Microsoft

Microsoft Research

Faculty Summit 2012

ADVANCING THE STATE OF THE ART

Side Channels in the Cloud: Good News and Bad News

Michael Reiter
University of North Carolina at Chapel Hill

July 16, 2012

Side Channels

Extracts information from a computation based on its *implementation* …

… despite the fact that its implementation is faithful to its design (and the design is correct)

Usually (but not always!) used to attack cryptographic implementations

The target value is a cryptographic key

Access-Driven Side-Channel Attacks

The attacker runs a program on the system that is performing the cryptographic operation of interest

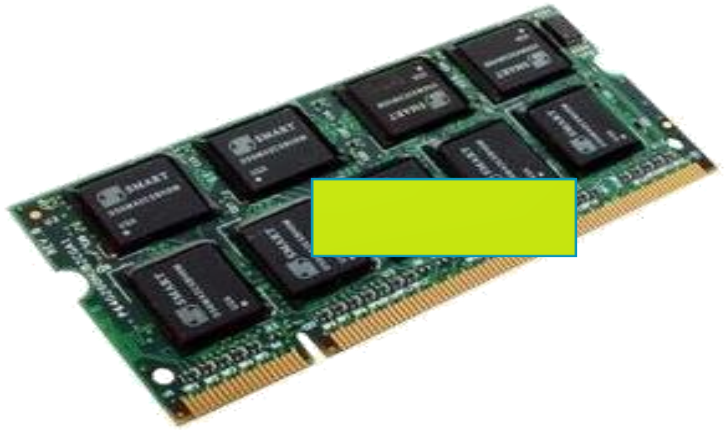
Basic idea: observe computation's effects on the system, and learn information from that

Recent attacks are *asynchronous*, in that they do not require the attacker to achieve precisely timed observations of the victim

