nccgroup

Microarchitectural Attacks on TEEs

Keegan Ryan

Processor Security (x86)

Privilege

Ring 0	Kernel
Ring 1	Sometimes Guest OS
Ring 2	“Drivers”
Ring 3	Userland

Processor Security (x86)

Privilege

Ring -1	Hypervisor
Ring 0	Kernel
Ring 1	Sometimes Guest OS
Ring 2	“Drivers”
Ring 3	Userland

Processor Security (x86)

Privilege

Ring -2	System Management Mode
Ring -1	Hypervisor
Ring 0	Kernel
Ring 1	Sometimes Guest OS
Ring 2	"Drivers"
Ring 3	Userland

Processor Security (x86)

Privilege

Ring -3	Intel ME, Intel AMT
Ring -2	System Management Mode
Ring -1	Hypervisor
Ring 0	Kernel
Ring 1	Sometimes Guest OS
Ring 2	“Drivers”
Ring 3	Userland

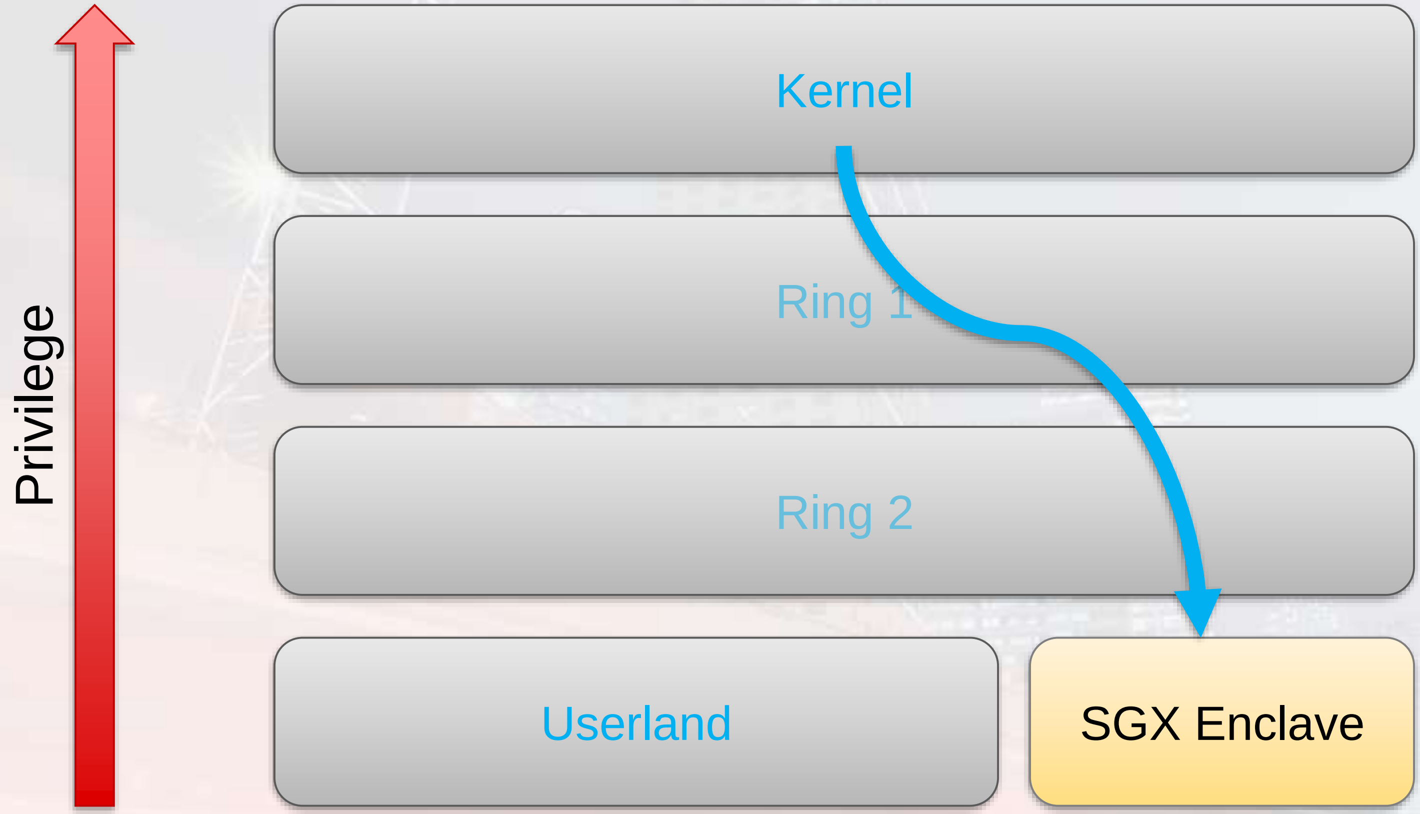
Processor Security (x86)

Privilege

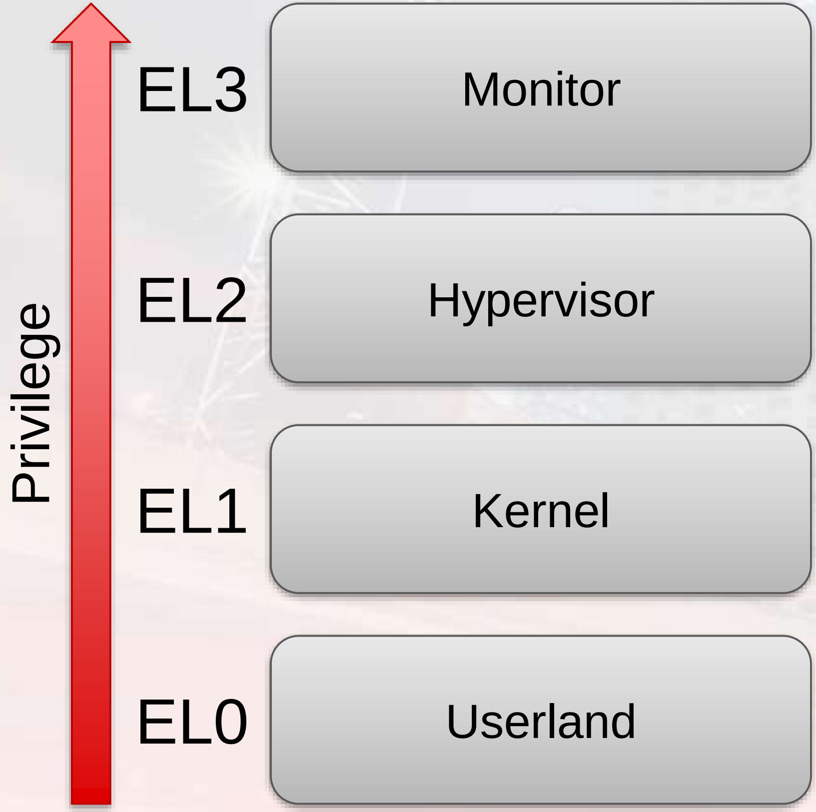
Ring -3	Intel ME, Intel AMT
Ring -2	System Management Mode
Ring -1	Hypervisor
Ring 0	Kernel
Ring 1	Sometimes Guest OS
Ring 2	“Drivers”
Ring 3 SGX	Userland + SGX Enclaves

Processor Security (x86)

