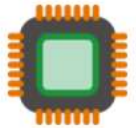


CASPER

CS773-2025-Spring: Computer Architecture for Performance and Security

Lecture 5: No Flush Only Conflicts

https://www.cse.iitb.ac.in/~biswa/

CASPER

No sharing ??

What If I do not share anything with you ??

Do not worry, I have Amazon Prime

amazon.com Prime

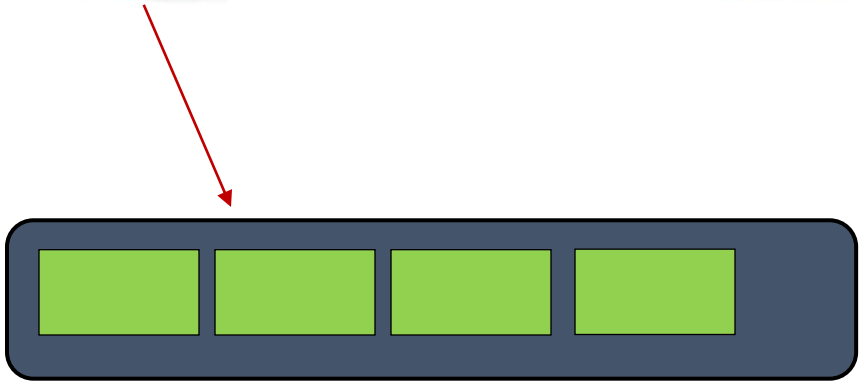
Sorry:

amazon.com Prime+Probe

CASPER

Prime + Probe

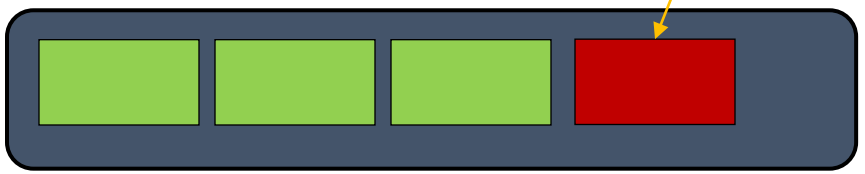
Step 0:Spy *fills* the entire shared cache