Utilize SMT or ability to game the OS scheduler

None shown to work in virtualized SMP settings

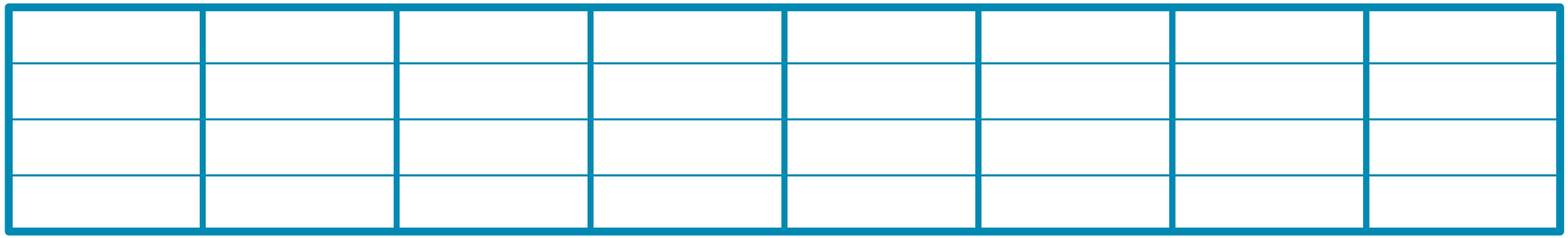
CPU Cache



Physical Address

4-way set associative cache

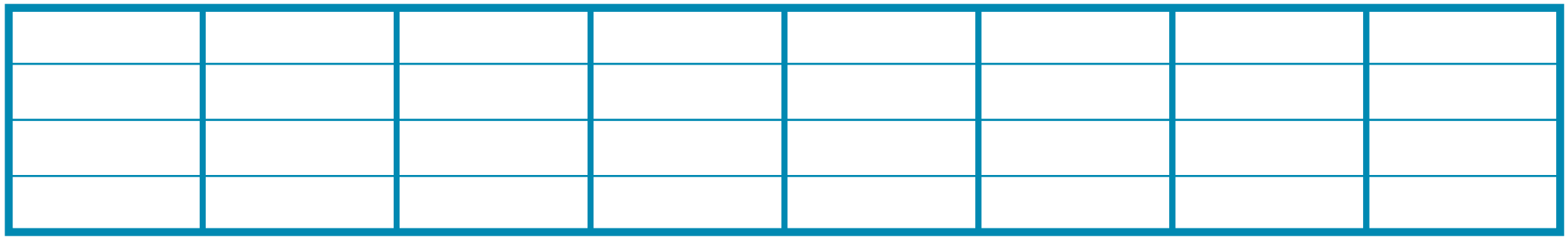
Cache Set



Cache Line

PRIME-PROBE Protocol

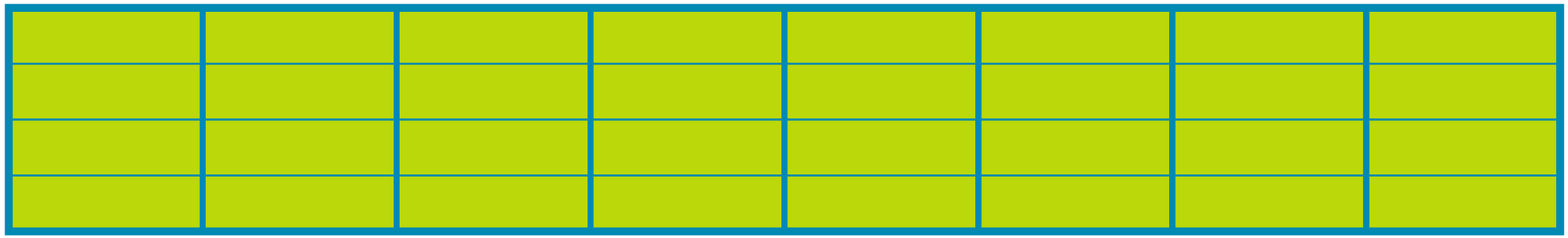
PRIME



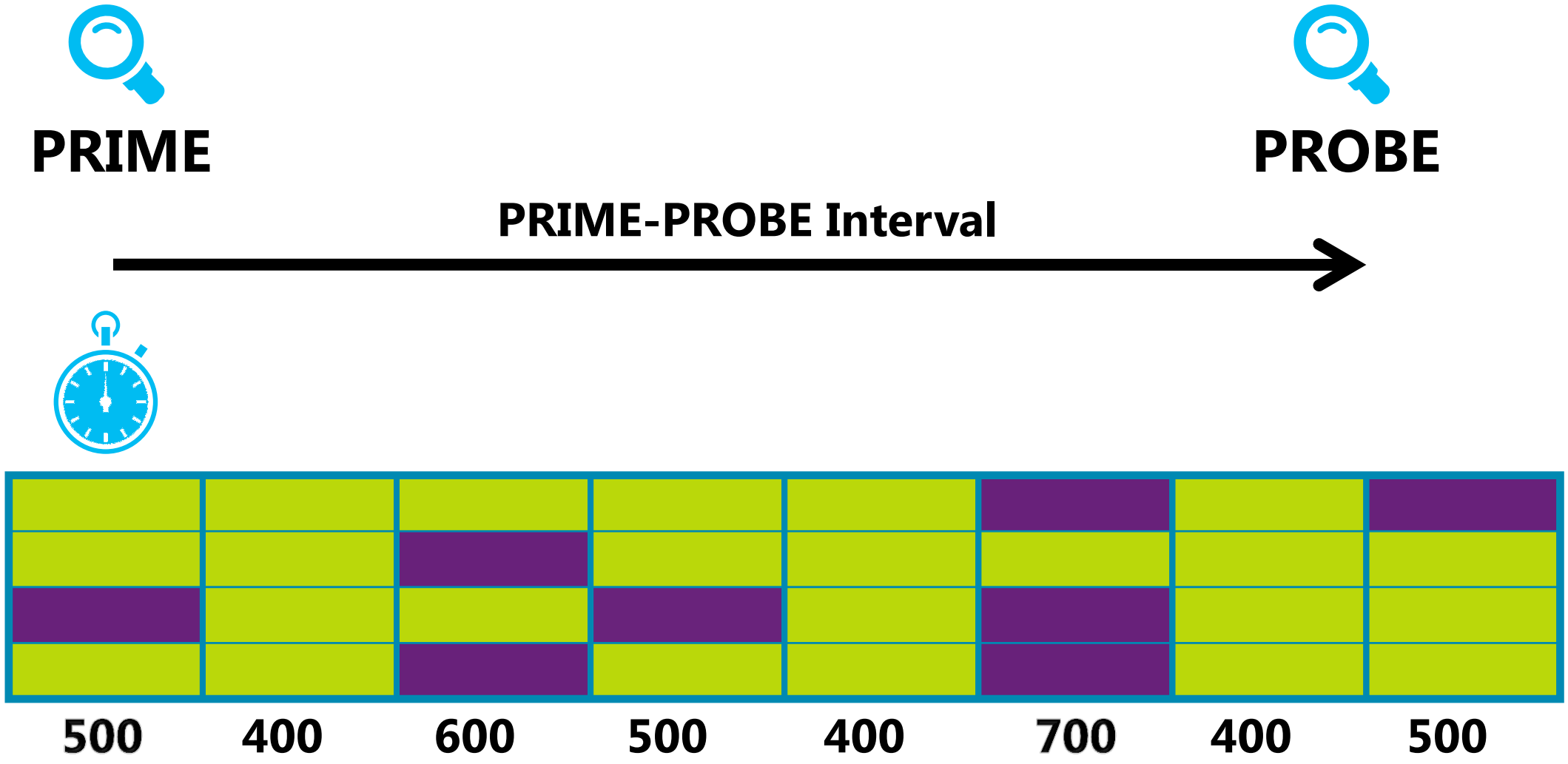
PRIME-PROBE Protocol

PRIME

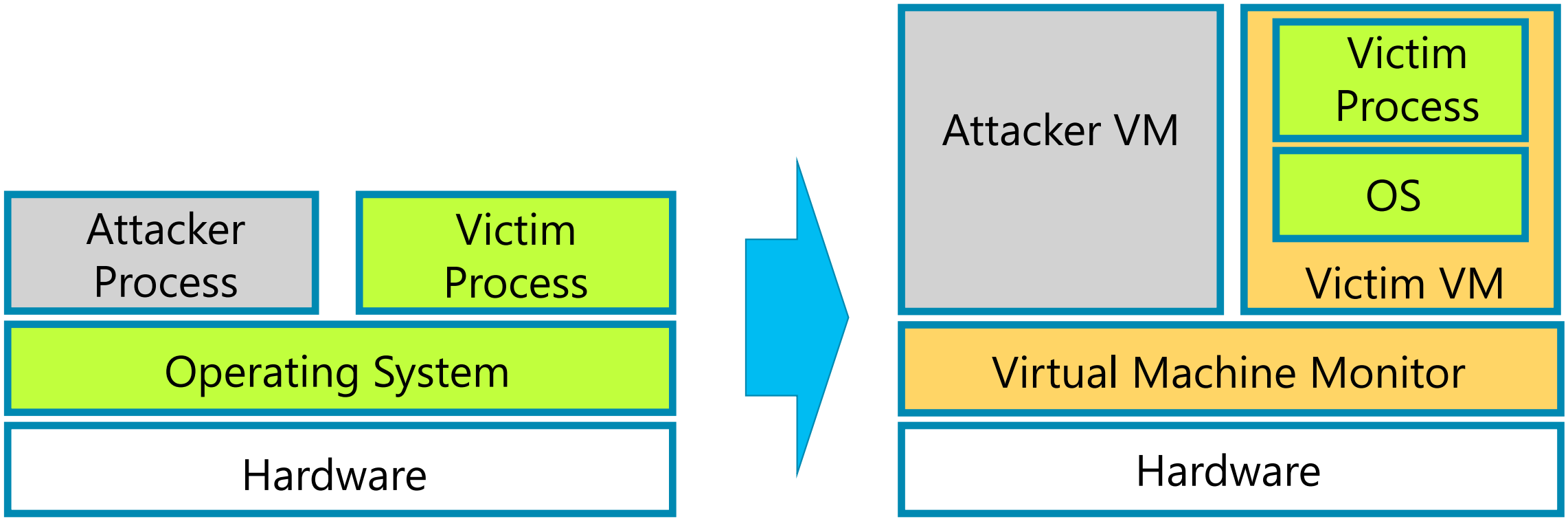
PRIME-PROBE Interval



PRIME-PROBE Protocol



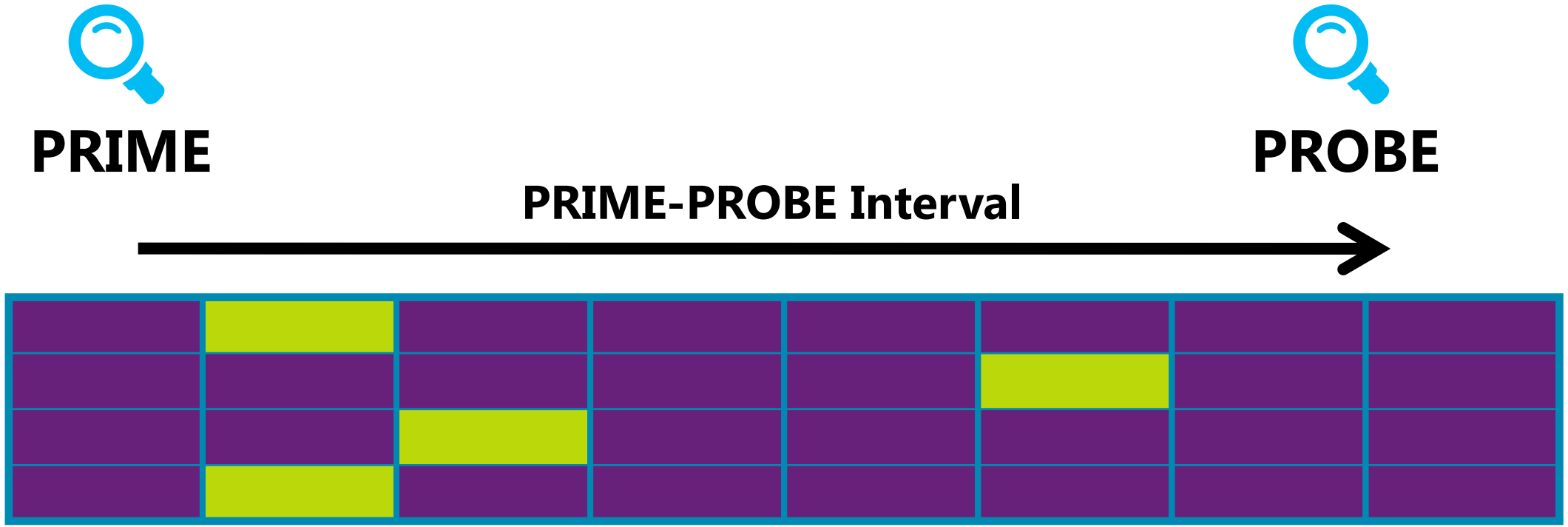
Cross-VM Side Channels



Virtualization adds a layer of software between the attacker and the victim

Cross-VM Side Channels
Challenge #1: Observation Granularity

- Scheduling quantum of Xen is 30ms
- Does not permit many observations of a crypto op

