VERNAM Group
Security & Privacy @ WPI

Cache Attacks on the Cloud

Thomas Eisenbarth
Joint work with Gorka Irazoqui, Mehmet Sinan Inci, Berk Gulmezoglu and Berk Sunar

Real World Cryptography 1/8/2016

WPI

Outline

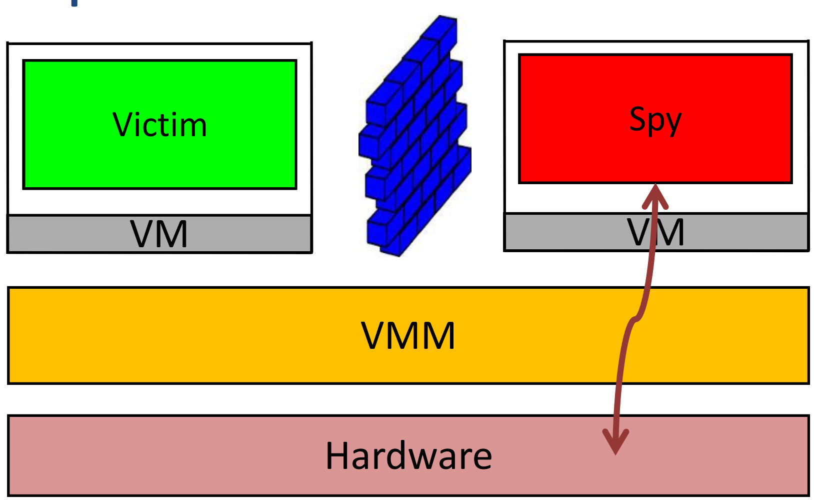
- Cloud Computing and Isolation
- Extracting Information from Co-located VM
- Attacking Crypto across VM Boundaries
- RSA Key Recovery in a Public Cloud

Cloud Computing

- Computation increasingly outsourced to cloud servers
- CSPs: many users on shared, homogeneous platforms
- Users rent VMs, share same computer
- **Shared resources → Information Leakage?**

Security through Isolation

- Virtual machines: Abstraction of physical machine
- Hypervisor (VMM) ensures Isolation through virtualization
- VMs might *feel* each other's load on some low-level resources→ potential side channels



Outline

- Cloud Computing and Isolation
- **Extracting Information from Co-located VM**
- Attacking Crypto across VM Boundaries
- RSA Key Recovery in a Public Cloud

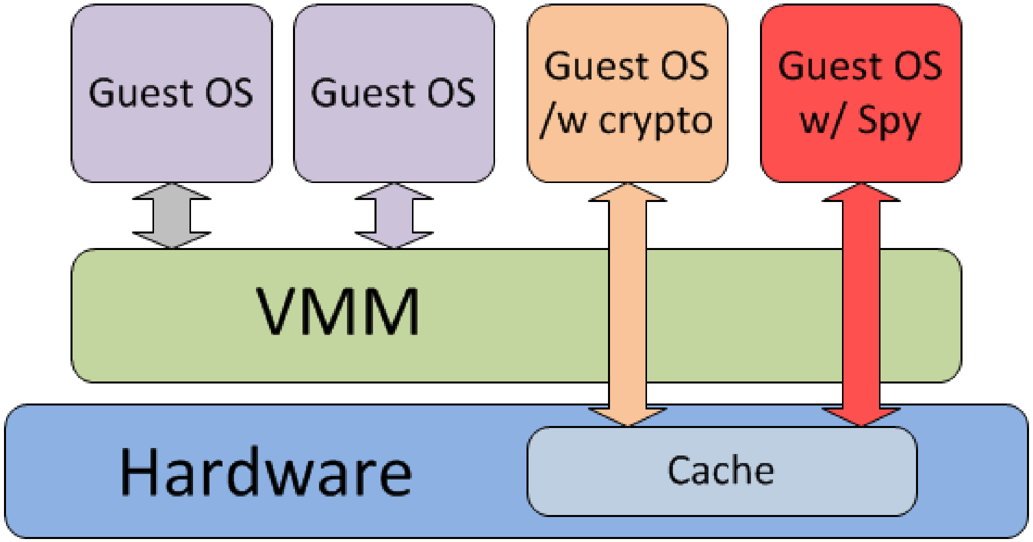
Cross-VM Side Channel Attack

Suitable covert channel in the cloud?

