the receiver

https://orenlab.sise.bgu.ac.il/p/PP0