Ultimate SLH: Taking Speculative Load Hardening to the Next Level

Zhiyuan Zhang, Gilles Barthe, Chitchanok Chuengsatiansup,

Peter Schwabe, Yuval Yarom

THE UNIVERSITY of ADELAIDE

institute IMDEA

THE UNIVERSITY OF MELBOURNE

Radboud Universiteit

Spectre-V1 Revisit

Misprediction

👿

Obtain secret

```
if (index < arrayLen) {
 👿= array[i👿x];
  y = array2[👿 4096];
}
```

Covert Channel

👿

Leak

Speculative Load Hardening (SLH)

```
mask = 0;
if (index < arrayLen) {
  r -1  ‹ = index < arrayLen ? mask :  -1
-1  = array[index] | r -1  k;
}
```

Fixed memory access under misprediction

Track speculative state

Poison memory

Limitation of SLH

- Only protect memory

- Leakages could also from ***Control Flow*** ***(CCS 2021)***

	SLH	SSLH (CCS 2021)
Memory	✓	✓
Control Flow	✗	✓

Leakage Model

- Constant-time Model
  - Memory
  - Control Flow
  - Variable-time Instructions
- Bring constant-time model to speculative execution

	SLH	SSLH (CCS 2021)
Memory	✓	✓
Control Flow	✗	✓
Variable-time	✗	✗

Really Leak?

Timing variable-time instructions

```
value = sqrtsd(value);
value = mulsd(value, value);
```

On i7-10710U:

ecuting a pair of SQRTSD and MULSD:

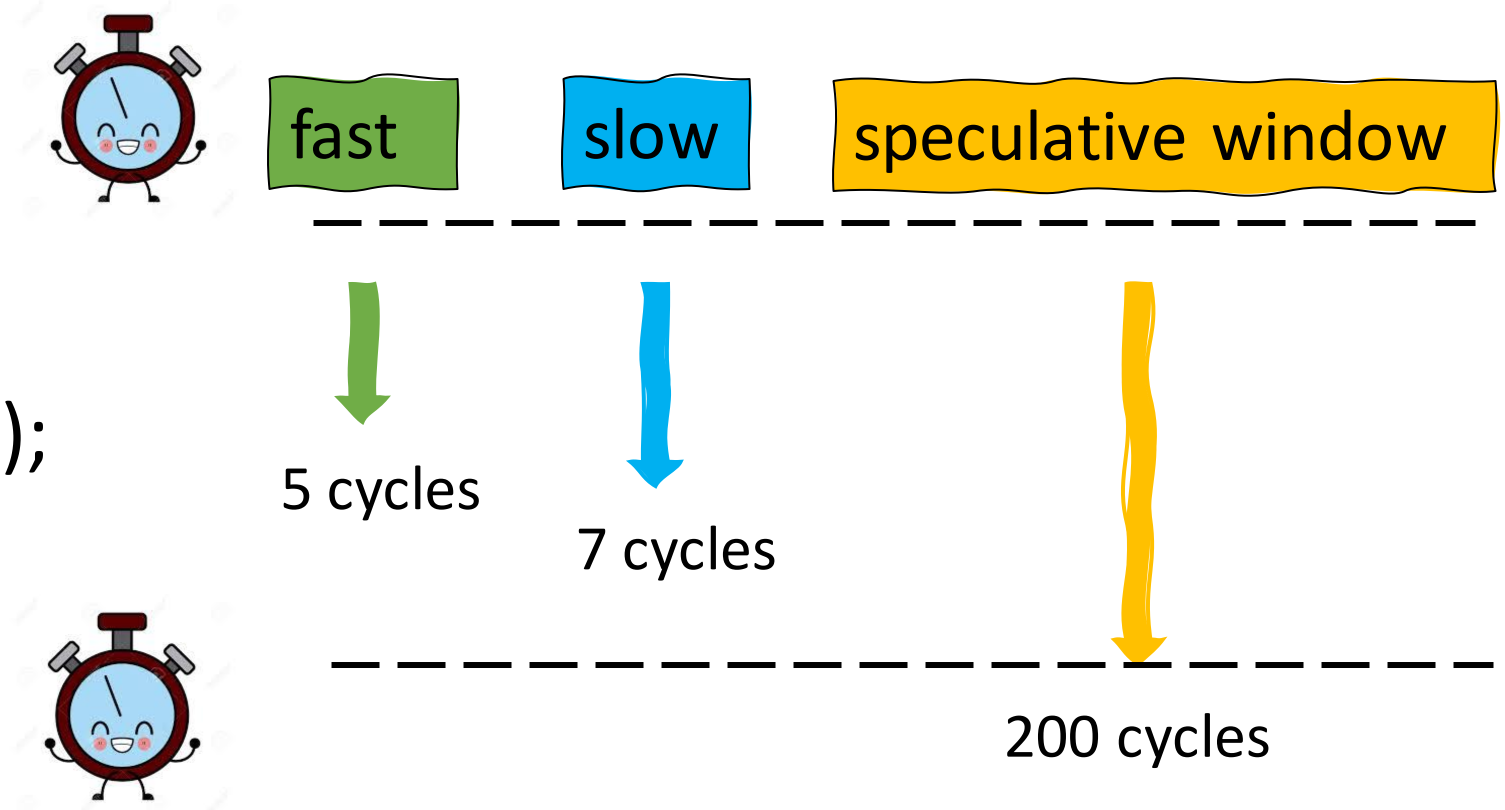
- 65536: 5 cycles
- 2.34e-308: 7 cycles

fast

slow

Timing speculative variable-time instructions

```
if (isPublic) {
  value = sqrtsd (value);
  value = mulsd (value, value);
}
```



fast

slow

speculative window

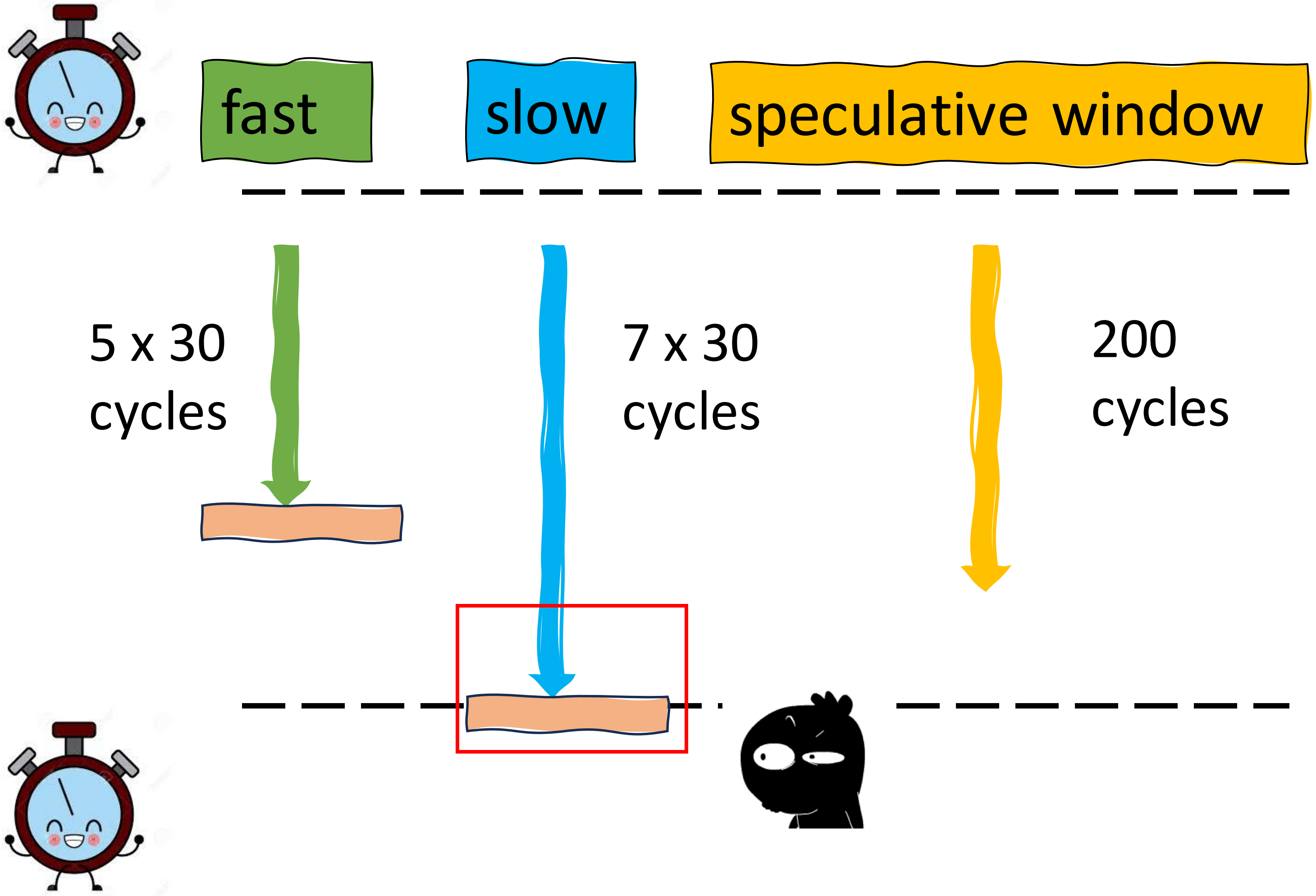
5 cycles

7 cycles

200 cycles

Timing speculative variable-time instructions

```
if (isPublic) {
  value = sqrtsd (value);
  value = mulsd (value, value);
  value = sqrtsd (value);
  value = mulsd (value, value);
  ……
}
  independent_access(x);
```



Ultimate Speculative Load Hardening

- Constant-time Model
  - Memory access (load + store)
  - Control Flow Transfers
  - Variable-time Instructions
- Bring constant-time model to speculative execution

Really Leak?