– Cross Core: Last Level Cache (L3 Cache) accesses

→Adversary and victim share full access to L3 cache

→Cache Access cannot be virtualized (70x slowdown)



Cache Attacks?

- Cache attacks are old [Hu92]
- General technique: *Prime+Probe* [OST06]:
  1. **Prime** desired memory lines
     fill monitored cache lines with data making an eviction set
  2. Wait for some time
  3. **Probe** memory lines
     read eviction set data and time read
- Problems:
  - Usually only applied on L1-Cache (64kB) →not cross-core
  - L3-Cache is too large (25MB!) not controlled by spy
  - **Solution**: **Huge Pages** give spy control over L3$

[Hu92] Hu, W.-M. (Digital Equipment Corp., Littleton, MA, USA) *Lattice scheduling and covert channels.* IEEE Oakland 92
OST06] DA Osvik, A Shamir, E Tromer *Cache attacks and countermeasures: the case of AES*. CT-RSA 2006

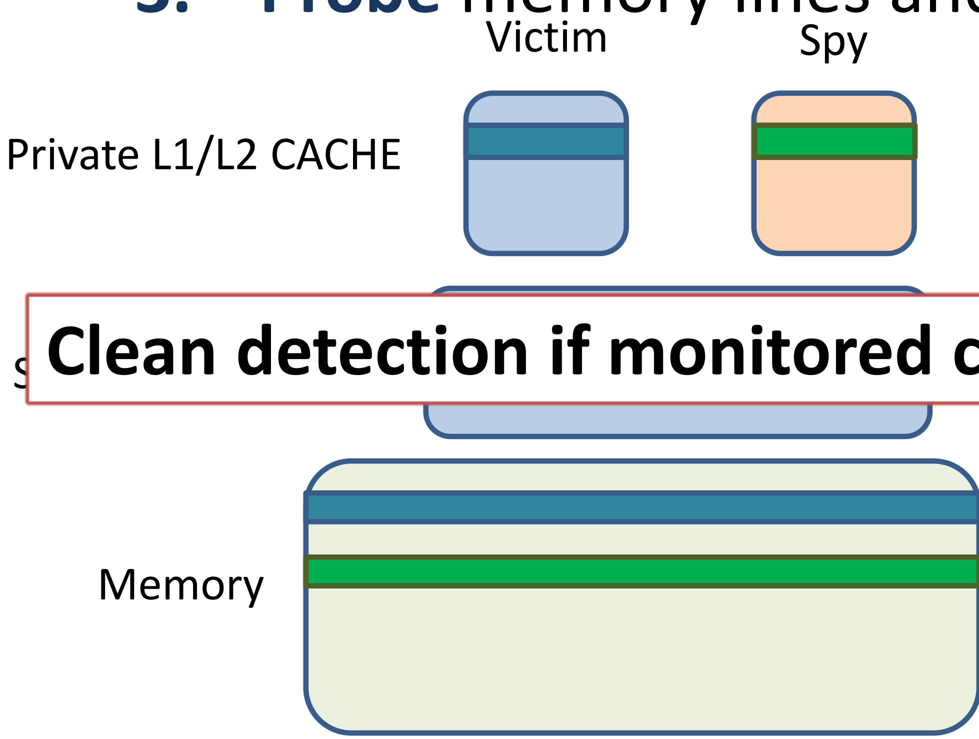
Outline

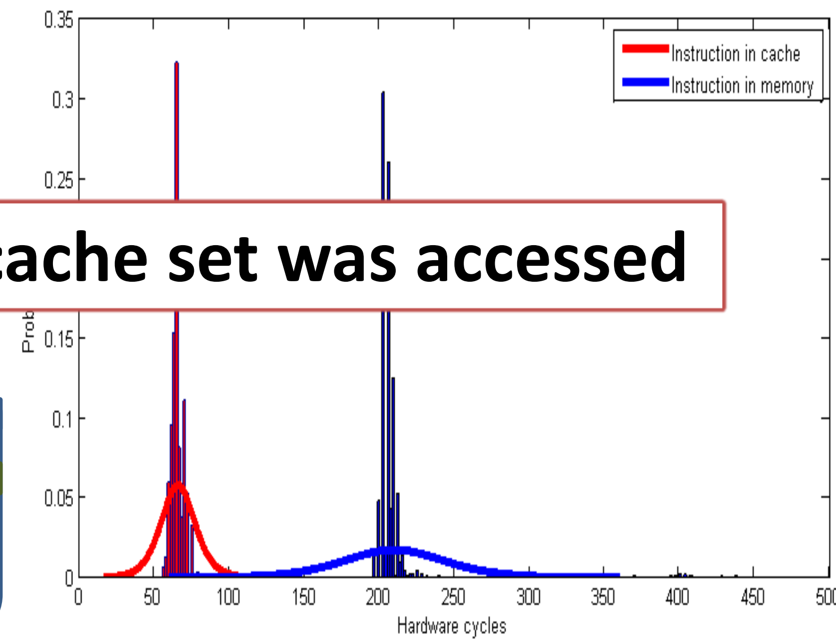
- Cloud Computing and Isolation
- Extracting Information from Co-located VM