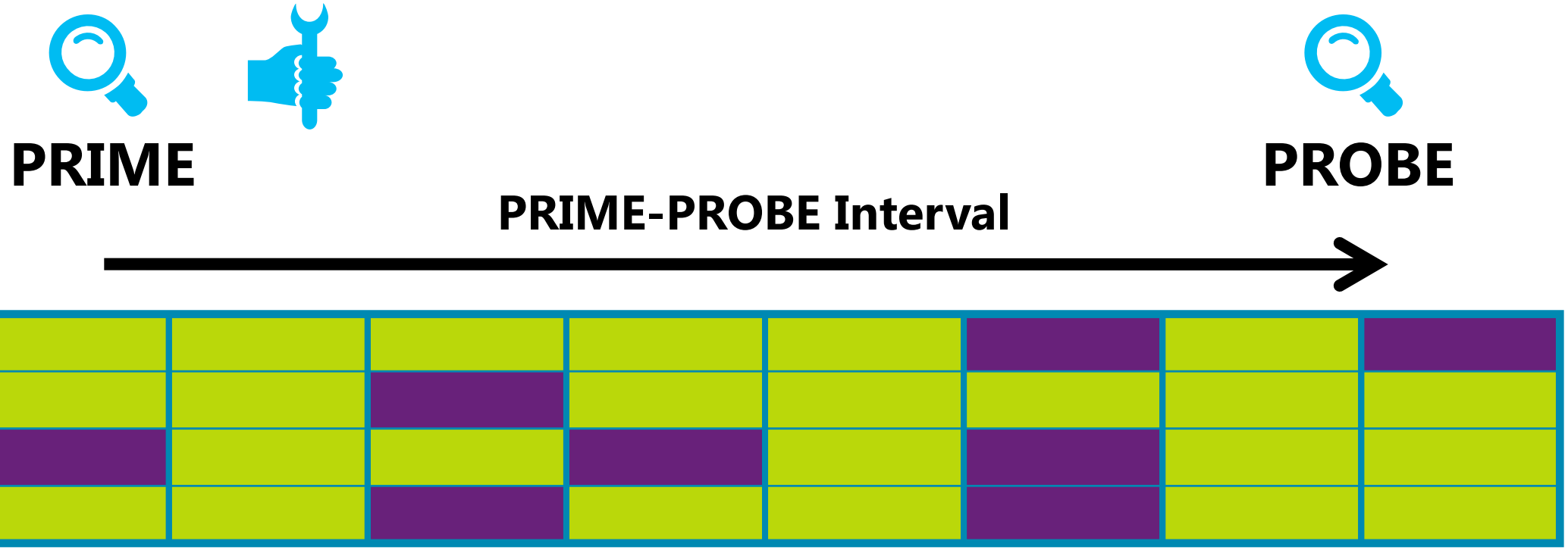


Cross-VM Side Channels
Challenge #2: Observation Noise

Numerous HW and SW sources of cache noise

Hardware: TLB misses, power saving, …

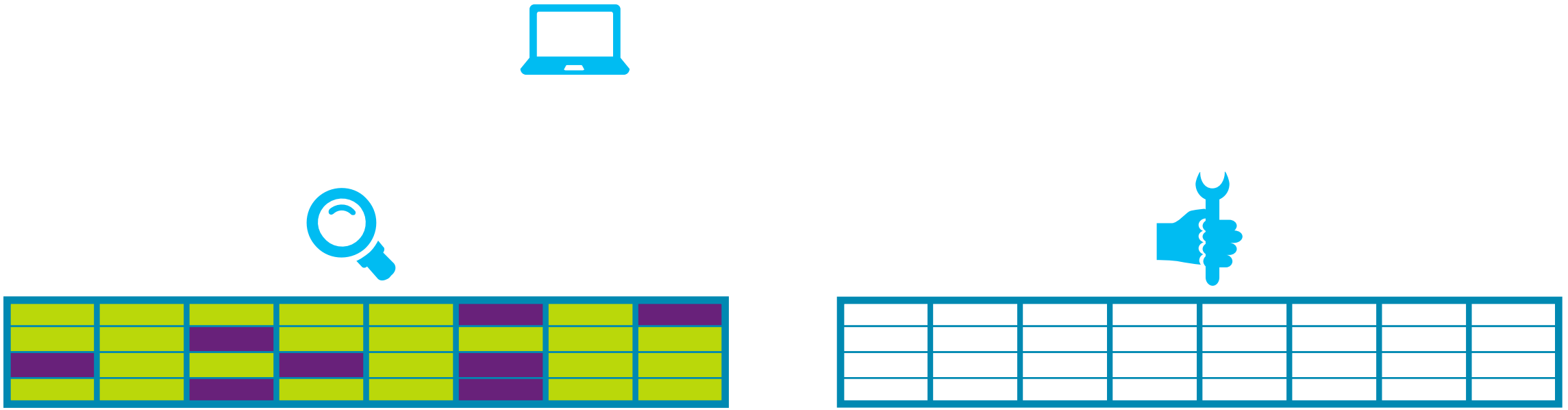
Software: Hypervisor, Dom0, …



Cross-VM Side Channels
Challenge #3: Core Migration

Attacker VM and victim VM will migrate across cores over time

So, many observations might not be the victim



Cross-VM Side Channels
Challenge #3: Core Migration

Attacker VM and victim VM will migrate across cores over time

So, many observations might not be the victim

Cross-VM Side Channels
Attack Strategy Against Modular Exponentiation

[w/ Zhang, Juels and Ristenpart 2012]

1. Game the Xen scheduler to interrupt victim with sufficient frequency
2. Classify cache-pattern observations, first individually and then in sequence

   Yields a collection of "execution fragments"
3. Apply customized sequence reconstruction algorithms to fragments

   Corrects errors and assembles full key "template"
4. Exhaustively search remaining key possibilities

Cross-VM Side Channels Experimental Setup

Attacked the implementation of ElGamal decryption in libgcrypt v.1.5.0

Specifically loaded the victim VM with Gnu Privacy Guard (GnuPG) v.2.0.19

Utilized the I-cache on a single-socket quad-core processor (Intel Core 2 Q9650)

Xen 4.0 as virtualization substrate

Each VM ran Ubuntu 10.04 server with Linux kernel 2.6.32.16

Scheduler was work-conserving

Non-work-conserving also possible, but harder

Cross-VM Side Channels
Summary of Results

Victim utilized a 4096-bit ElGamal modulus

Private exponent was 457 bits

Victim repeatedly performed decryptions, as if they could be triggered by the attacker

Done simply to speed up the test

~30,000,000 prime-probe trials over ~6 hours

After of several hours of post-processing, key space narrowed to 9,862 possibilities

Exhaustive search easily identified the key

Physical Isolation

A natural defense is to *physically isolate* VMs

Customer has exclusive use of a physical machine

Amazon offers dedicated instances in virtual private cloud

Confirming Physical Isolation

Cloud provider may accidentally violate service level agreement or take shortcuts

Configuration error

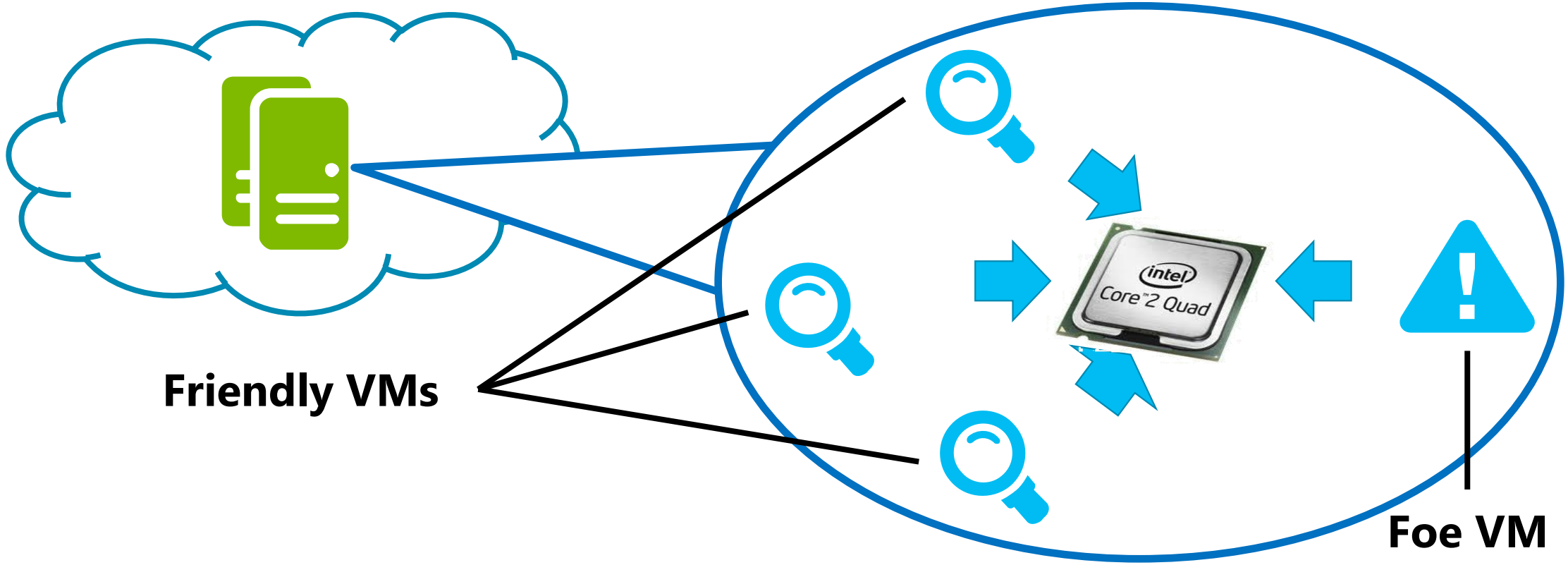
Lower cost

Cloud customer has no control or visibility into the virtualization layer

Verification and auditing is difficult

HomeAlone

[w/ Zhang, Juels and Oprea 2011]