LLC

CASPER

Prime + Probe



LLC

CASPER

Step 0:Spy *fills* the entire shared cache

Step 1: Victim *evicts* cache blocks while running

Prime + Probe

LLC

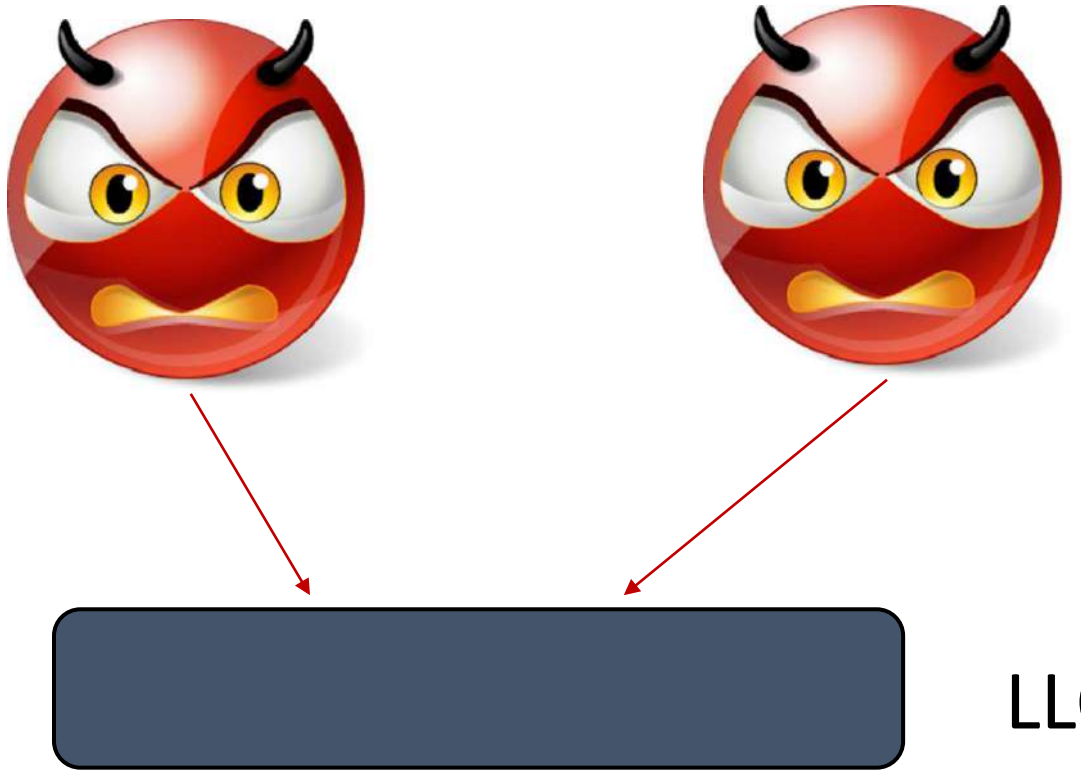
Step 0:Spy *fills* a cache set

Step 1: Victim *evicts* cache blocks while running

Step 2: Spy *probes* the cache set

If misses then victim has accessed the set

Covert Channel



LLC

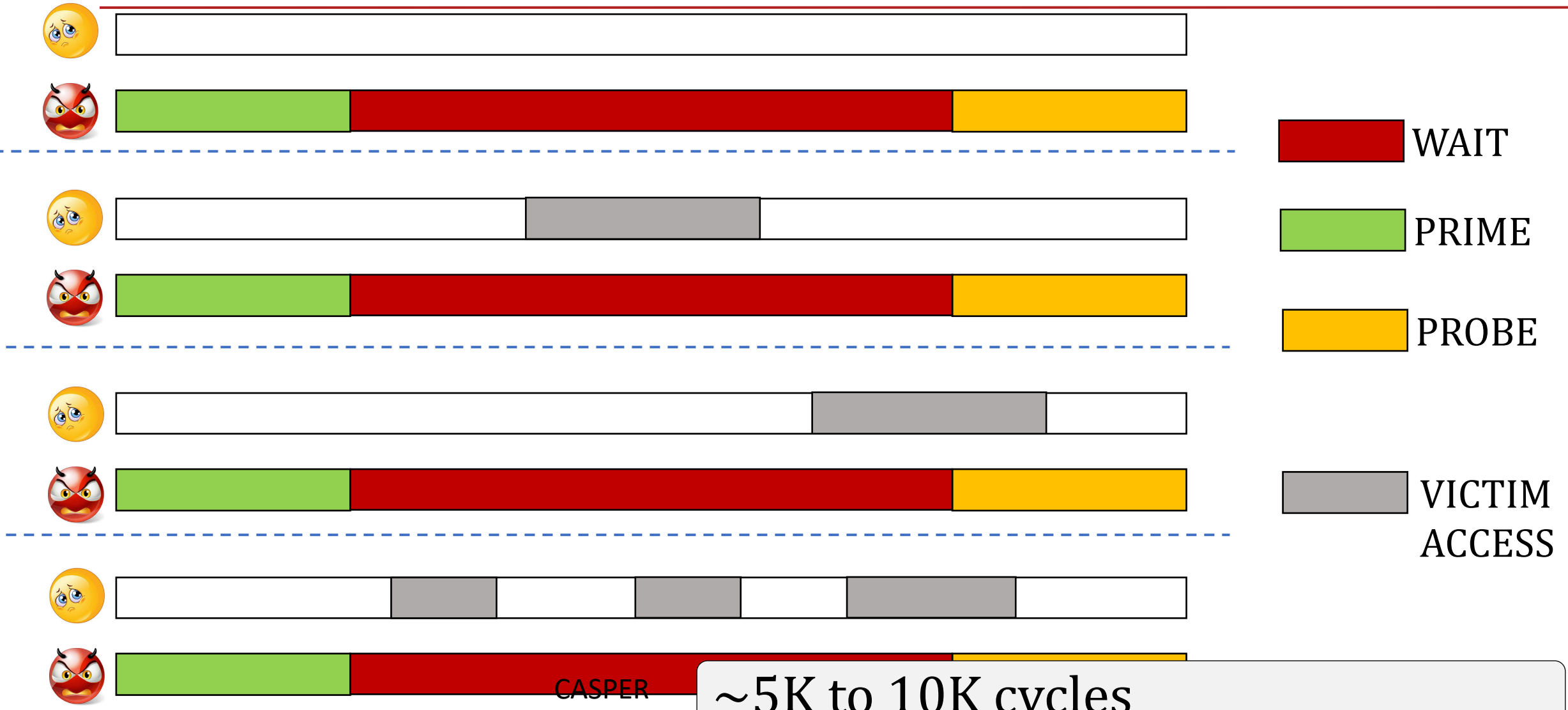
Step 0: Receiver gets data from L1 (fast, bit “0”)