- **Attacking Crypto across VM Boundaries**
- RSA Key Recovery in a Public Cloud

Prime+Probe Attack: Concept

Steps: (Preparation: Find **eviction set**)

1. **Prime** desired memory lines
2. Wait for some time
3. **Probe** memory lines and measure reload time.



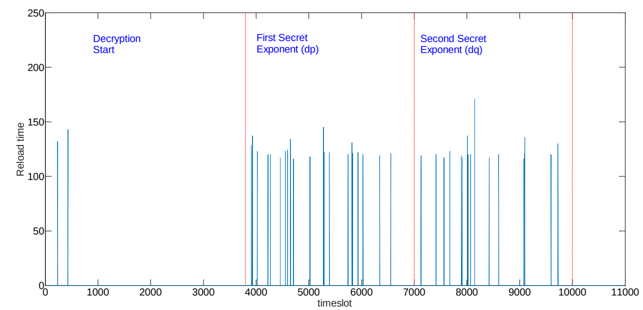


Clean detection if monitored cache set was accessed

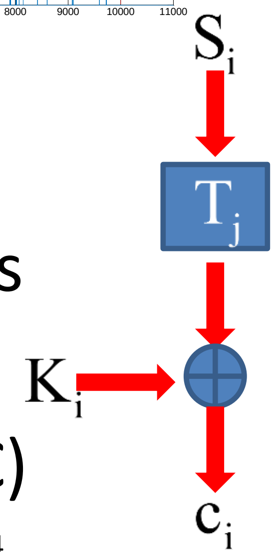
How to get crypto keys?

Detect **key-dependent** cache accesses:

- **RSA/ElGamal:**
  - Sliding window exponentiation
  - Occurrence of multiplicands in cache reveals key bits



- **AES:**
  - T-table implementation: Xors and table lookups
  - Detect t-table access in last round (table entry corresponding to c_i is always in LLC)



[YF14] Y Yarom, KE Falkner *Flush+ Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack,* USENIX Security 2014

[IIES14] Irazoqui, G., Inci, M. S., Eisenbarth, T., & Sunar, B. *Wait a minute! A fast, Cross-VM attack on AES*. RAID 2014

Are Cross-VM Cache Attacks Realistic?