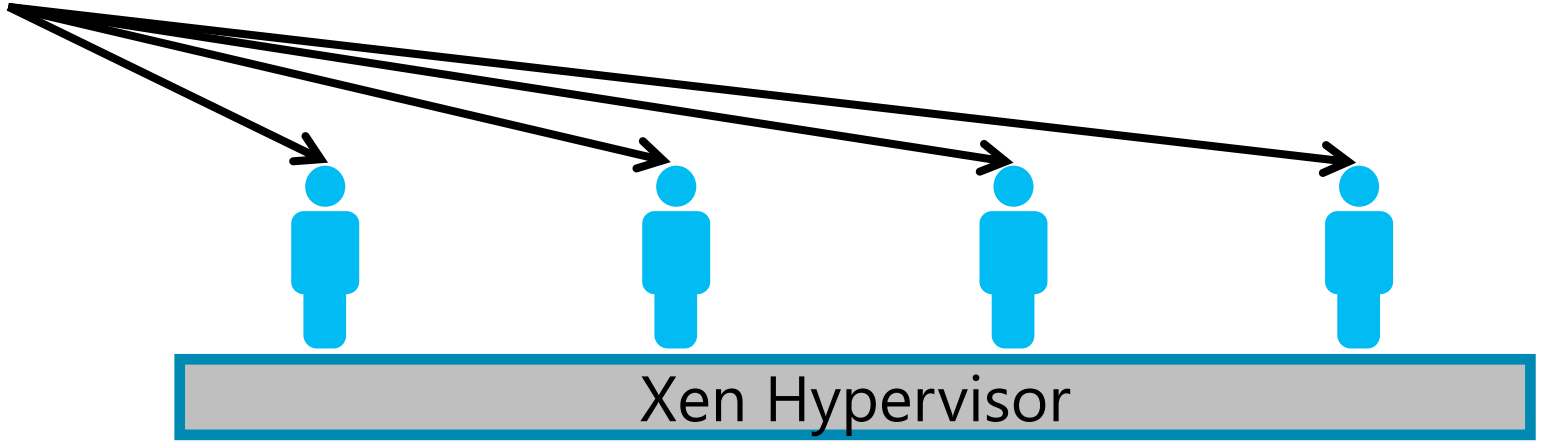


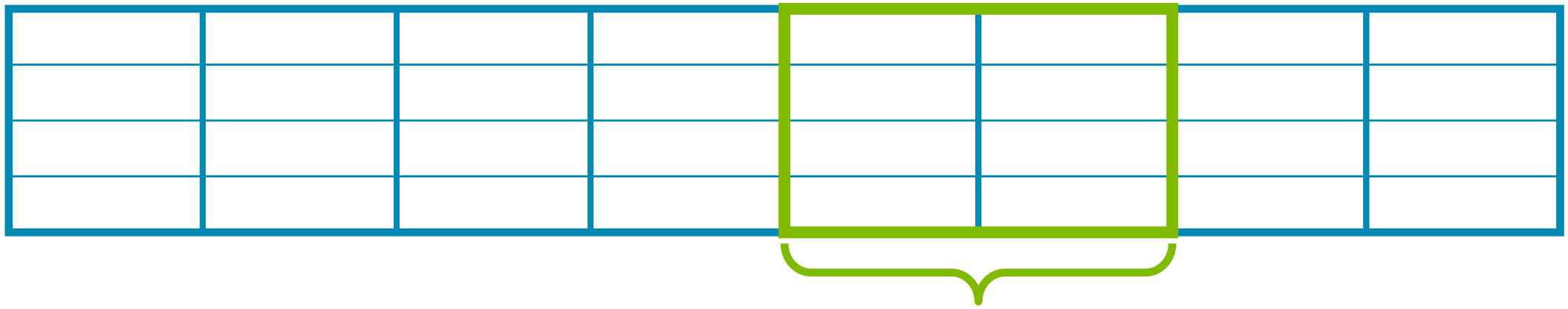
Friendly VMs: VMs controlled by the legitimate tenant