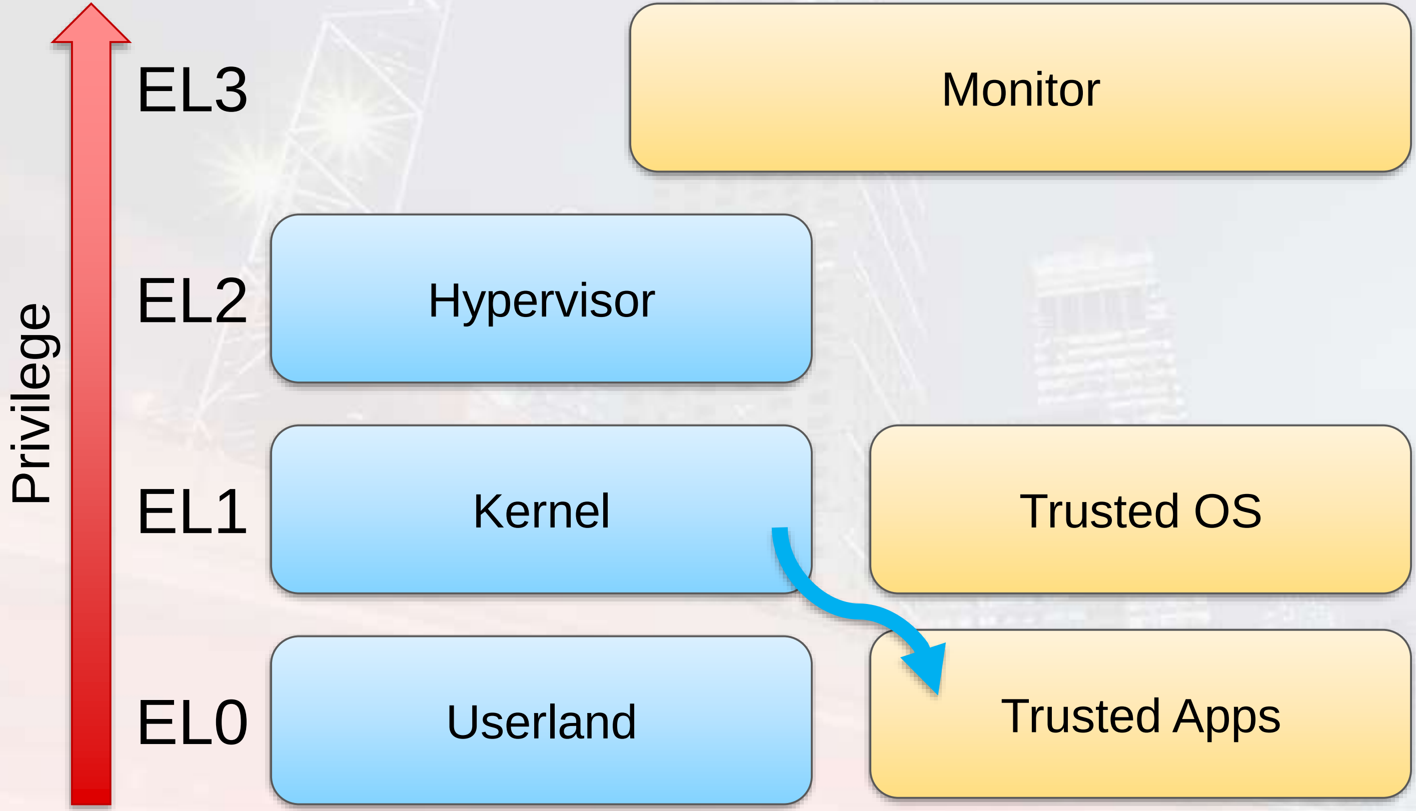
Processor Security (x86)



Processor Security (ARMv8)



Processor Security (ARMv8)



Can we use privileged modes to attack TEEs in SGX and TrustZone?

CLKSCREW (TrustZone)

(Tang, Sethumadhavan & Stolfo 2017)

- Non-secure OS manages energy consumption
- This is important for performance
- Change the energy management parameters on target core
- Undervolt and overclock to induce faults in secure world
- Used to extract private keys and bypass code signing

Side-Channel Attacks

- CLKSCREW is an active attack. What about passive attacks?
- In many SGX and TrustZone implementations, trusted and untrusted code share underlying hardware
- Cache activity from trusted code might influence behavior of untrusted code
- Observe these side effects to infer secrets in TEE

Intel on Side-Channel Attacks

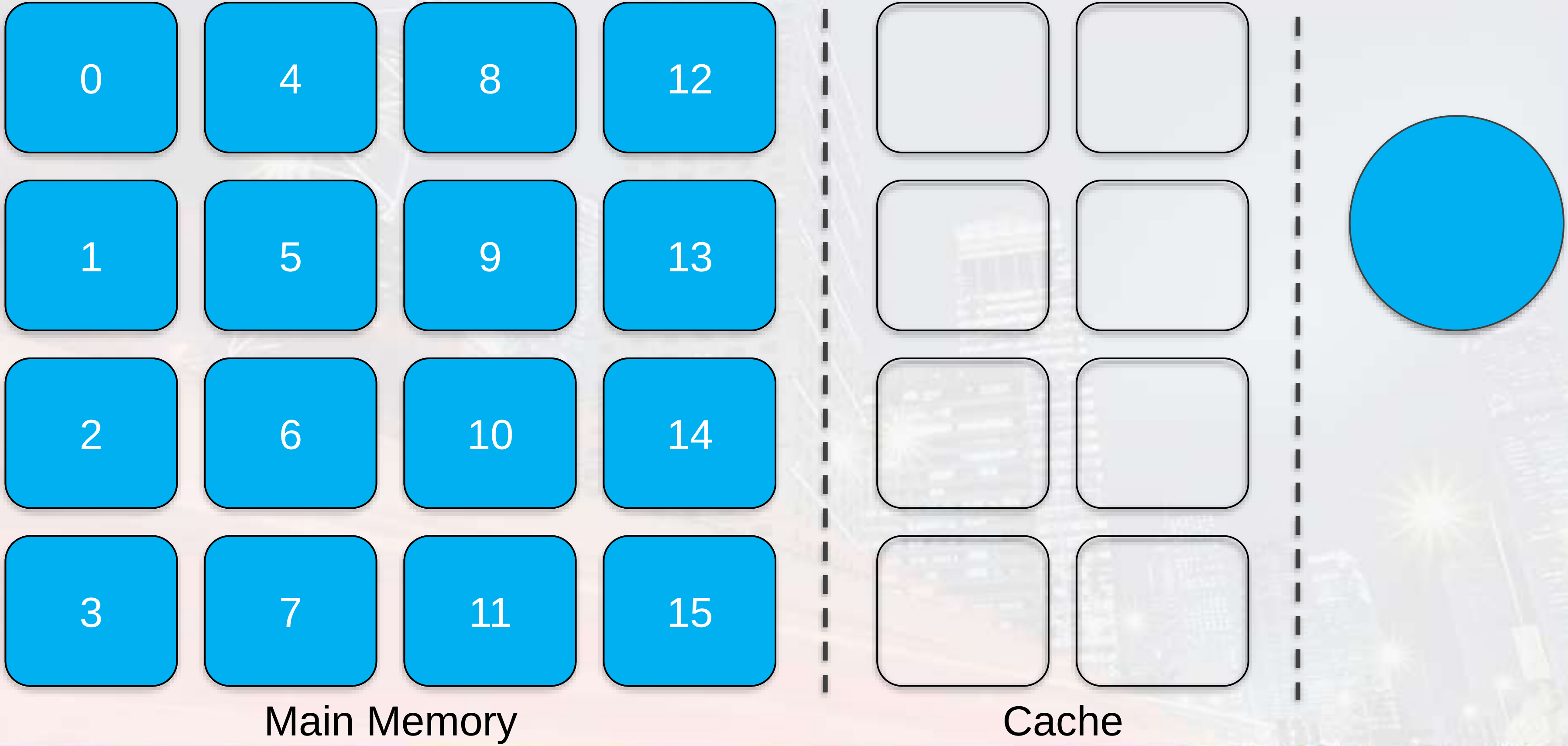
(Intel 2015)

“SGX does not defend against this adversary”