Cross-VM Cache Attacks on El Gamal [LY+15] and on AES [IES15] work if

- Server has a shared level of cache
- Attacker and the victim are physically co-located
- ~~VMM implements memory deduplication~~

[LY+15] Liu, F., Yarom, Y., Ge, Q., Heiser, G., & Lee, R. B. (2015). *Last-Level Cache Side-Channel Attacks are Practical*. (S&P 2015).
[IES15] Irazoqui, G., Eisenbarth, T., & Sunar, B. *S$A: A shared cache attack that works across cores and defies VM sandboxing—and Its application to AES*. 36th IEEE Symposium on Security and Privacy (S&P 2015)

Outline

- Cloud Computing and Isolation
- Extracting Information from Co-located VM
- Attacking Crypto across VM Boundaries
- **RSA Key Recovery in a Public Cloud**

Co-location

First success in 2009 [RTS09]:

1. Launch many instances on cloud
2. Check if any are co-located

*** In Sept 2008**

- How to detect Co-location?
  - Ping time?
  - IP address of instance or hypervisor?
  - Disk Load?

[RTSS09] Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. *Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds.* ACM CCS '09

Test Setup

- AWS EC2 m2.medium instances:
  - Intel Xeon E5 2670 v2 CPU @2.5 GHz
  - 10 cores share 25 MB of L3 cache
  - Modified (Hardened) Xen VMM
  - Up to 10 co-located instances (VMs)

- 4 accounts w/ 20 instances (no within-acc colocation)
- Ping is constant time
- HDDs replaced with SSDs
- Dom0 IPs hidden

New Co-location detection needed

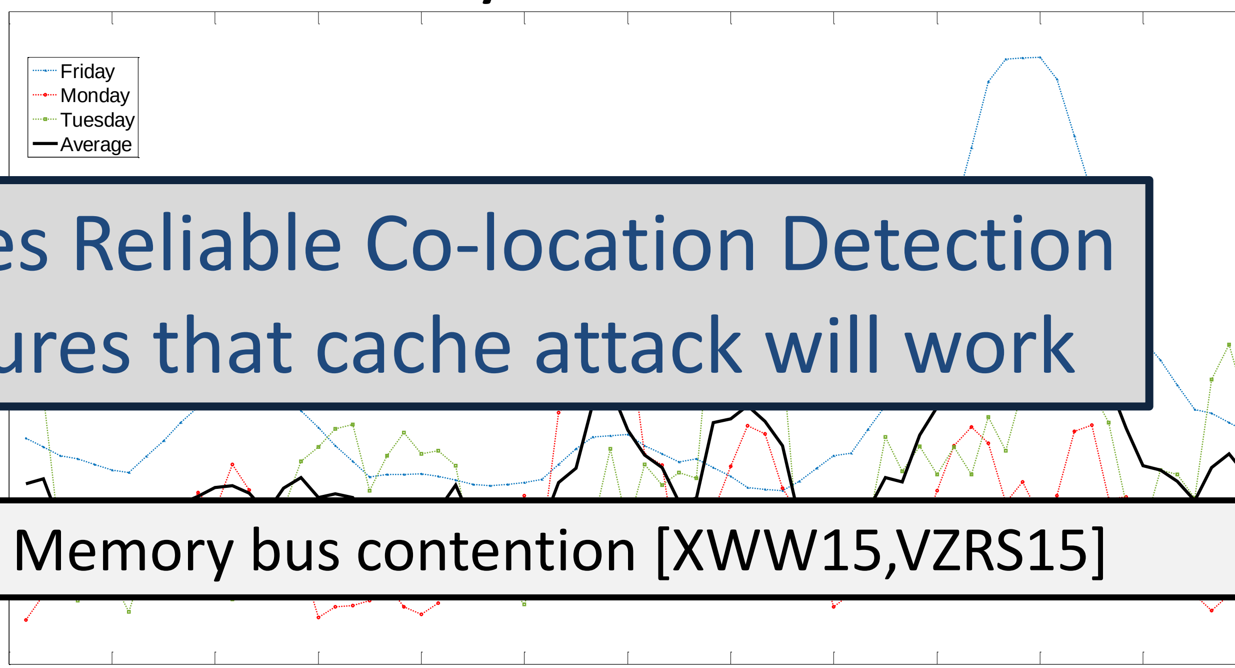
Co-Location Attempt:
LLC Cache Accesses

\+ Works reliable and we know how to do it

\+ Difficult to block

\- Requires slice recovery

\- Noise?