Step 1: Sender thrashes LLC and back-invalidates L1

Step 2: Receiver gets data from DRAM (slow, bit “1”)

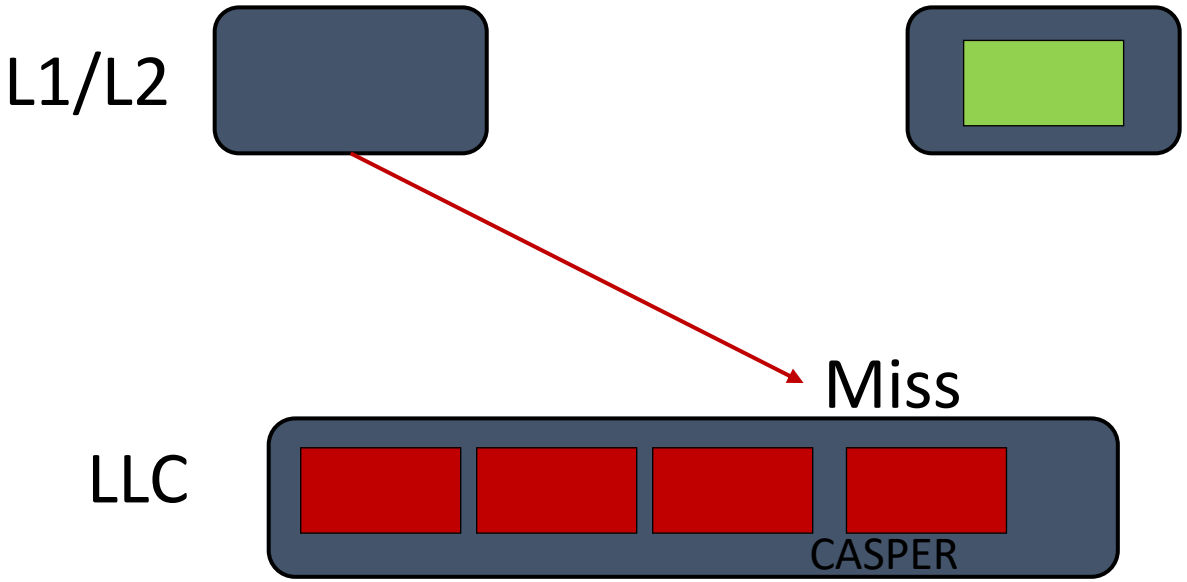
Difficult to mount in a non-inclusive cache ☹