Foe VMs: unexpected third party VMs

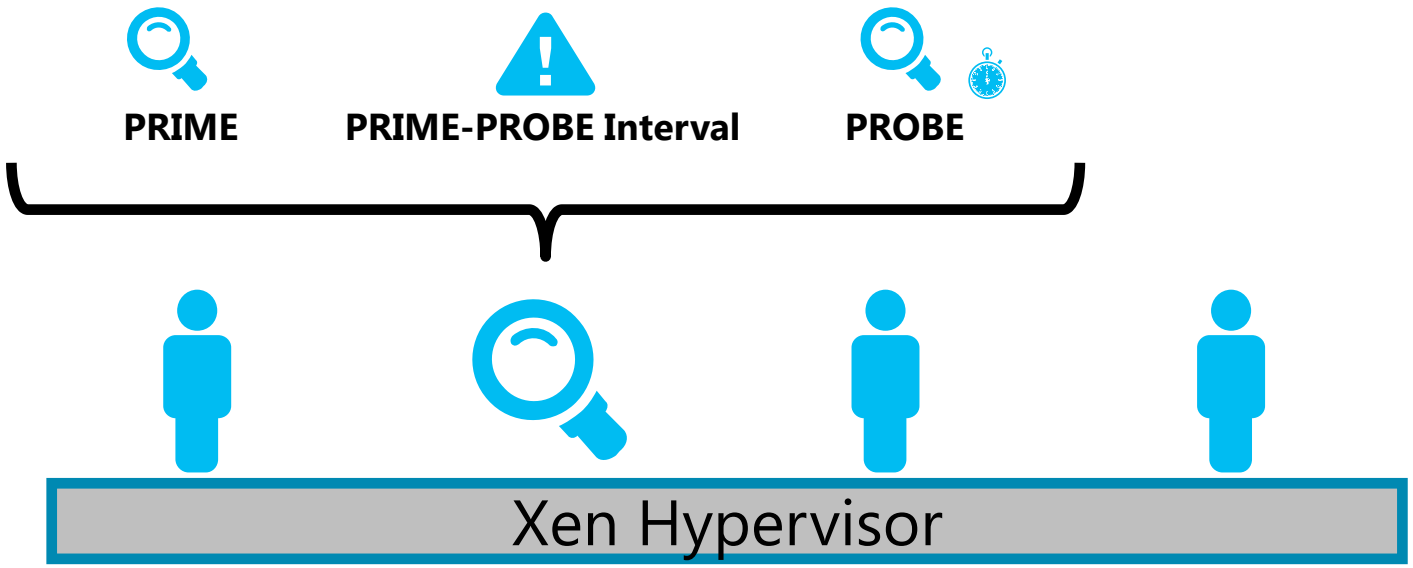
HomeAlone
How it Works

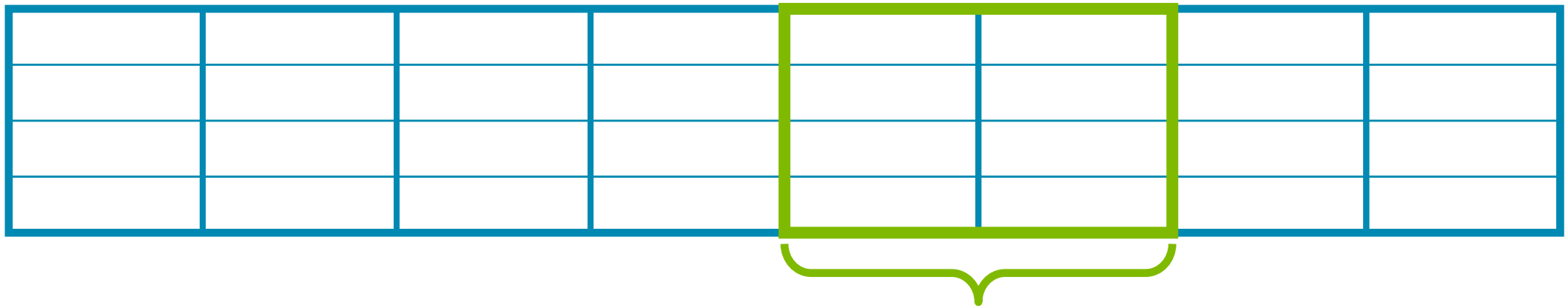
Friendly VMs



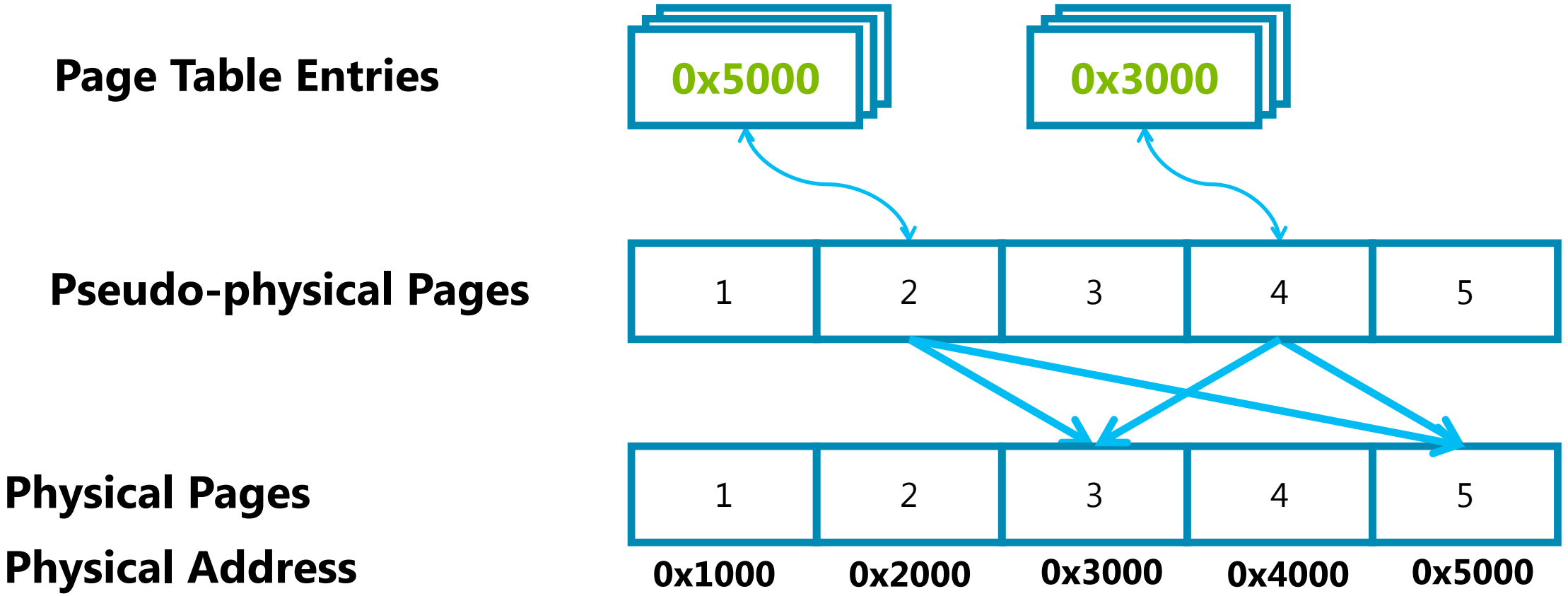


HomeAlone
How it Works

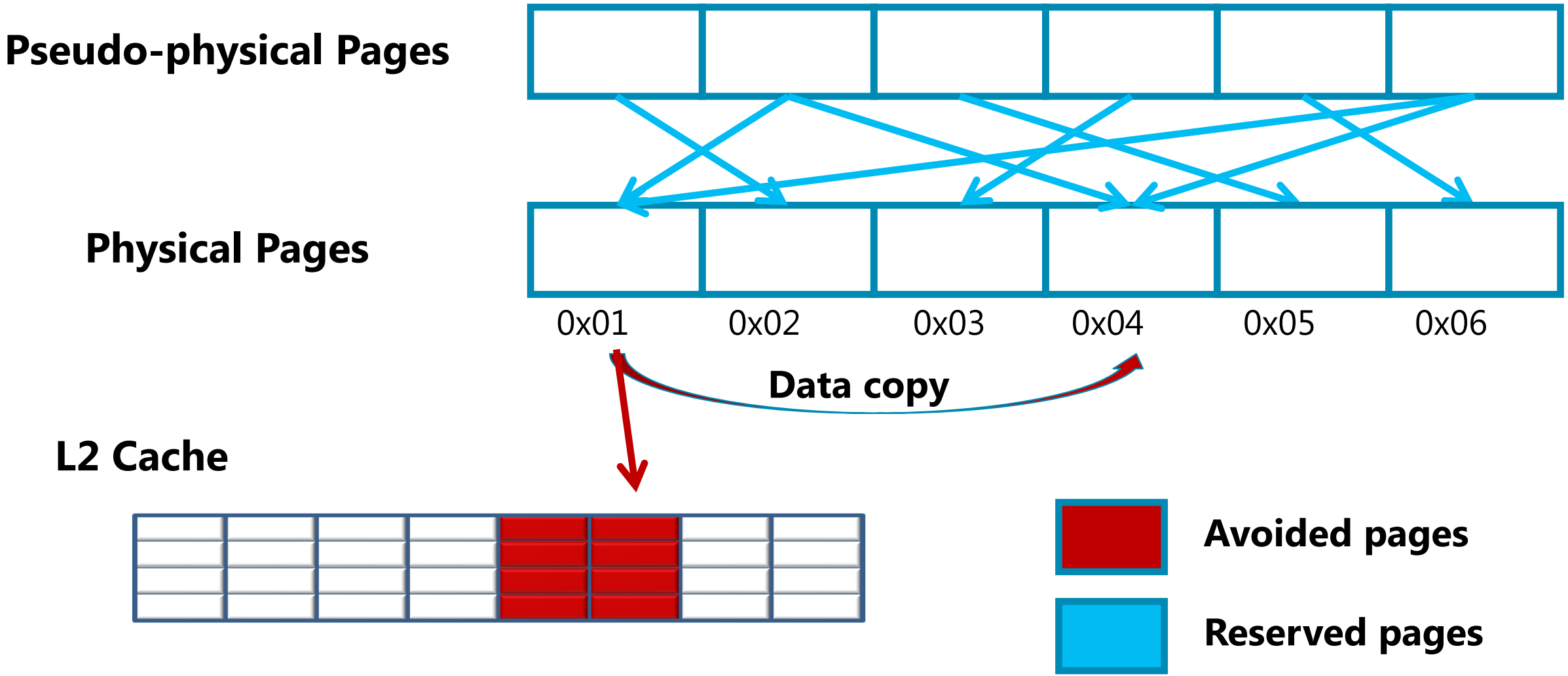




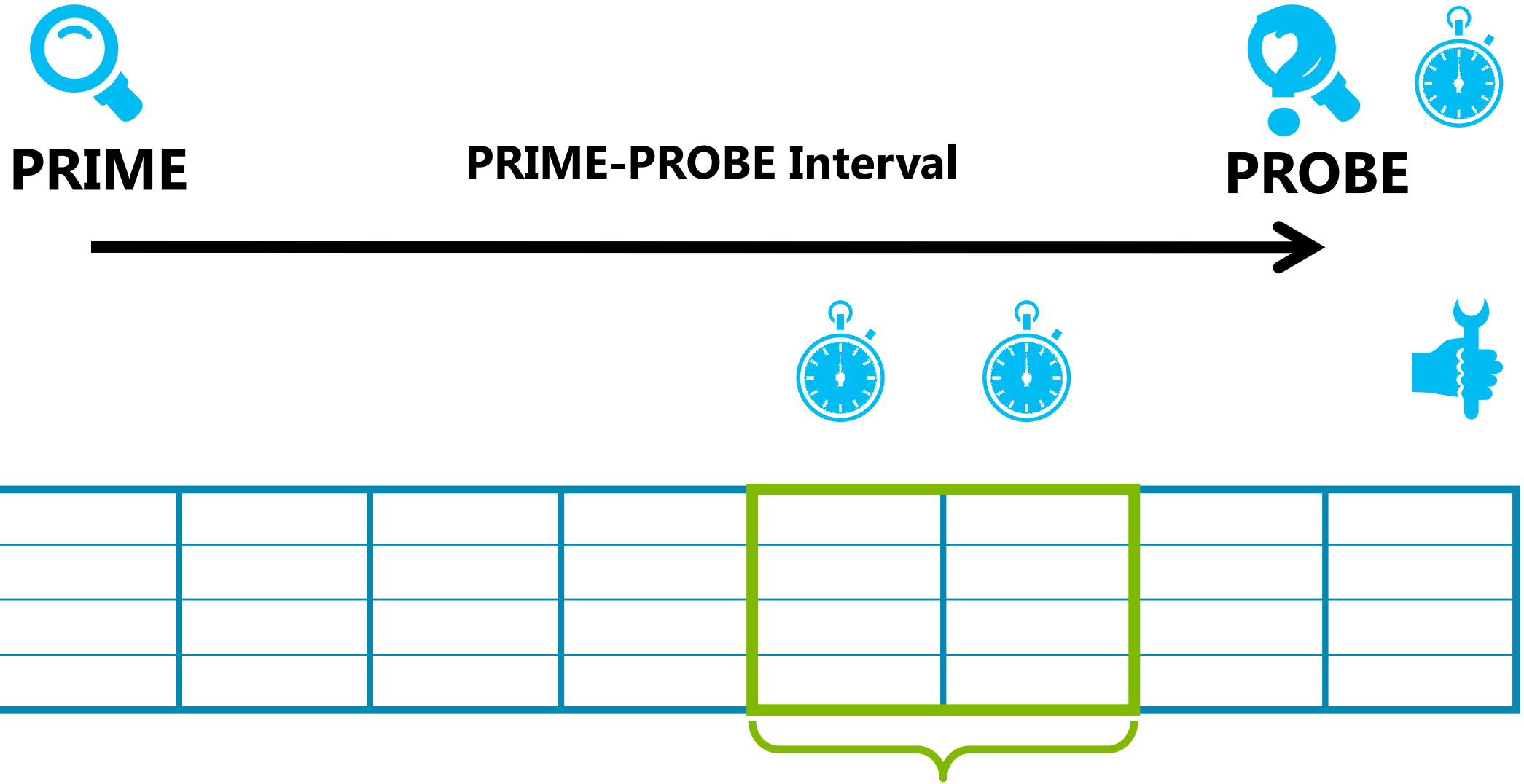
HomeAlone
Address Remapping in PVM



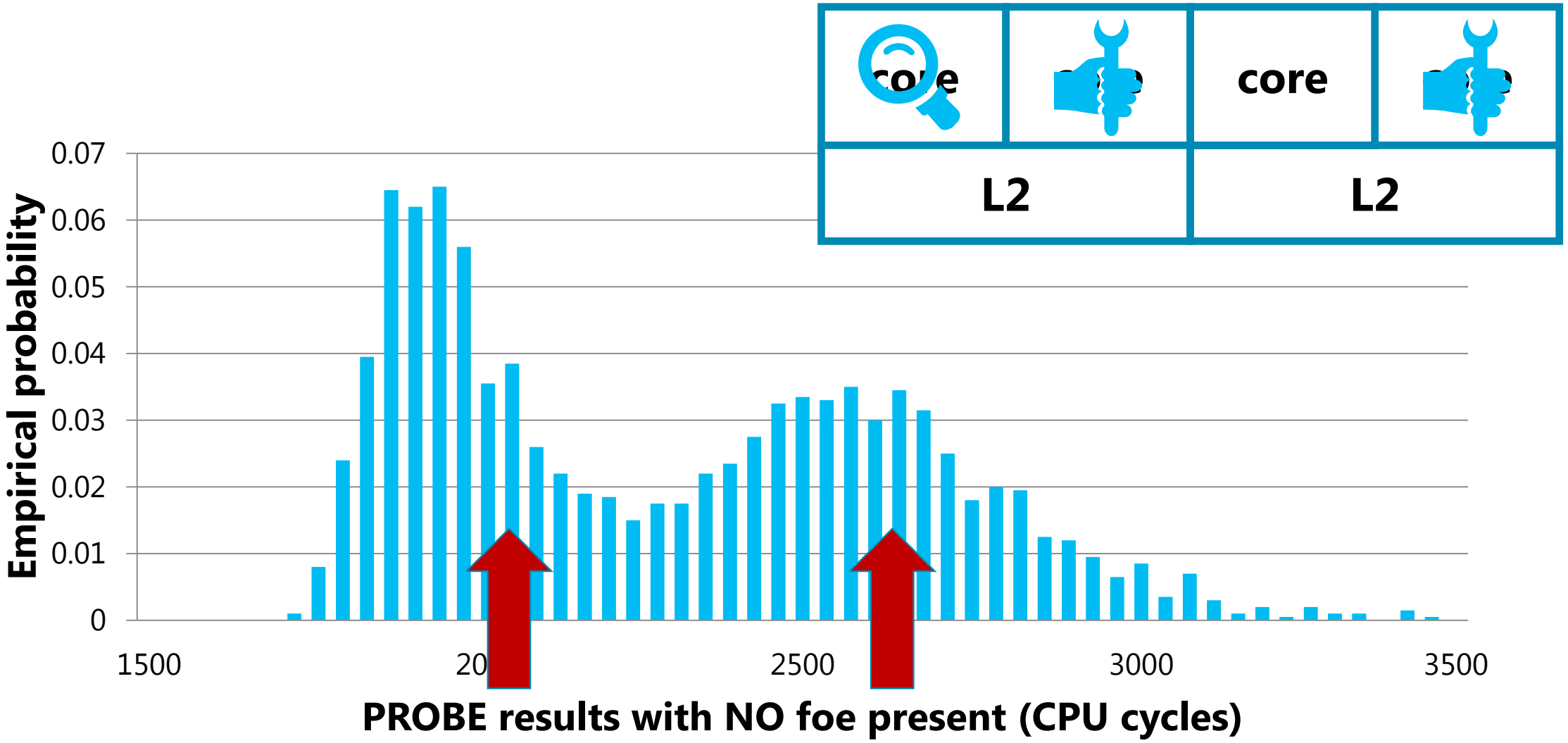
HomeAlone
Address Remapping in PVM



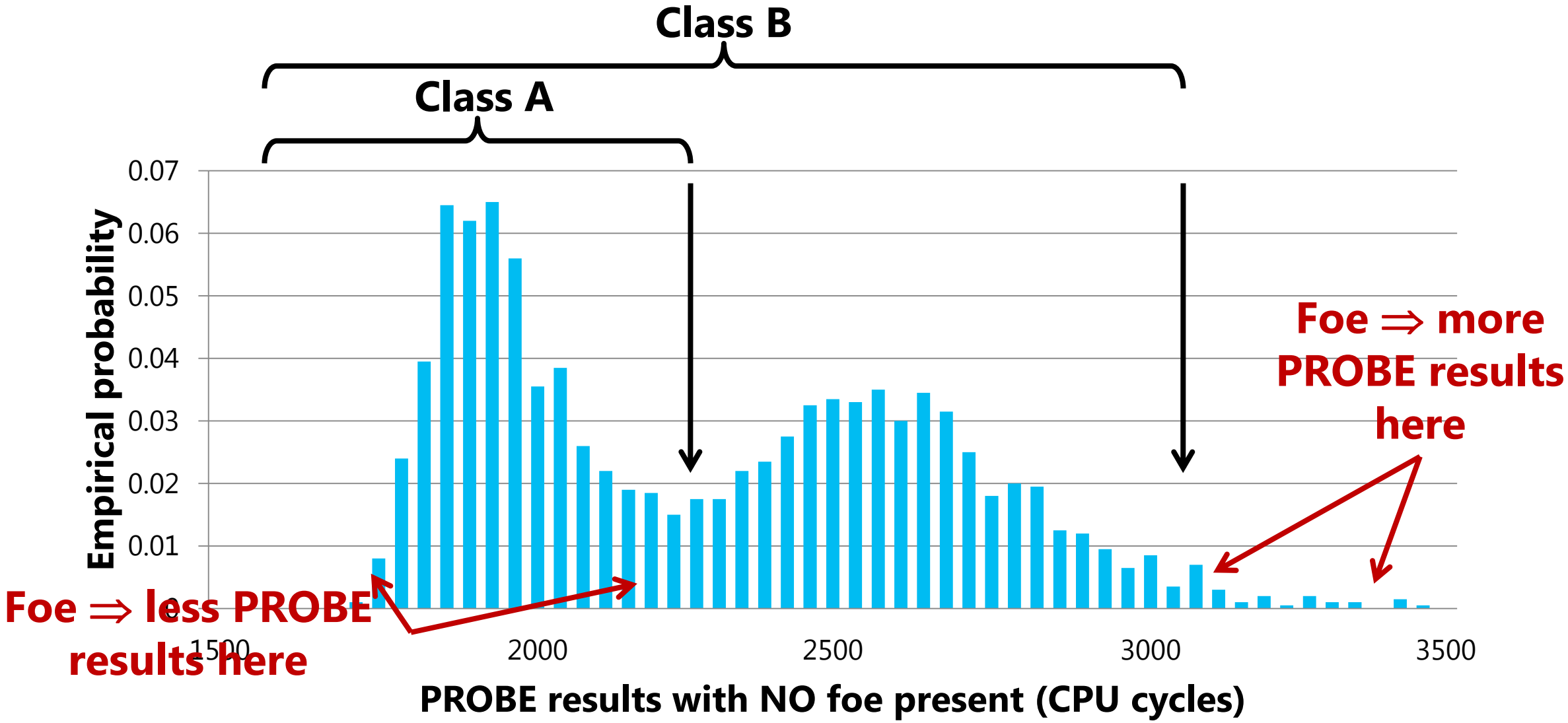
HomeAlone
Dom0 Cache Activity