- Gives Reliable Co-location Detection
- Ensures that cache attack will work

Alternative: Memory bus contention [XWW15,VZRS15]

[XWW15] XU, Z., WANG, H., AND WU, Z. *A measurement study on co-residence threat inside the cloud*. USENIX Security 15
[VZRS15] VARADARAJAN, V., ZHANG, Y., RISTENPART, T., AND SWIFT, M. *A placement vulnerability study in multi-tenant public clouds*. USENIX Security 15

Target Cryptosystem

- Libgcrypt 1.6.2 ‘s RSA implementation
  - RSA CRT with 2048 bit modulus size
  - Sliding window exponentiation (5 bits)
  - Message blinding to prevent chosen ciphertext attacks

Is this state-of-the-art?

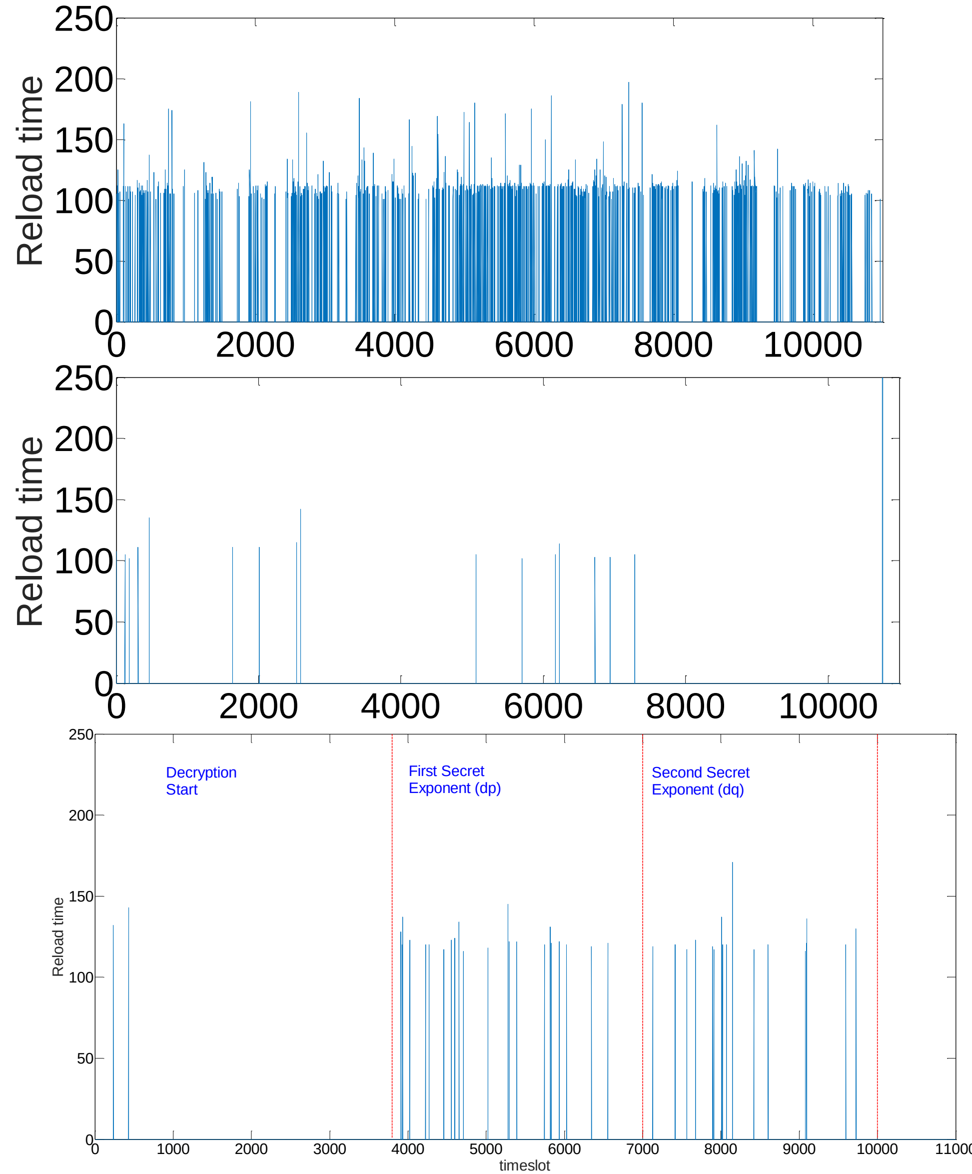
- Libgcrypt 1.6.3 (February 2015)
  - Table accesses now constant execution flow (no more cache games)