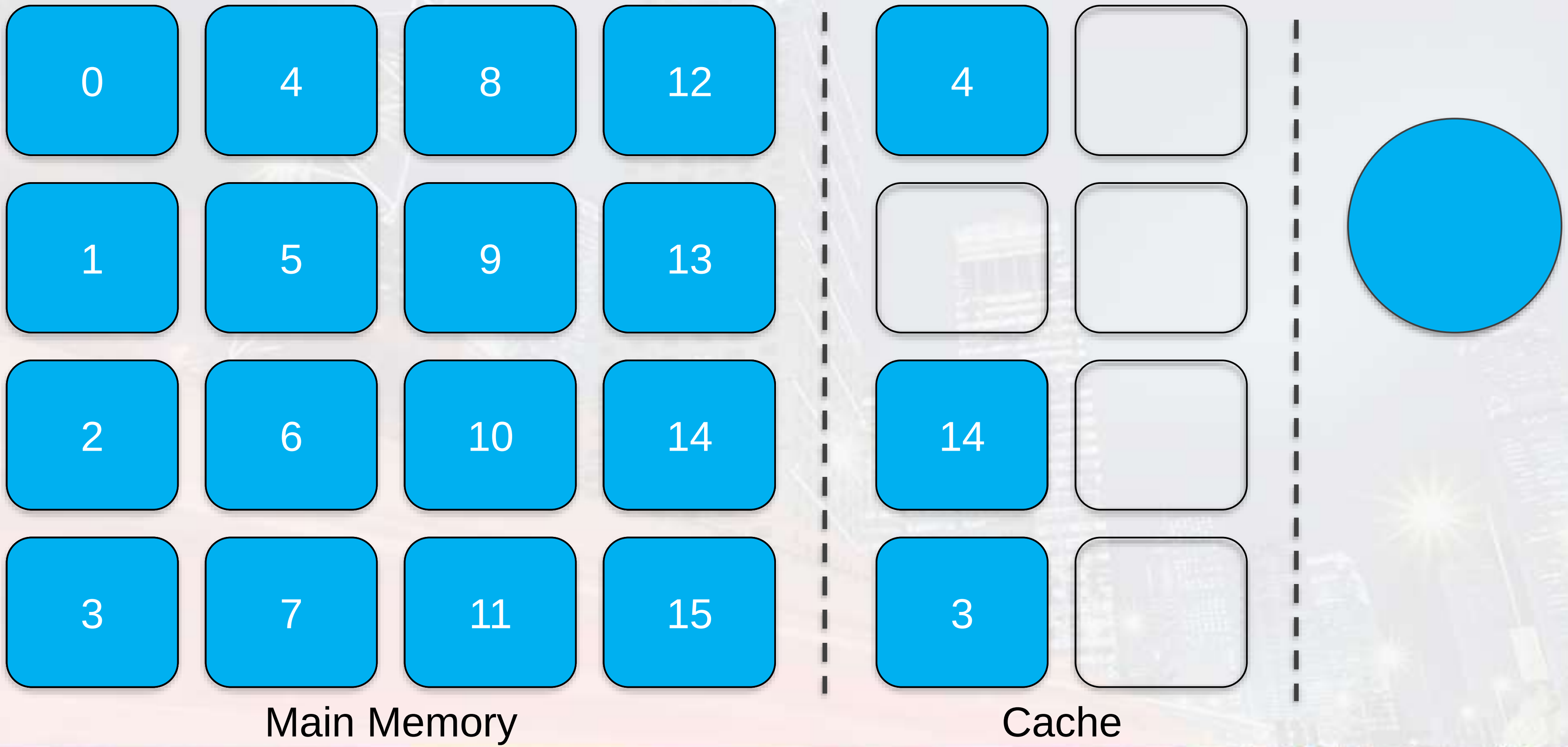
- Intel

Cache Attacks

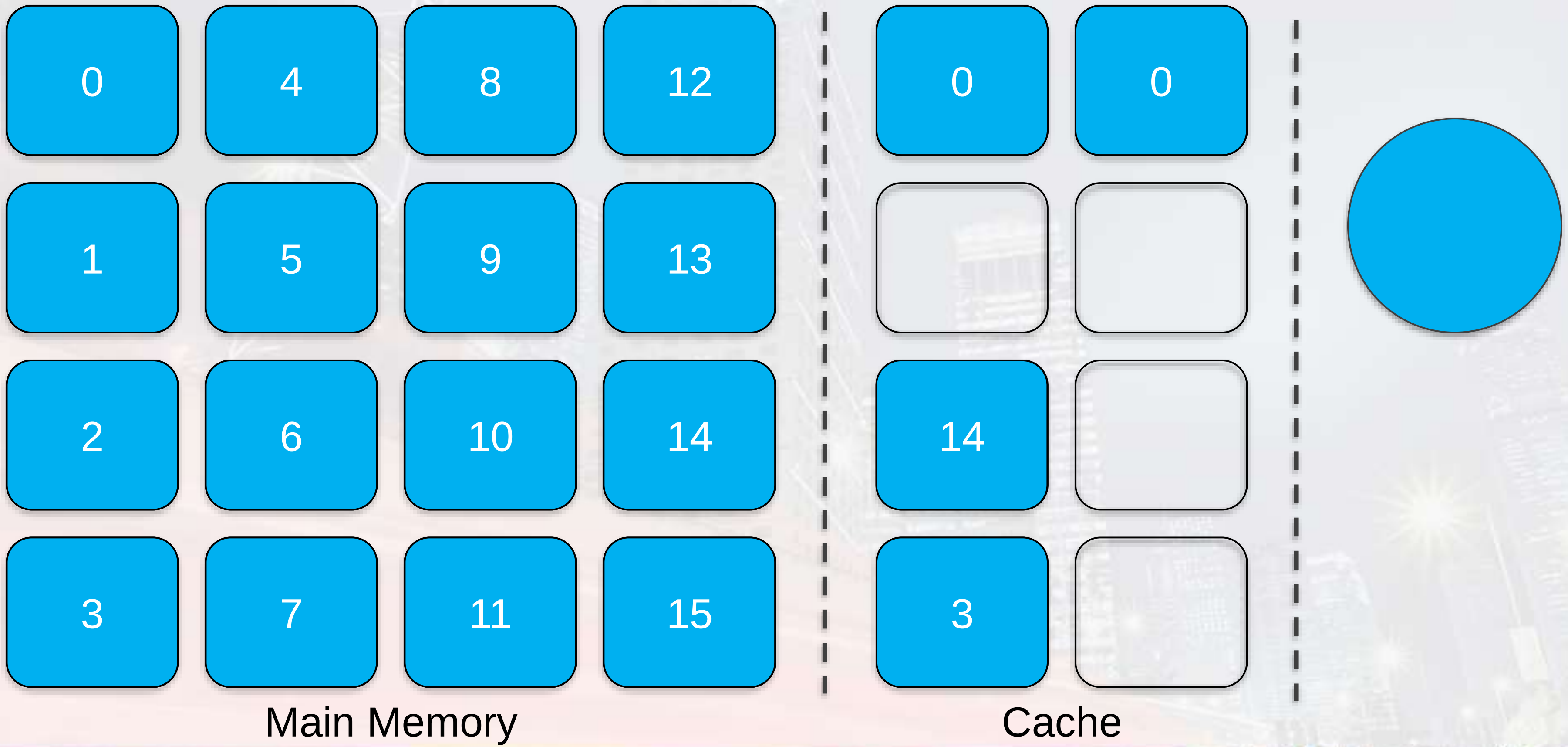
Cache Attacks



Cache Attacks