Notion of Time Gap

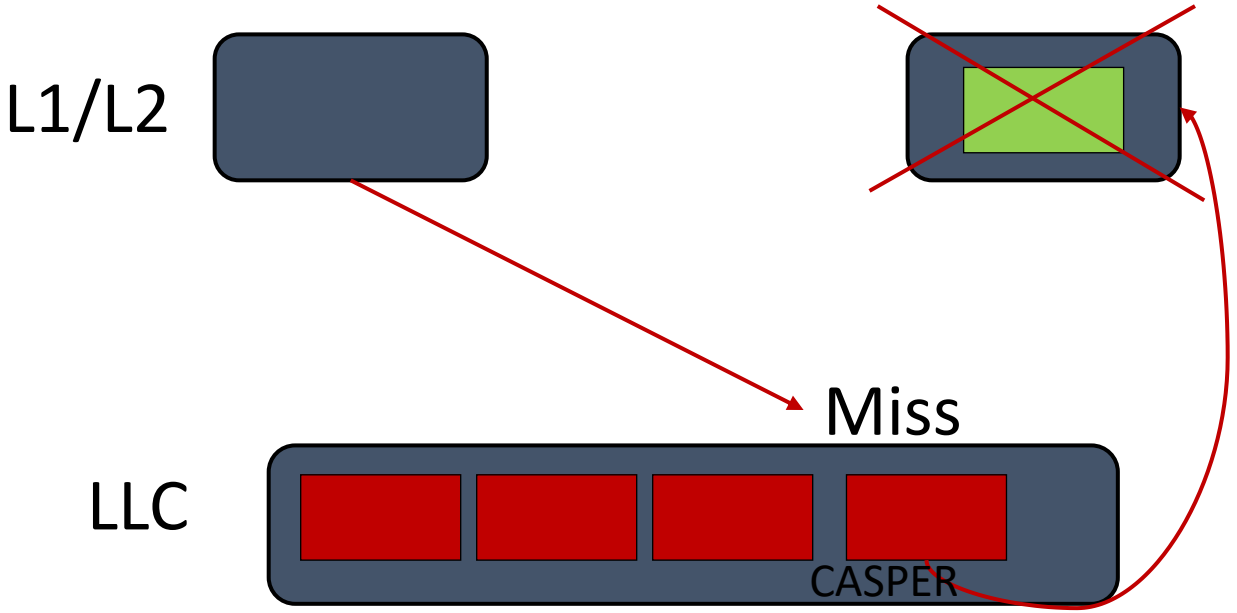


~5K to 10K cycles

Inclusiveness



Inclusiveness



Inclusiveness

Miss

L1/L2

Miss

LLC

CASPER

Attacker knows whether victim has accessed a set or not

Cache Occupancy Attack

```
int Trace[T*1000];
loop {
  counter = 0;
  t_begin = time();

  do {
    // count iterations
    counter++;
    // memory accesses
    for (i=0; i<size; i++) {
      tmp = buffer[i * 64]
    }
  } while(time()-t_begin < P);

  Trace[t_begin] = counter;
}
```

The attacker takes a parameter of period length P as input. It then constructs a trace, where each element in the trace measures how many iterations of the inner-most loop were executed every P milliseconds.

In the sweep-counting attack's code, the loop body contains an increment operation, memory accesses to a large buffer, and a call to the time() function.

Note that the buffer's size matches the size of the last-level cache so that one completion of the inner loop sweeps the entire last-level cache.

The counter value can thus be used to infer how many of the accessed cache lines reside in the cache.

What is the deal

Within a fixed window, you can have extremes of counter values.

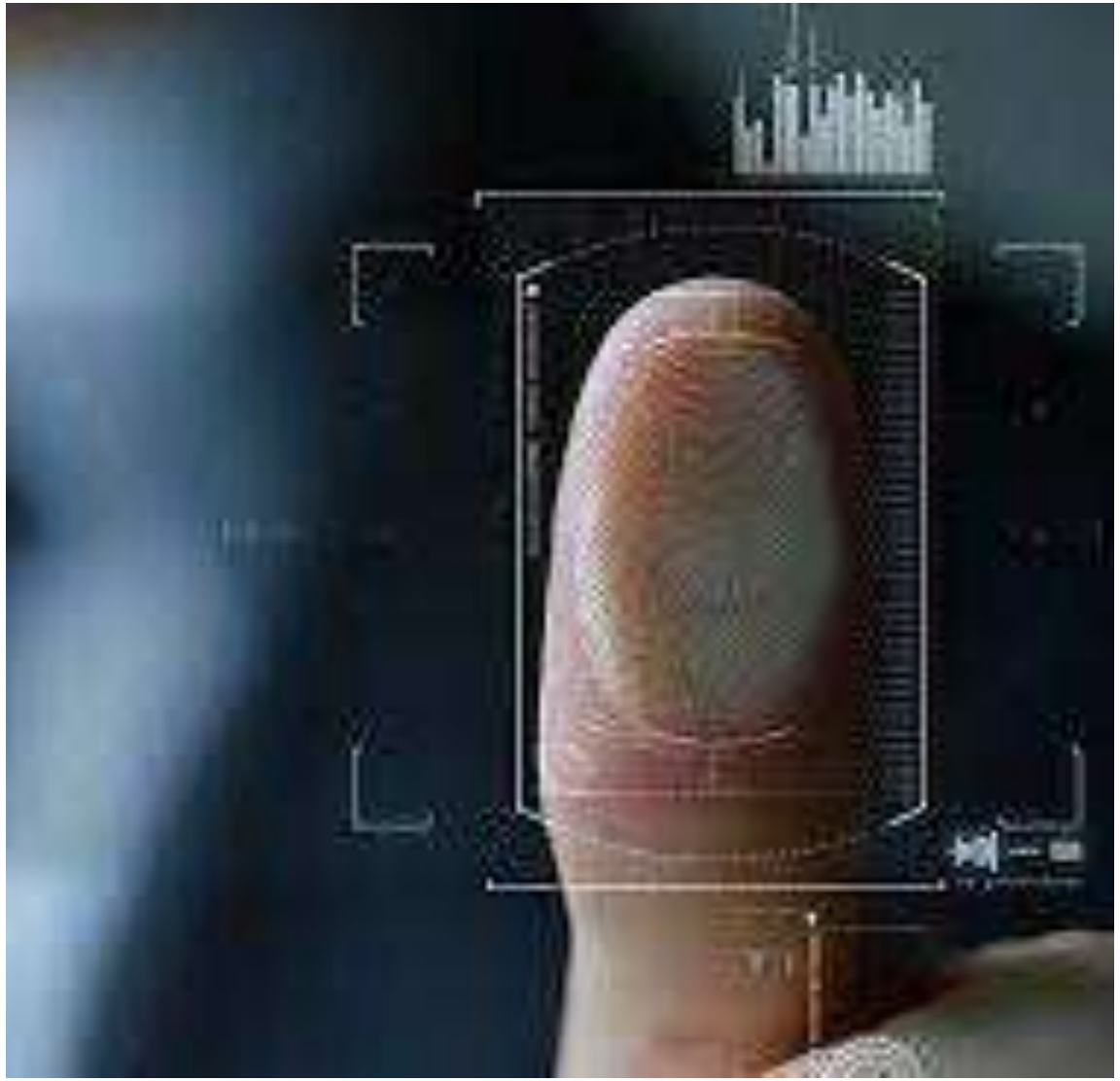
(a) If the count is small, which means someone has kicked out a large part of LLC. So, occupancy of attacker is low, and it reached the window quickly, but the counter value is small.