Attack on RSA-CRT Sliding Window

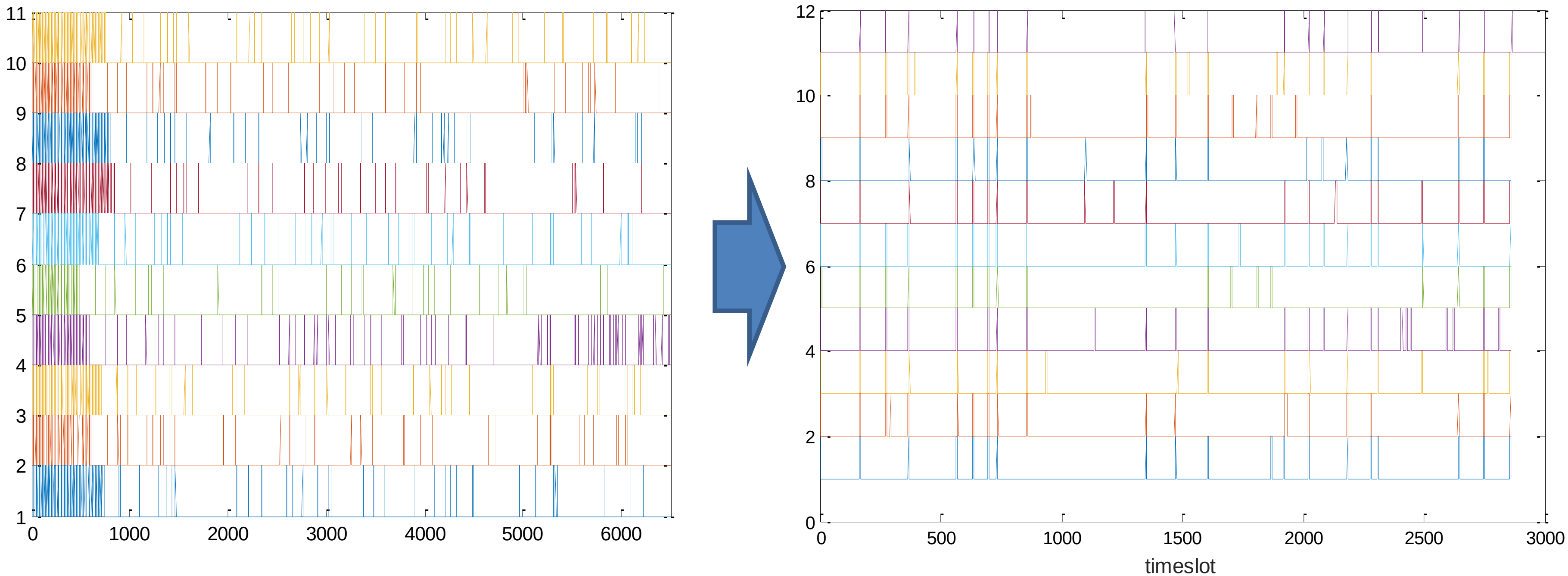
1. Find cache trace of sliding window multiplicands
2. Observe several exponentiations *to reduce noise*
3. Align and filter observations *to reduce noise*
4. Run error correcting key recovery to fix remaining noise errors