Really Leak!

	SLH	SSLH (CCS 2021)	USLH
Memory	✓	✓	✓
Control Flow	✗	✓	✓
Variable-time	✗	✗	✓

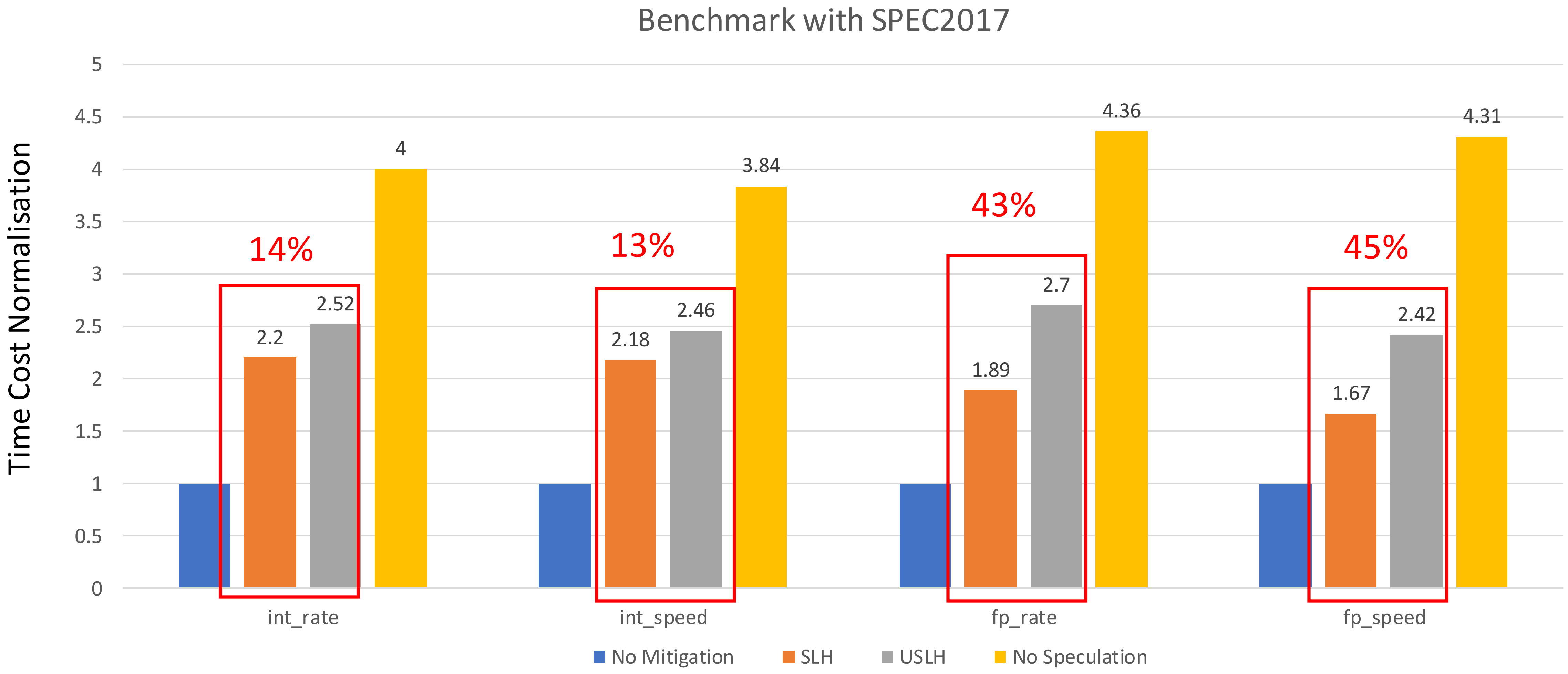
Ultimate Speculative Load Hardening

```
if (isPublic) {
  mask = isPublic ? Mask : -1;
  value = value | mask;
  value = sqrtsd (value);
  value = mulsd (value, value);
  ……
  independent_access(x);
}
```

🟩 Compiled by SLH

🟦 Compiled by USLH

USLH Benchmark



Summary

- Leakages could from variable-time operations
- Implement and benchmark Ultimate SLH
- Gadget search tool
- Formal Proof

https://github.com/0xADE1A1DE/USLH

Code

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

Paper