(b) If the count is large, which means no one else has kicked out the LLC lines, and the attacker has occupied the LLC completely. So, larger counter value.

There is mystery, please remind me in April. What is the utility? Memorygrams ☺

Physical Fingerprinting

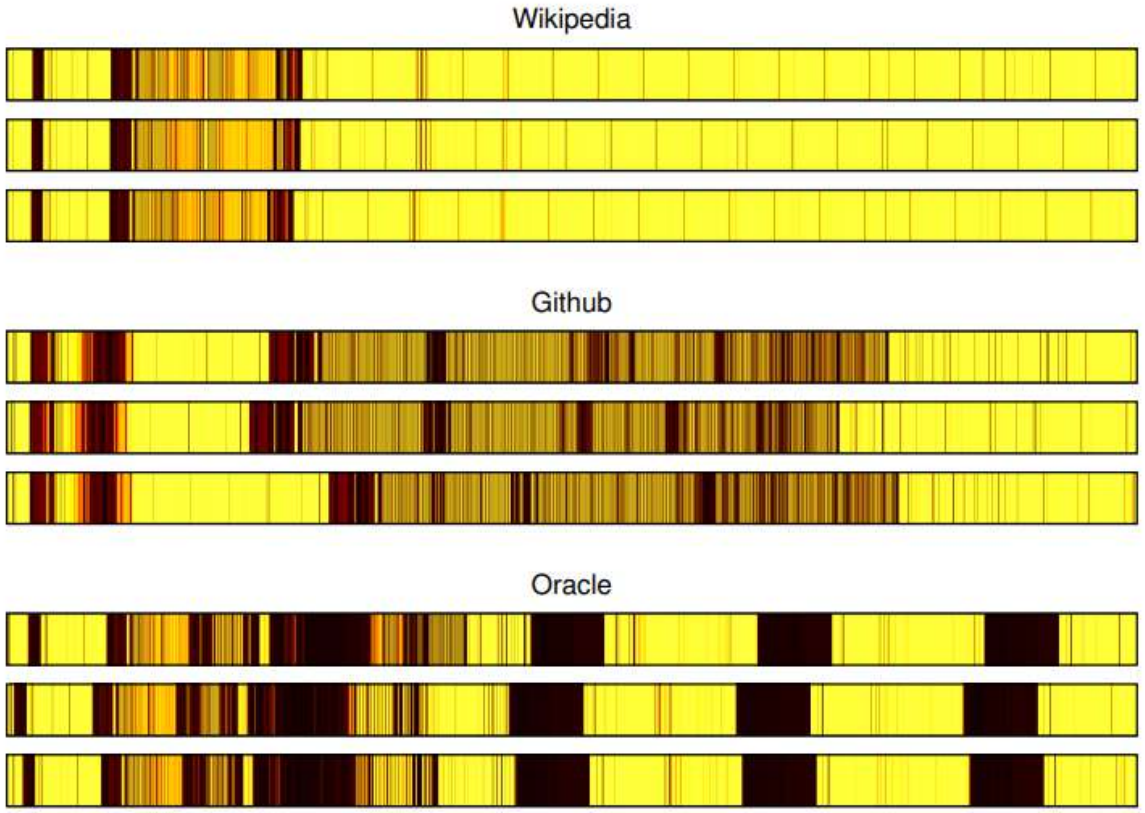
Identifying a person using his/her fingerprint, which is unique



Digital Fingerprinting

Many types - device, browser, website fingerprinting etc.