Identifying a Correct Cache Line

- 10x2048 cache lines
- Source code reveals approximate position
- Search through remaining choices
- Once found, repeat observations



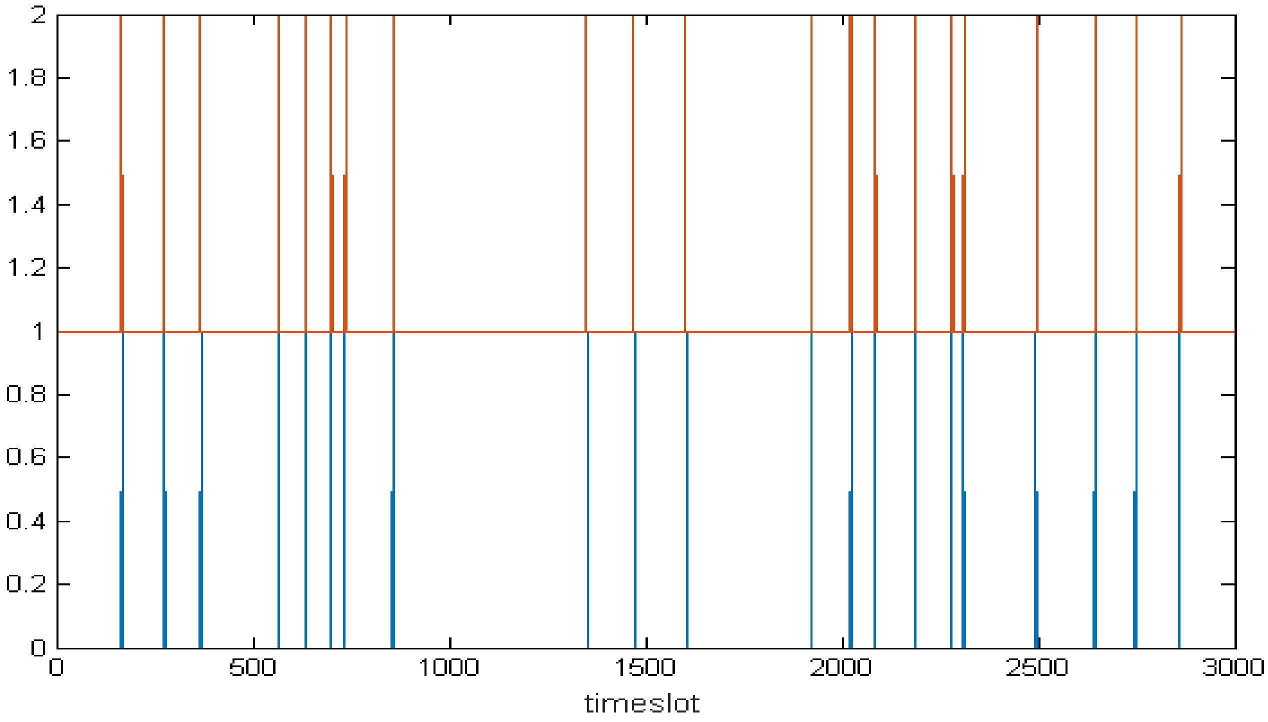
Processing Noisy Observations



Accesses to specific cache line during subsequent encryptions

1. Alignment to remove temporal shifts
2. Remove noise artifacts

After Processing and Alignment



- Correct (red) vs recovered (blue):

→little remaining noise

Final key recovery?