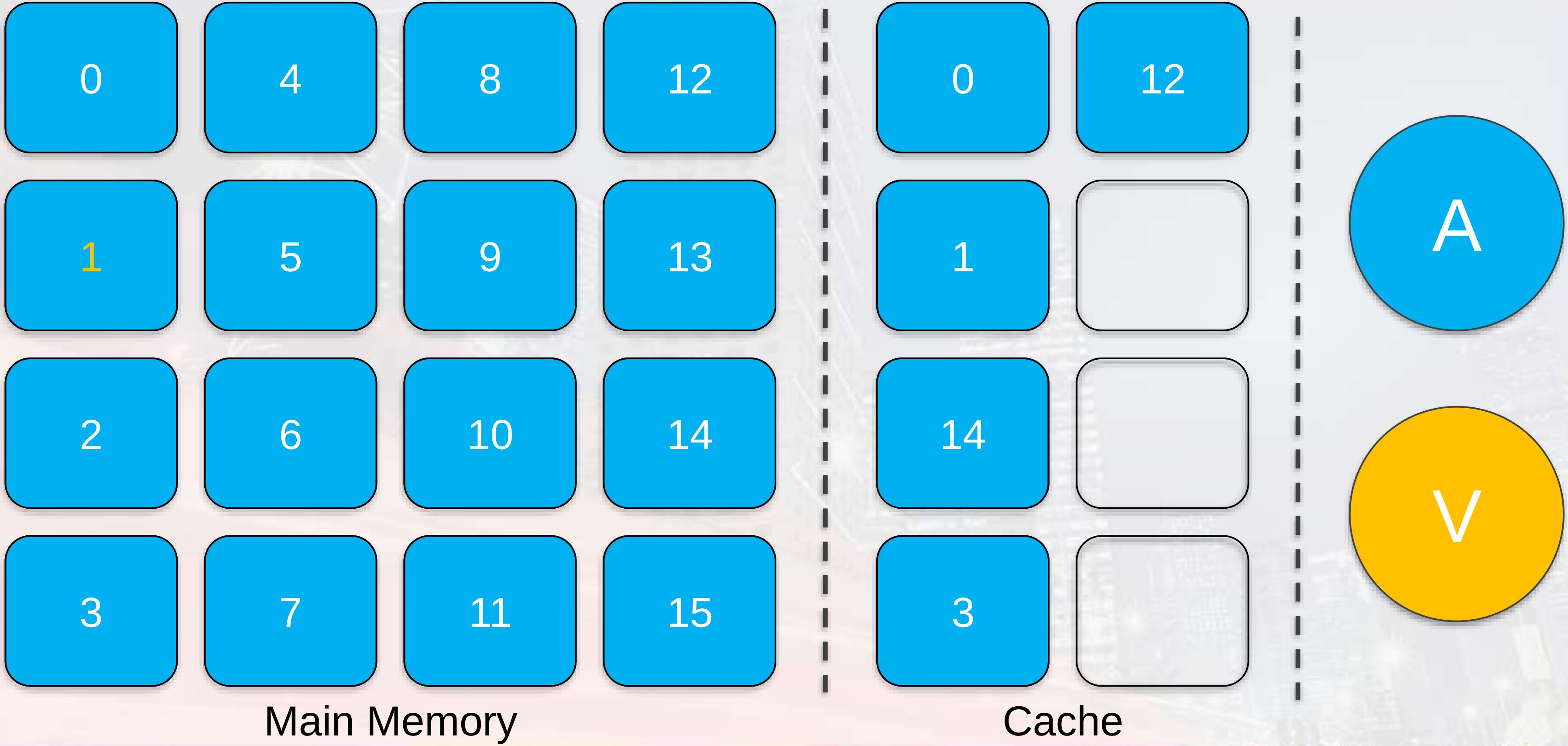


Cache Attacks



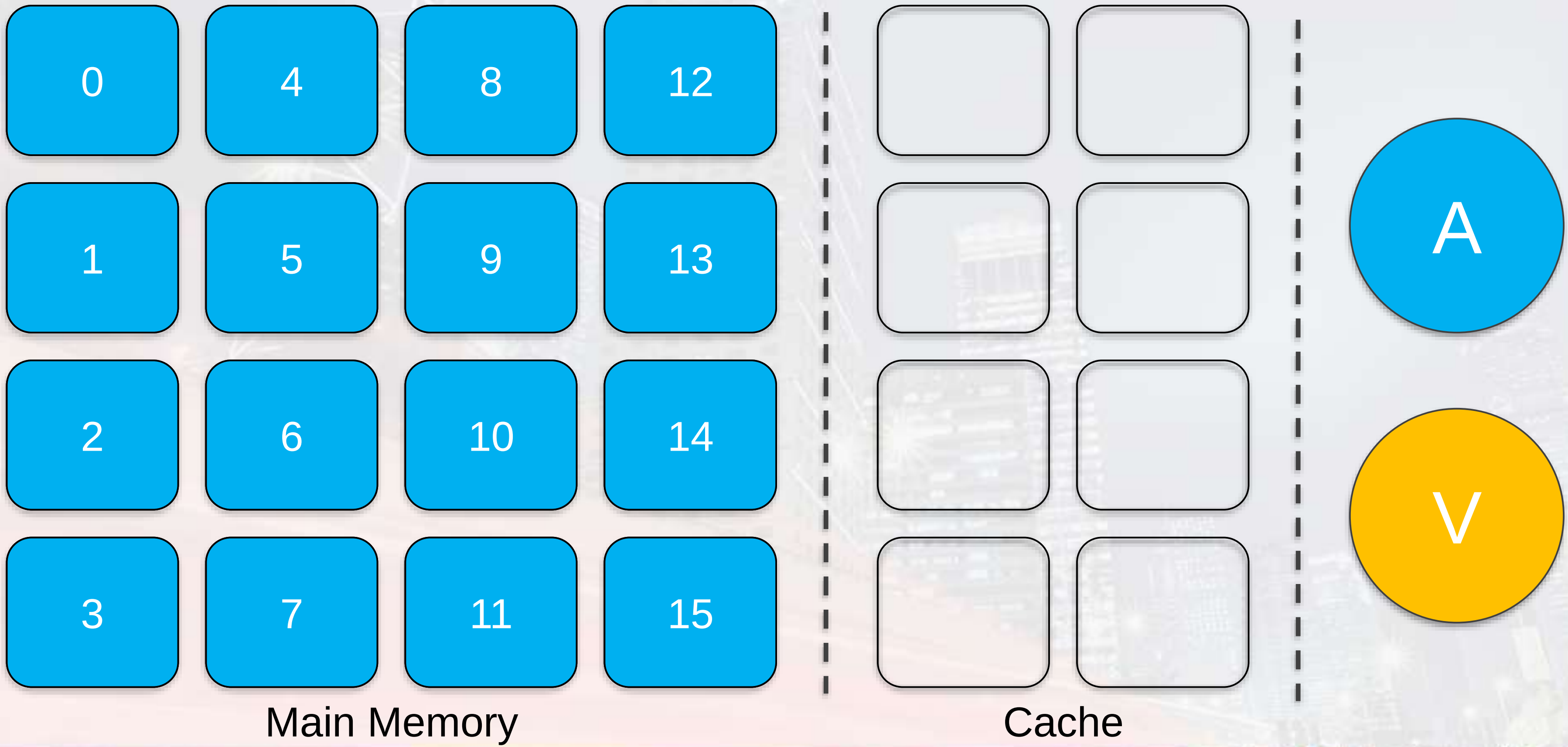
Cache Attacks (Flush+Reload)

(Yarom & Falkner 2014)