Here it is: Cache based website Fingerprinting



Examples of memorygrams. Time progresses from left to right, shade indicates the number of evictions. (Darker shades correspond to more eviction.)

Usecases: Close World and Open World

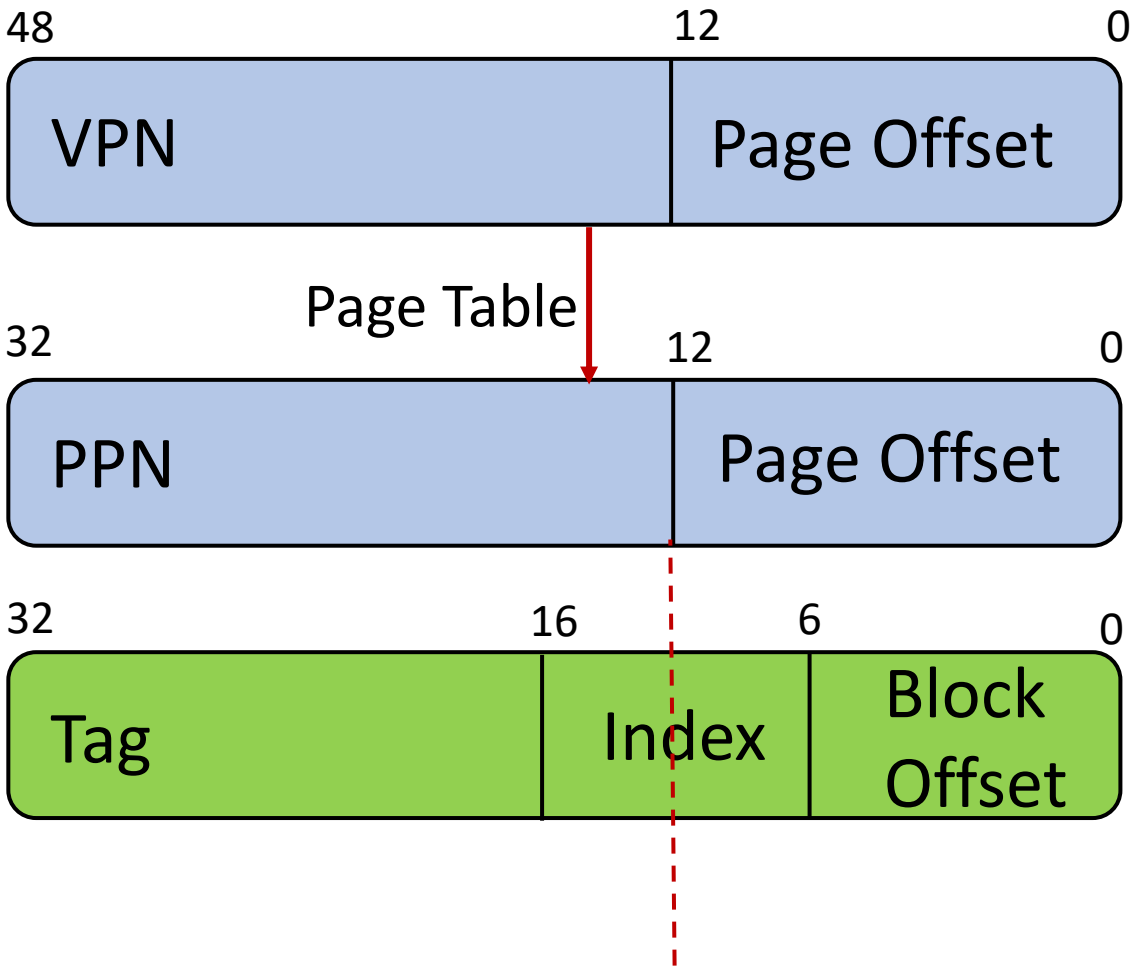
- Close World
  - Attacker needs to distinguish between a finite set of web pages.
  - But attacker should have knowledge of a complete list of sites the victim visits.
- Open World
  - Attacker wants to monitor access to set of sensitive websites, and classify them with high accuracy.