- Distance to table initialization reveals multiplicand value
- d must be recovered from noisy d_p and d_q

More details in: ia.cr/2015/898

Conclusions

- Cache Attacks in public clouds work
  - Noise and co-location need to be tackled
- Fully patched crypto libraries (at least major open source ones) are no longer vulnerable
- Countermeasures are still open problem: Many proposed, but cost overhead prohibitive?
- How about non-crypto code?

Thank you!

vernam.wpi.edu

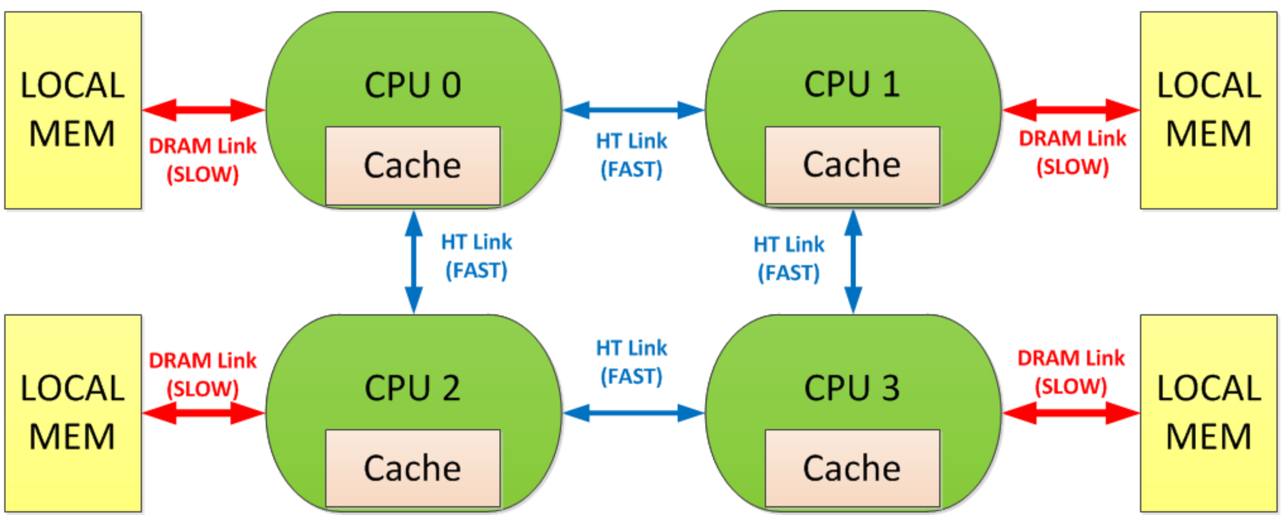
teisenbarth@wpi.edu

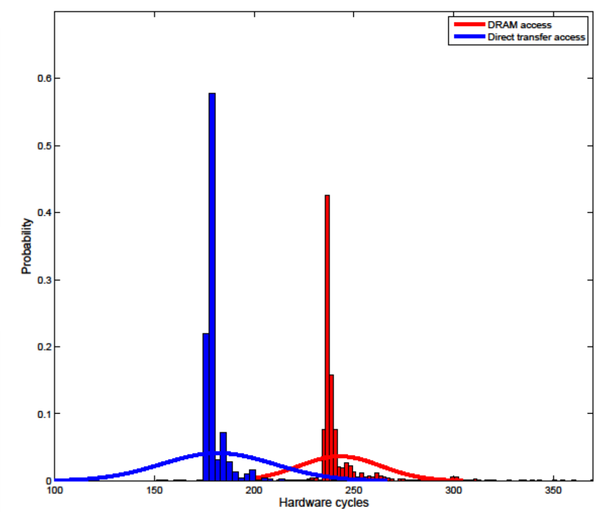
VERNAM Group
Security & Privacy @ WPI

WPI

Cross Processor Cache Attacks?

- Interprocessor Communication:
  Cache Coherence Protocols use fast direct links between processors:
- Faster than memory access → Timing behavior





[IES15] G Irazoqui and T Eisenbarth and B Sunar *Cross Processor Cache Attacks* ia.cr/2015/1155