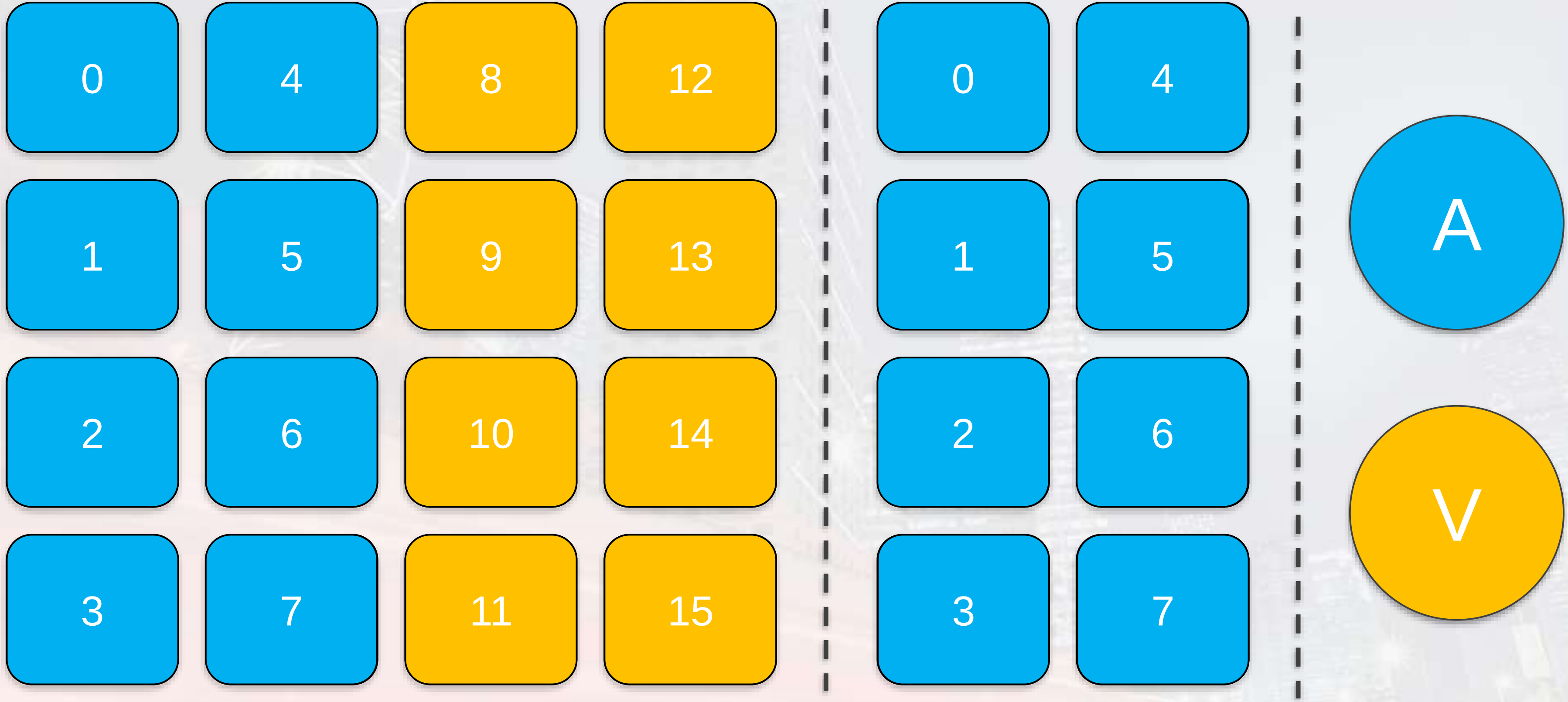
Cache Attacks (Flush+Flush)

(Gruss, Maurice, Wagner & Mangard 2016)



Cache Attacks (Prime+Probe)

(Osvik, Shamir & Tromer 2006)



