KU LEUVEN

DEPARTMENT OF COMPUTER SCIENCE

# ShowTime: Amplifying Arbitrary CPU Timing Side Channels

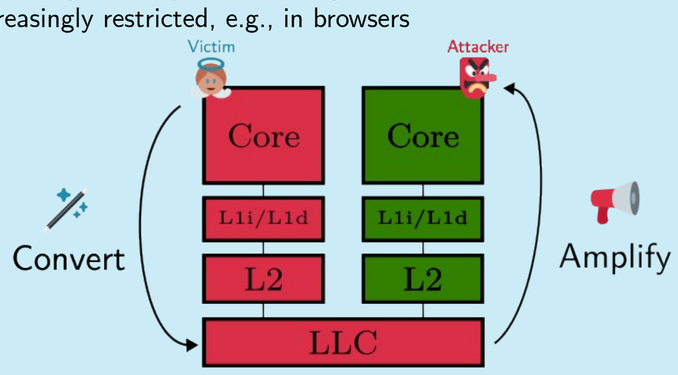
Antoon Purnal[1], Marton Bognar[2], Frank Piessens[2], Ingrid Verbauwhede[1]
[1] imec-COSIC, KU Leuven, Belgium; [2] imec-DistriNet, KU Leuven, Belgium
marton.bognar@kuleuven.be

## The ShowTime framework

**Research question: is restricting timers a good countermeasure against timing attacks?**

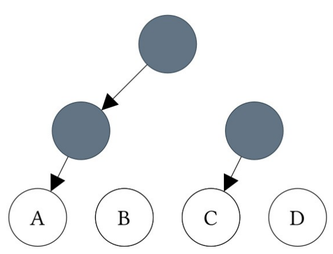
**Goal: expose secret leakage from anywhere in the CPU to coarse-grained timers**
- Realistic attacker model: cross-core, no hugepages or fixed CPU frequency
- Conditions for the leakage:
  - *Visible:* the target component is observable by the attacker
  - *Measurable:* leakage is strong enough to be measured by the attacker

ShowTime code routines:
- **Convert:** transform leakage from one microarchitectural component to another
  - Initial leakage might be stateless or local to the victim
- **Amplify:** increase the granularity of the leakage to measurable levels
  - Timers are increasingly restricted, e.g., in browsers



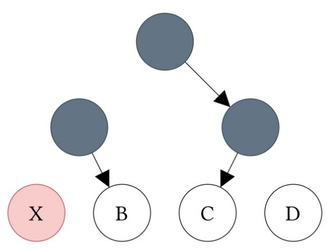

## Amplify

**L1 PLRU amplifier:**
- Based on `leaky.page`
- Exploits the replacement policy of the L1 cache
- Improved amplification rate from 1.3x to 2x
- Improved granularity from 500 us to 5 ms
- Generalized to detect reordering, back-invalidation