Back to Conflict based attack

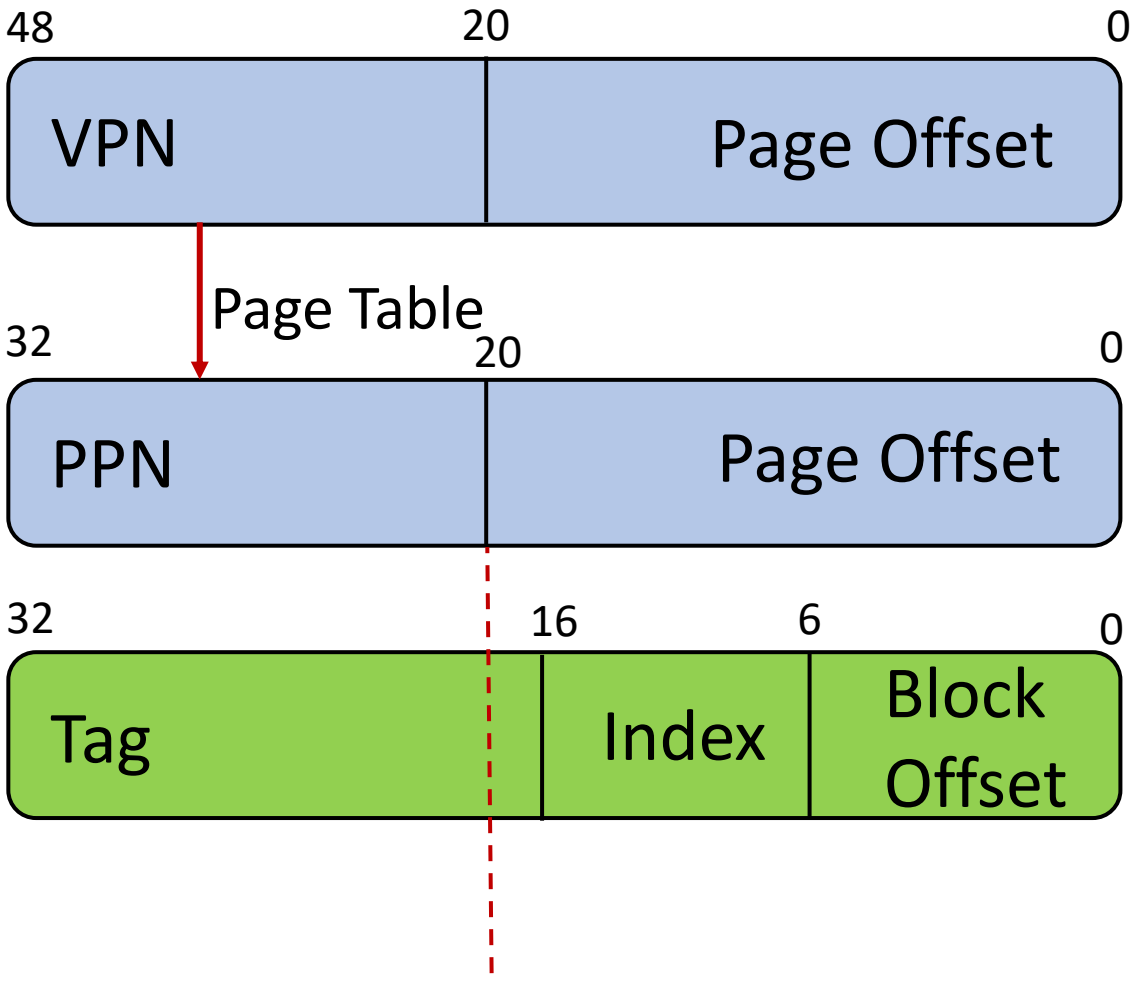
Thrashing Entire LLC (Prime+Probe): Questions of interest

- Extremely Slow pre-attack step: Think about an 8MB/16MB LLC
- Why not thrash a group of addresses that are mapped to the same set?
- Is there an algorithm to find out the same? Eviction set algorithm?
- But what about virtual to physical address translation? LLC will have the physical address.
- How to trigger requests that will go to the same set bypassing L1 and L2?

Attacker cannot control: LLC with 1024 sets



What if we have huge pages



Awesome. Now attacker can control all the accesses to a particular set.

What About?