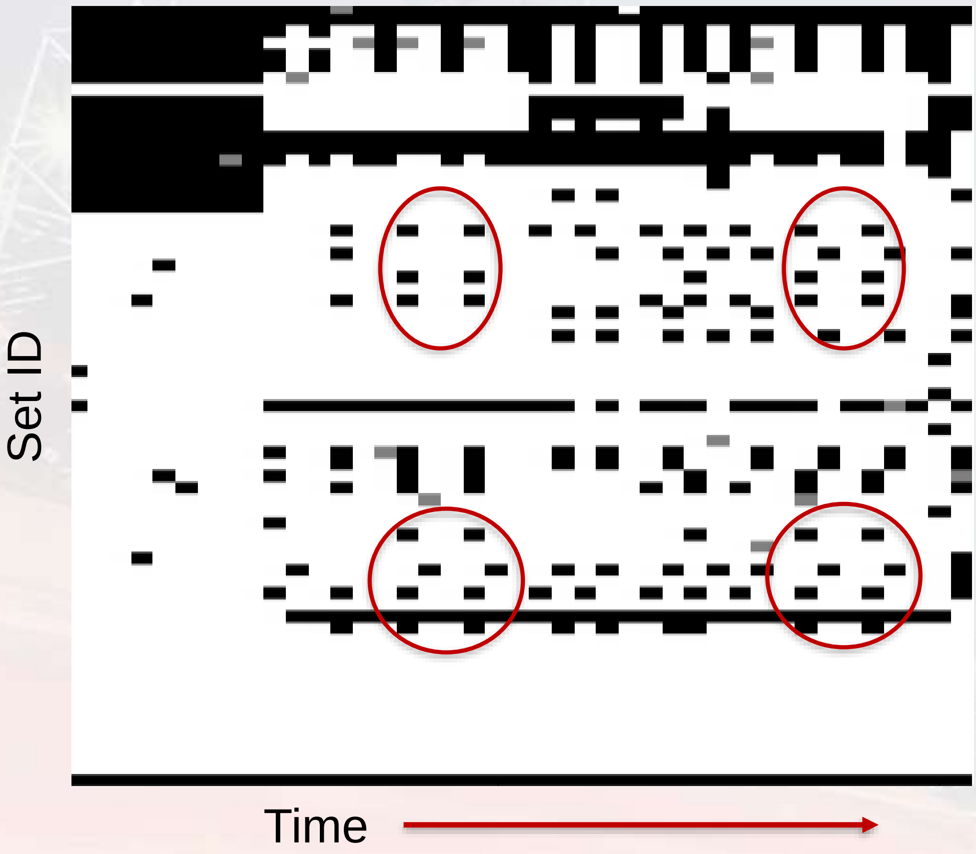
Main Memory

Cache

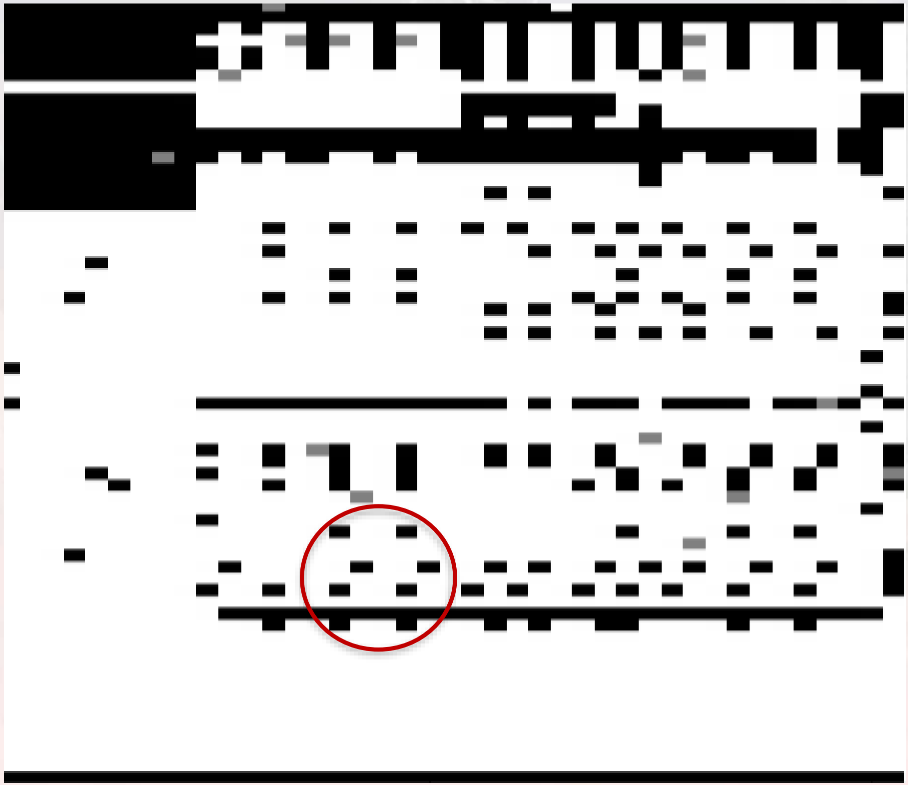
Cache Attacks

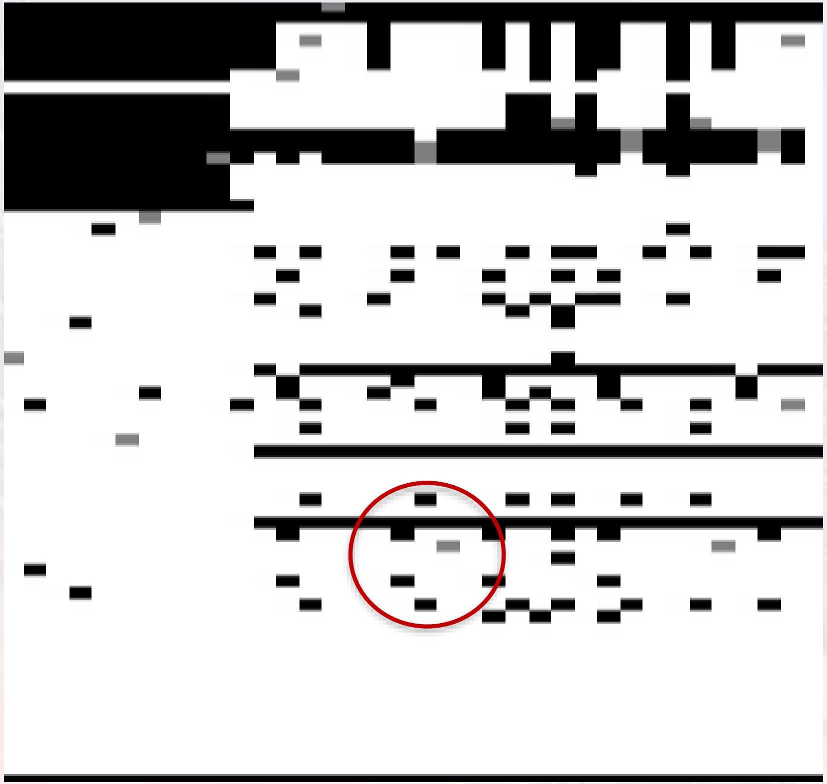
- Caches are important for good performance
- Use time difference to infer how the victim uses the cache
- Can exploit this when memory is shared (F+R, F+F) or not shared (P+P)
- We know *where* the victim is looking, but not *what* they see

Repeated AES Prime+Probe



Repeated AES Prime+Probe





Cache Attacks

What makes a cache attack powerful?

- Spatial Resolution
- Temporal Resolution
- Noise

The better these values, the more likely we can extract secrets

Attacks on SGX and TrustZone

Controlled-Channel Attacks (SGX)

(Xu, Cui & Peinado 2015)

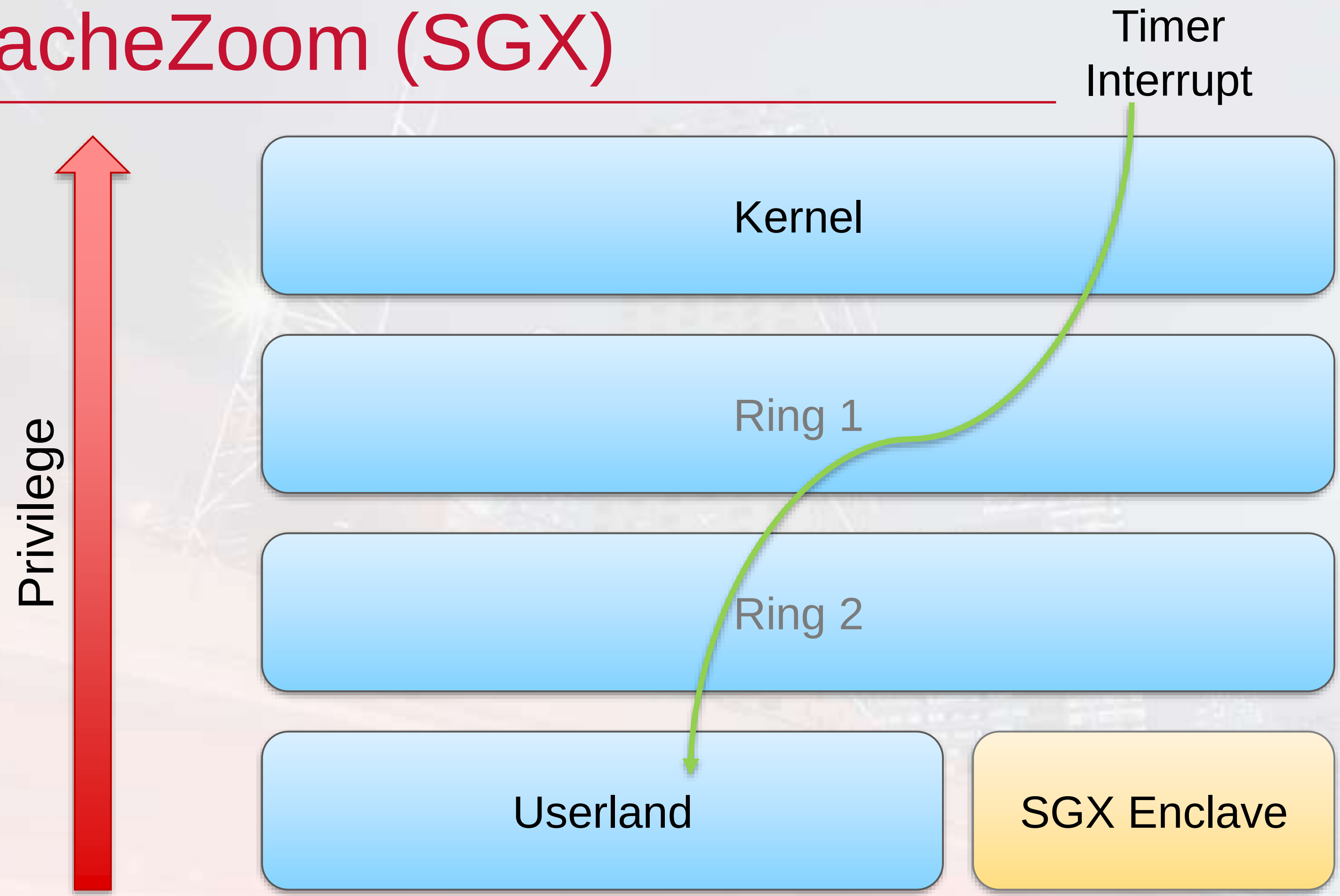
Controlled-Channel Attacks (SGX)

(Xu, Cui & Peinado 2015)

- Untrusted Operating System handles page faults
- OS learns base address of accessed page
- OS unmaps all other pages and resumes enclave until another page fault occurs
- General attack

Spatial Resolution: 4096 bytes
Temporal Resolution: As fast as new pages are accessed
Noise: None

CacheZoom (SGX)



CacheZoom (SGX)

(Moghimi, Irazoqui & Eisenbarth 2017)

- Prime+Probe attack on L1 data cache
- Uses timer interrupts to interleave attack process with victim enclave
- Can also be extended to L1 instruction cache
- General attack