BABCBDBA⋯

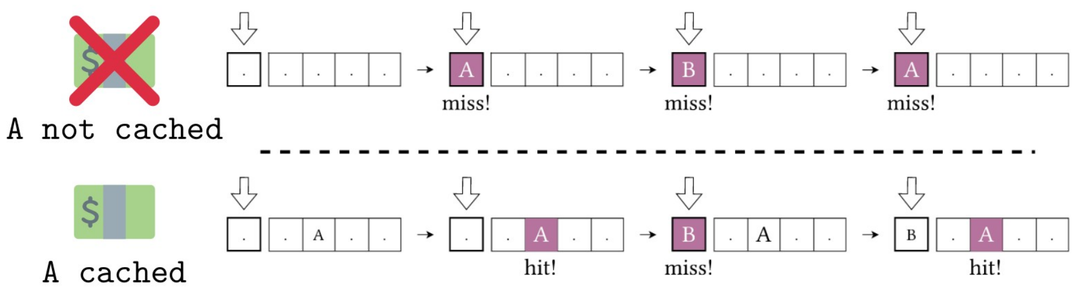
all L1 hits

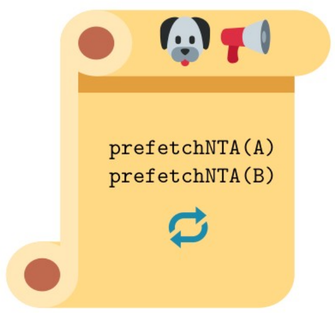



BABCBDBA⋯

many L1 misses

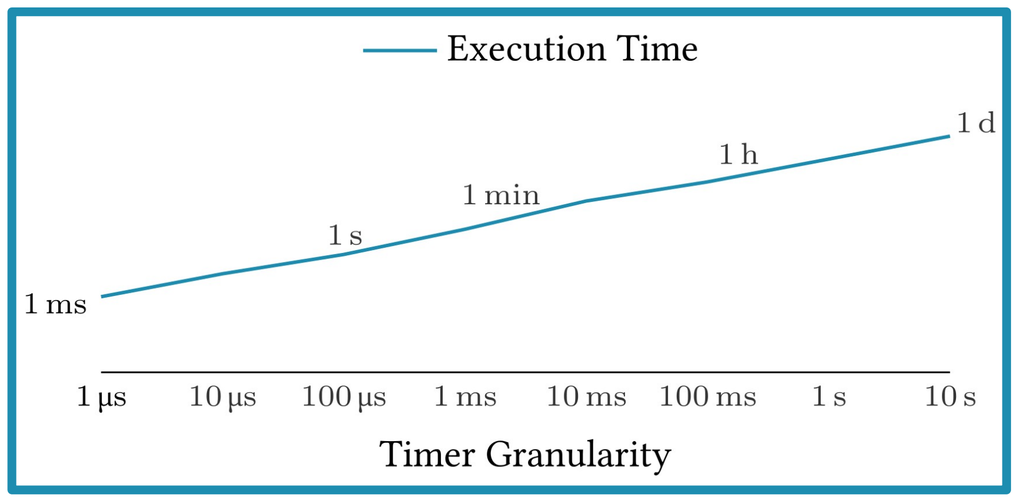
**PrefetchNTA amplifier:**
- Based on the `prefetchNTA` x86 instruction
  - Marks the loaded address as the eviction candidate in the cache
- 10x amplification rate
- 350 ms granularity




## Results at a glance

**Single-shot amplification up to seconds:**
- Human timers: classifying a cache hit or miss with the naked eye (med. 99% success rate)
- Eviction set construction using the Unix Epoch (even with 10 s granularity)



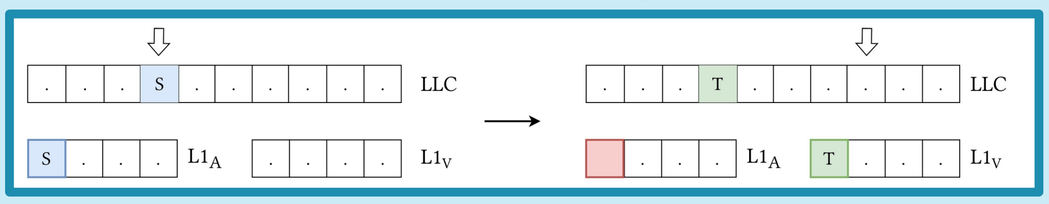

**Eviction set construction in the browser:**
- Using the default Chrome isolation settings, `performance.now()` granularity of 100 us
- Median runtime 25 seconds, successful in 70% of cases

**Measuring cross-core port contention:**
- Capturing a stateless timing difference of less than 20 ns with a coarse-grained timer

...and more!

## Convert

**LLC back-invalidation:**
- An eviction from the LLC results in an eviction in the L1/L2 caches

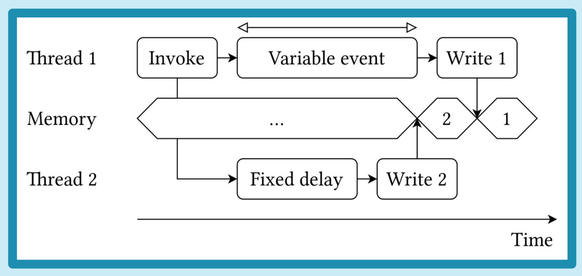



**Time to order:**
- Exploits out-of-order execution to encode a timing difference in the cache

```
                    d = evict(A)
// first leg                        // second leg
d1 = secret-delay(d)                d2 = fixed-delay(d)
d1 = load(A ^ d1)                   d2 = prefetchNTA(A ^ d2)

                    load(B ^ d1 ^ d2)
```

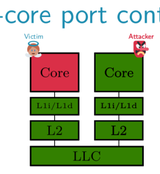
**Architectural reordering:**
- Encodes a timing difference in an architectural value through a race condition




## Further reading

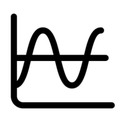
Paper: https://mici.hu/papers/purnal23showtime.pdf
GitHub: https://github.com/KULeuven-COSIC/ShowTime

Cross-core port contention

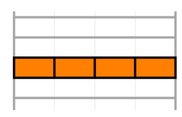



Architectural reordering

CPU frequency



Eviction set construction



KU Leuven
Department of Computer Science
Celestijnenlaan 200A box 2402
3001 Leuven, Belgium

erc DistriNet COSIC fwo