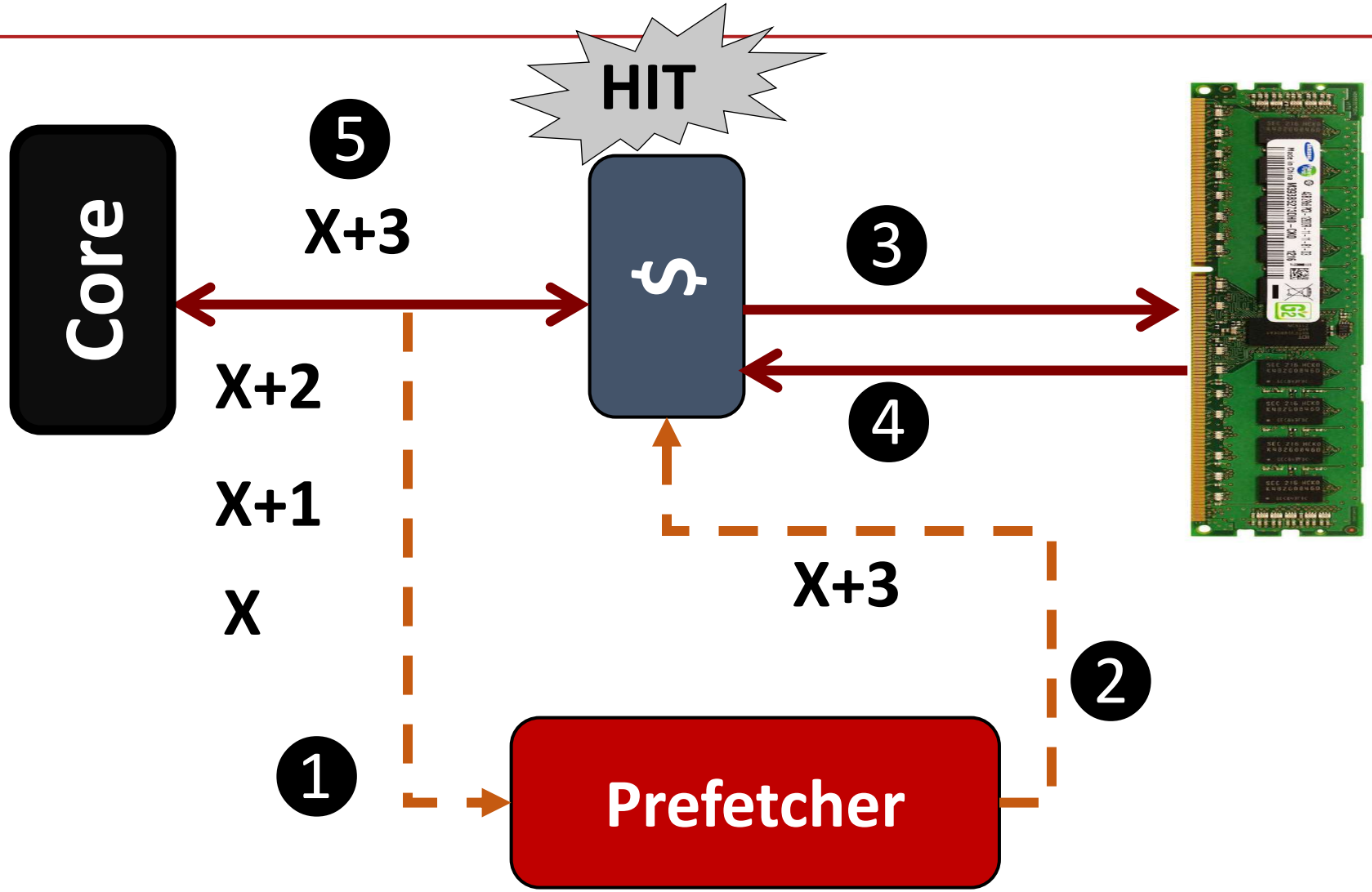
Effect of cache replacement policy at the LLC?

What if it is adaptive?

What if attacker's access pattern is predictable?

A hardware prefetcher can affect the eviction set creation process?

Hardware Prefetching



Effect of cache replacement policy at the LLC?

Fool the replacement policy too.

What if attacker's access pattern is predictable?

Fool the prefetcher too.

HOW GOOD IS THE ATTACKER?

ASSUMPTIONS

AGILITY (BANDWIDTH)

ADAPTIVE

ACCURACY

STEALTHY (DETECTOR CANNOT DETECT)