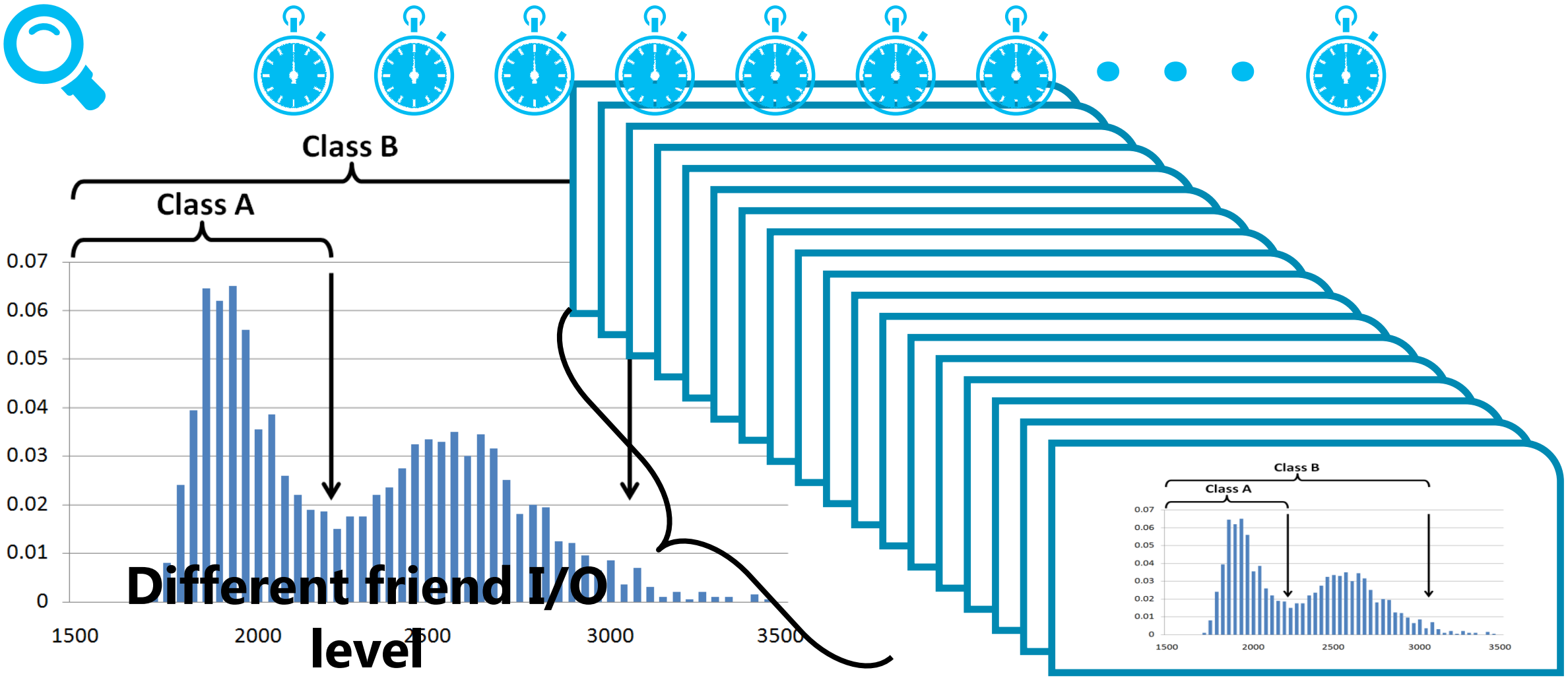
HomeAlone
Distribution of PROBE Results



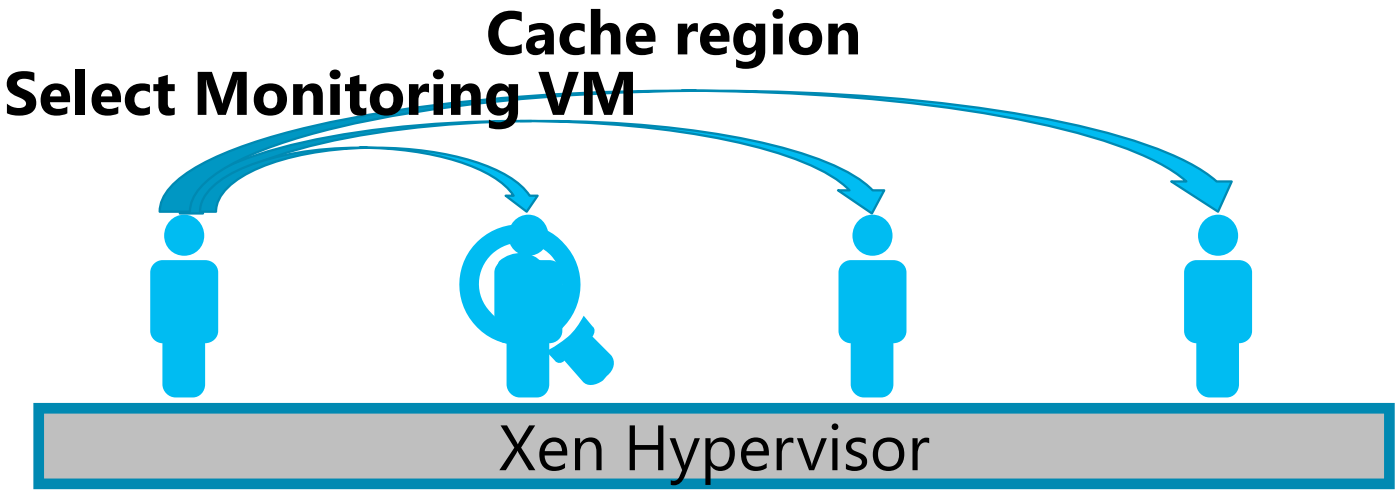
HomeAlone
Distribution of PROBE Results

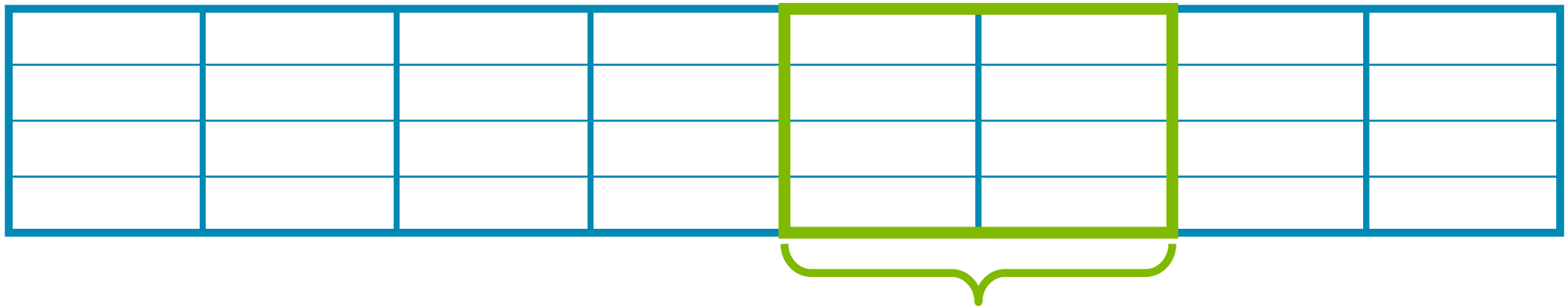


HomeAlone
Cache Activity Classifier

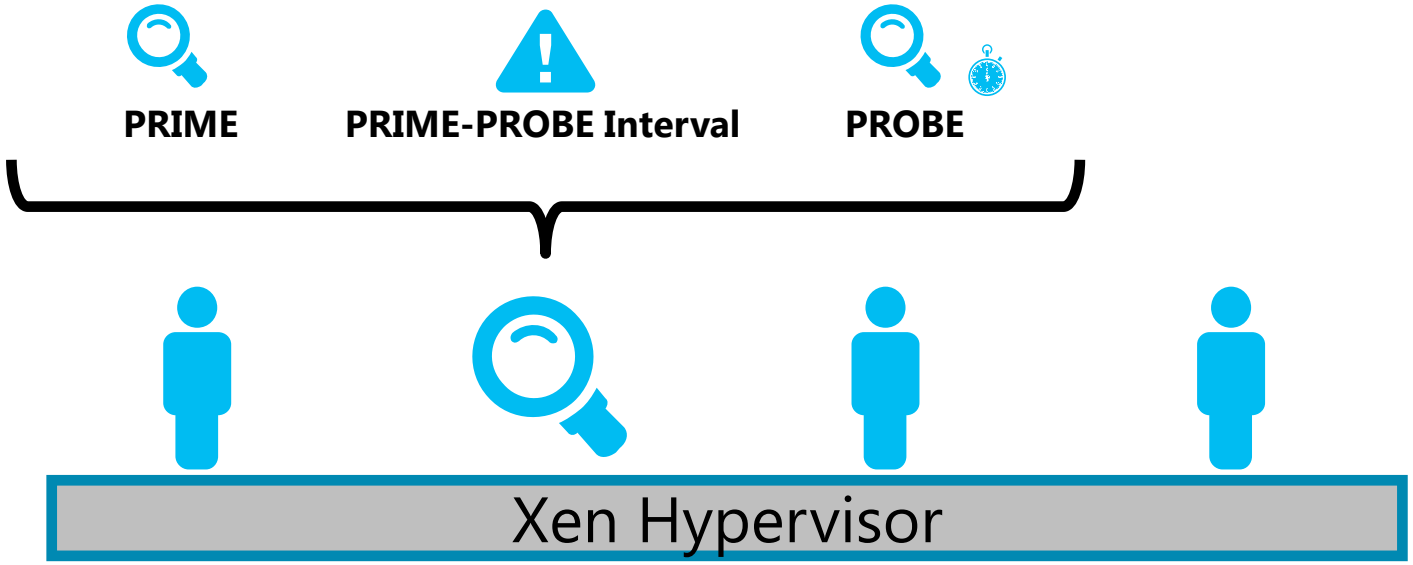


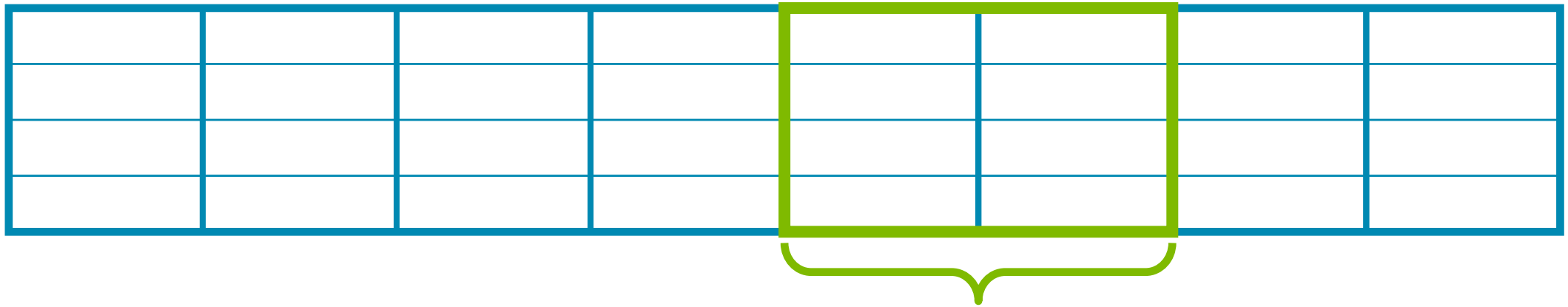
HomeAlone
Putting Everything Together



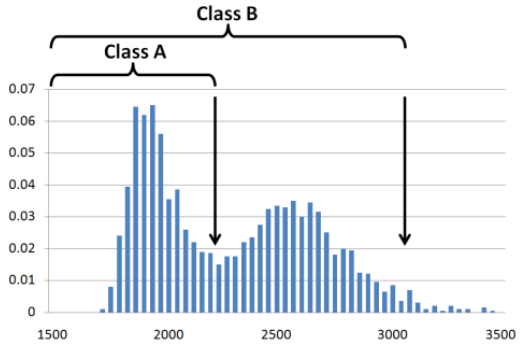


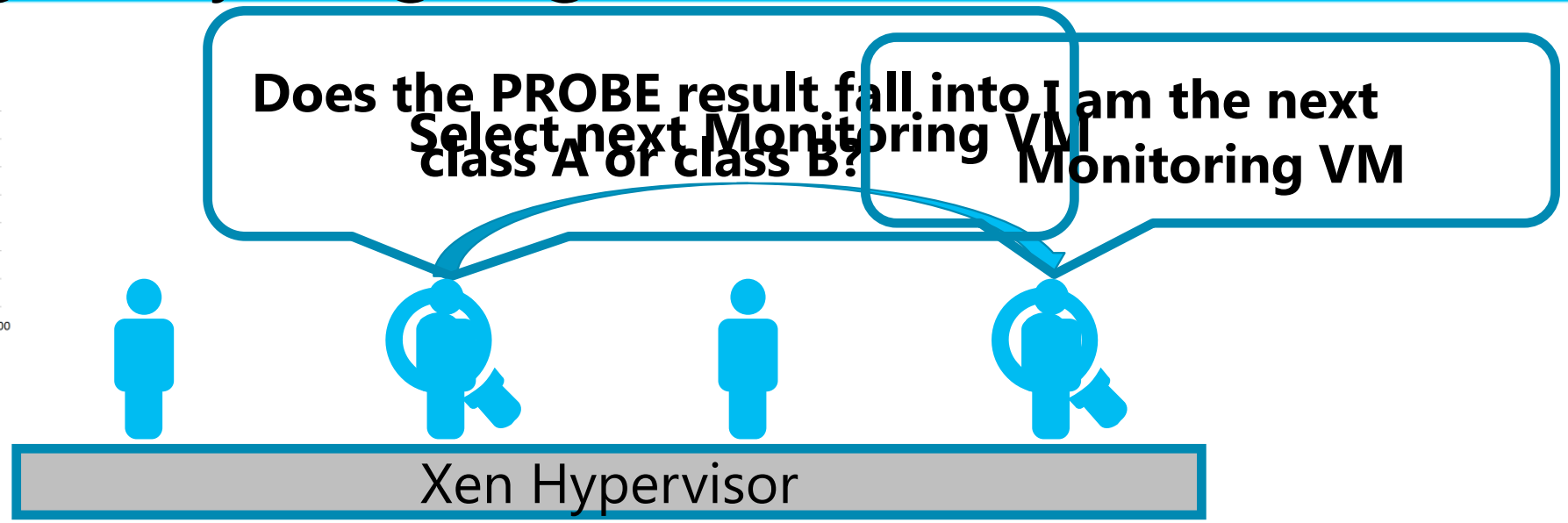
HomeAlone
Putting Everything Together

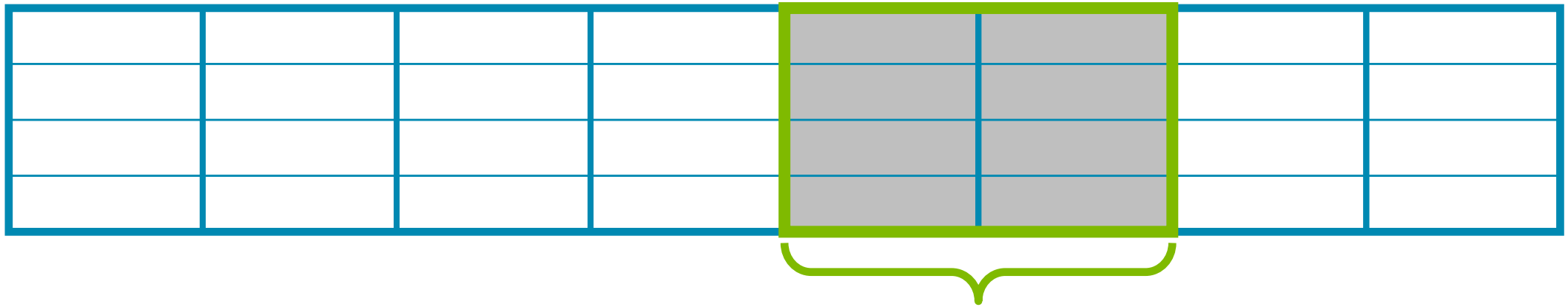




HomeAlone
Putting Everything Together







HomeAlone Experimental Setup

"Cloud" platform: Intel Core 2 Quad processor, two shared L2 cache, no SMT

HomeAlone implemented in 64-bit PVOps Linux kernel 2.6.32 for Xen 4.0

Classifier parameters:

Cache avoided: 1/16th

PRIME-PROBE interval: 30 ms

Detection period: 25 PRIME-PROBE trails

HomeAlone Result Highlights

True detection rate (with 1% false positive)

Foe VM running cloud applications

- Simulated with PARSEC benchmarks: 84% - 100%

Foe VM running PRIME-PROBE protocol

- Less frequent, smaller cache region: 15%
- More frequent, larger cache region: 85%

Performance overhead

Address remapping: 150ms for remapping a 2GB memory (1/16 mapped to monitored cache region)

Less than 5% overhead during detection period

Conclusions

Bad news: Cross-VM side channels with sufficient fidelity to extract private keys are possible

Good news: Friendly VMs can use side channels to confirm that they are physically isolated from others

Microsoft