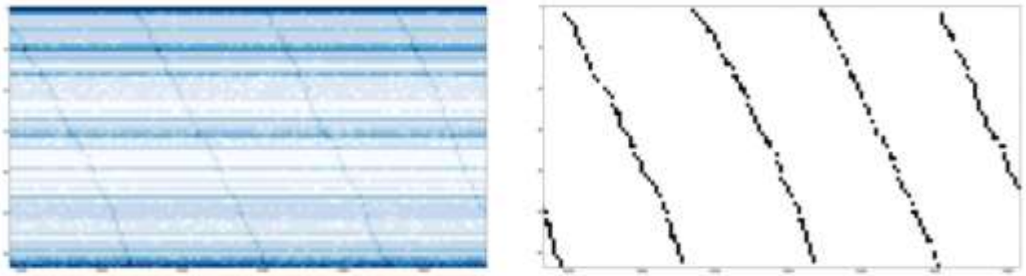


Fig. 3: Cache hit map before (left) and after (right) filtering for context switch noise. Enclave memory access patterns are clearly visible once standard noise from context switch has been eliminated

Spatial Reso

Temporal Re

Noise: Low

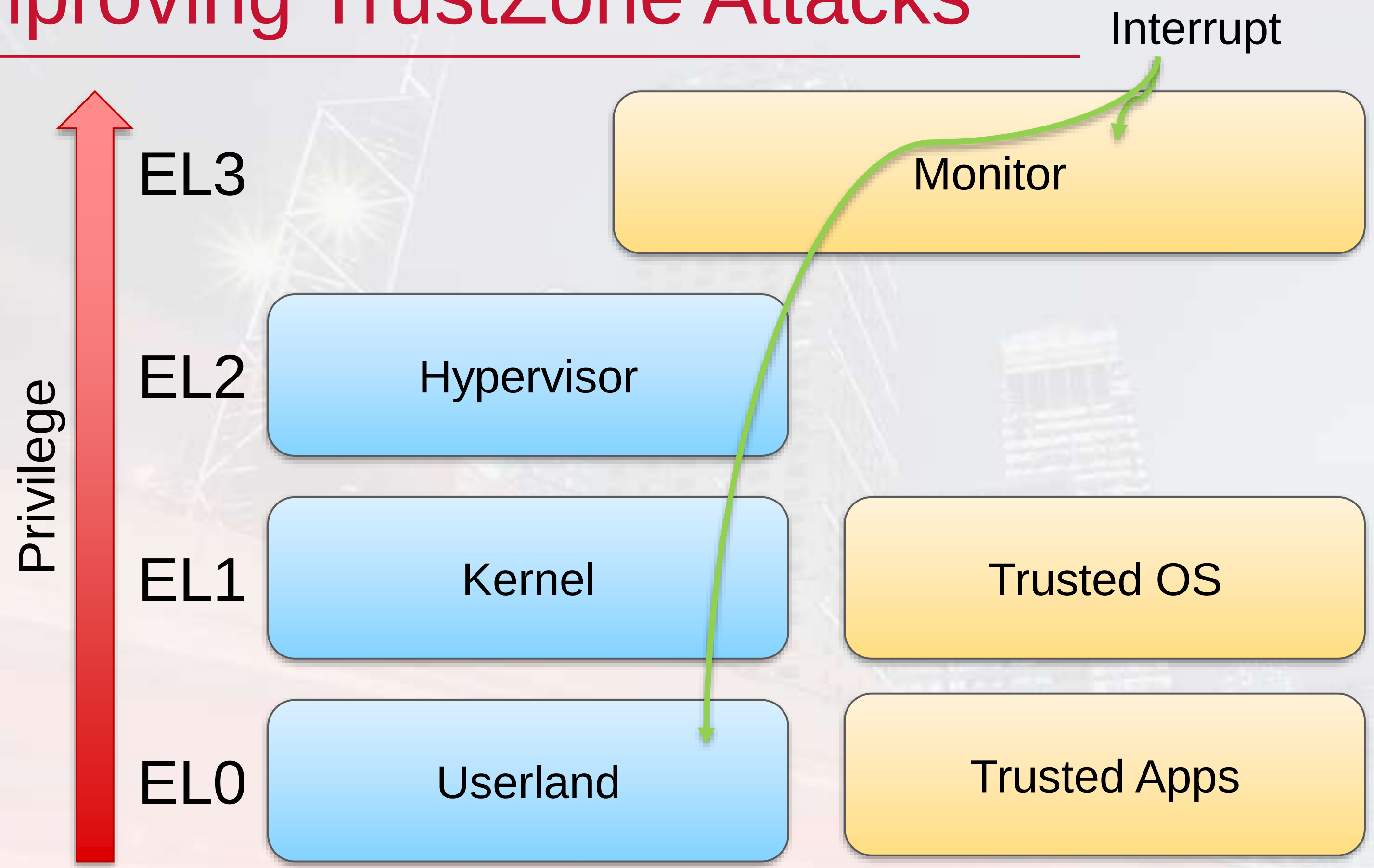
TruSpy (TrustZone)

(Zhang, Sun, Shands, Lou & Hou 2016)

- Userland Prime+Probe style attack on L1D
- Primes, then does full AES encryption, then probes once
- “Secure world is protected … and is not interruptible”
- Uses statistics to recover key from noise

Spatial Resolution: 64 bytes
Temporal Resolution: One measurement per execution
Noise: Noisy measurements from userland

Improving TrustZone Attacks



Improving TrustZone Attacks

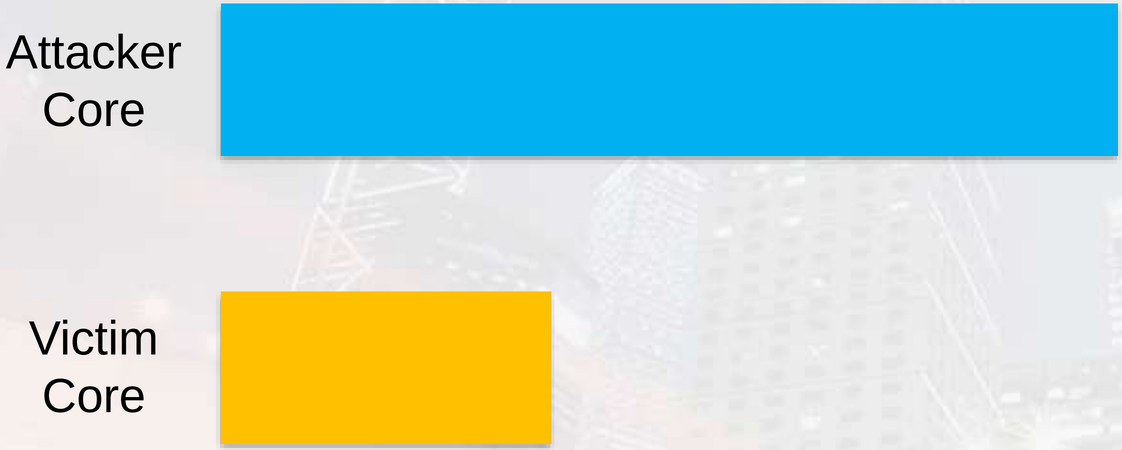
Idea: Use a second core to send interrupts to the core executing the trusted app

Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution

Noise: Noisy measurements from userland

Improving TrustZone Attacks

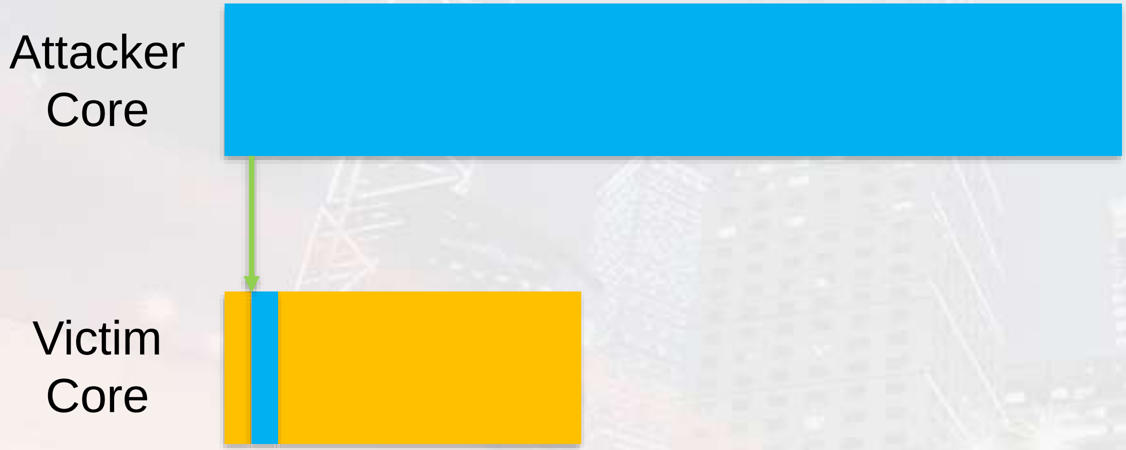


Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution

Noise: Noisy measurements from userland

Improving TrustZone Attacks



Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution

Noise: Noisy measurements from userland

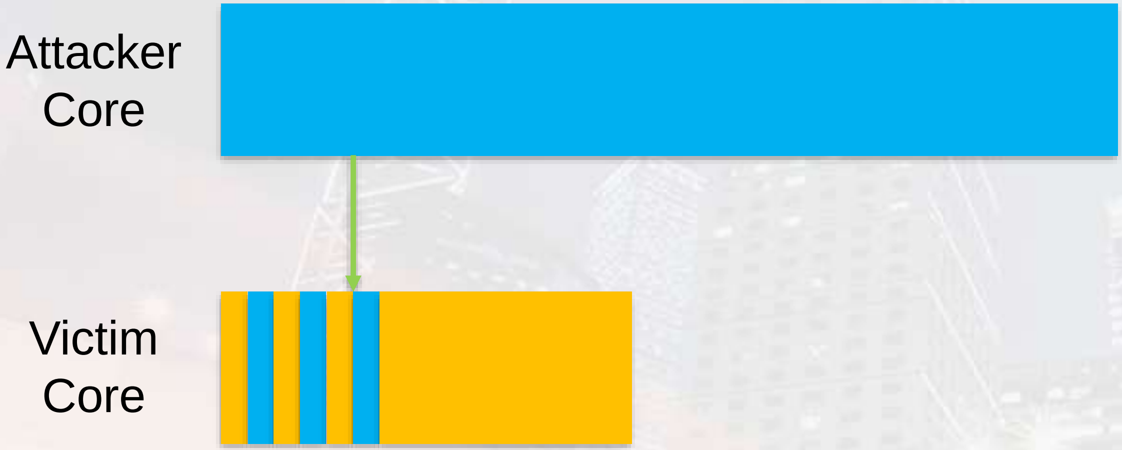
Improving TrustZone Attacks

Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution

Noise: Noisy measurements from userland

Improving TrustZone Attacks



Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution

Noise: Noisy measurements from userland

Improving TrustZone Attacks

Spatial Resolution: 64 bytes

Temporal Resolution: One measurement per execution Almost unlimited

Noise: Noisy measurements from userland

Improving TrustZone Attacks

How do we reduce the noise of measuring cache misses?

Idea: Use performance counters

Spatial Resolution: 64 bytes
Temporal Resolution: Almost unlimited
Noise: Noisy time based measurements

Performance Counters

(ARM 2012)
(ARM 2017)

- Allow developers to profile applications by counting cache hits, misses, and other events
- Require privileged access
- ARMv8 prevents non-secure code from counting events in secure world by default (ARMv7 doesn't)
- Can still use it for Prime+Probe attack
- "Counting events is never prohibited in Non-secure state"

- ARM

Spatial Resolution: 64 bytes

Temporal Resolution: Almost unlimited

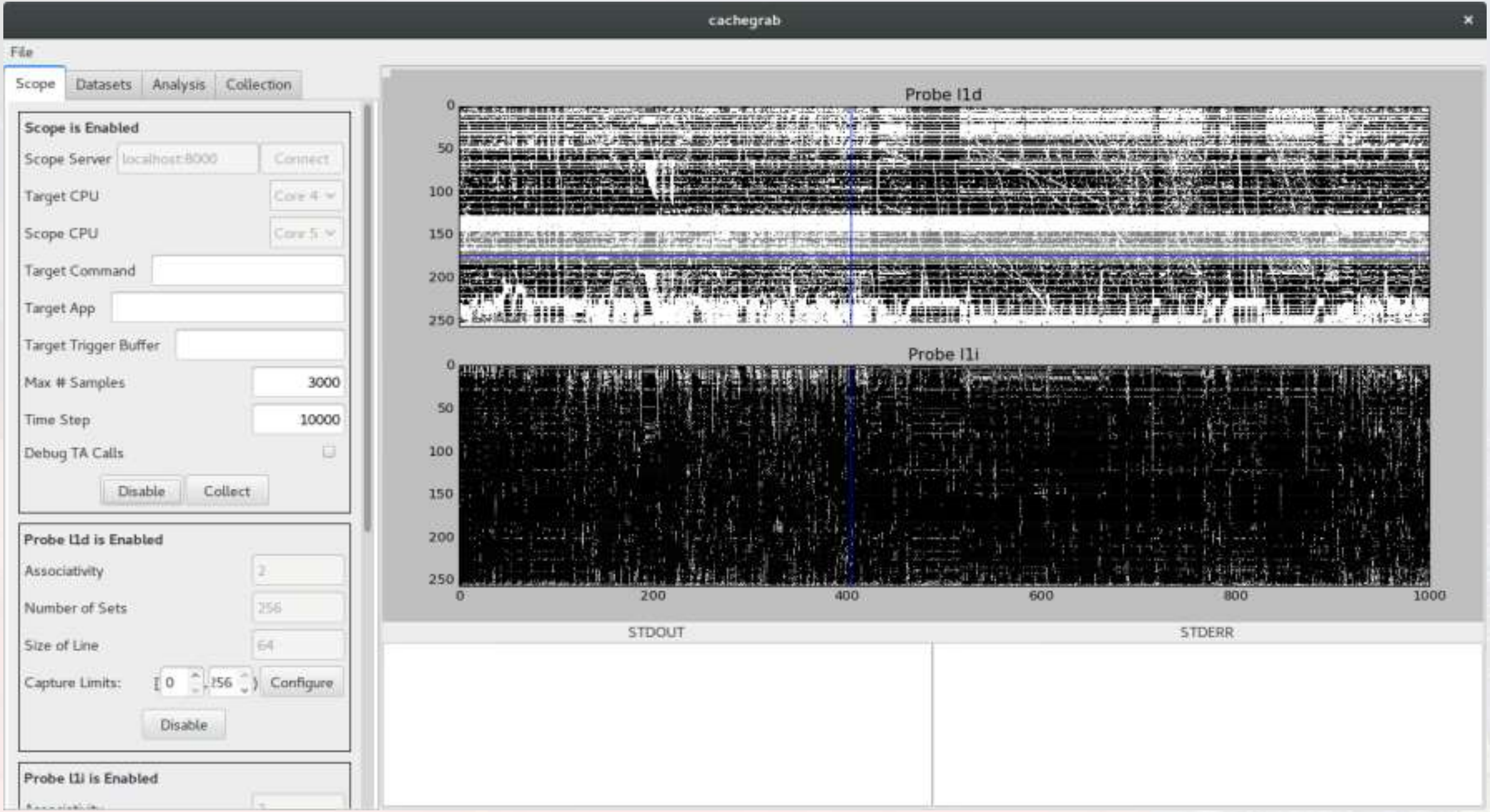
Noise: Virtually None

Noise: Noisy time based measurements

Implementing the Attacks

Cachegrab

- Goal: Implement these attacks on TrustZone
- Non-secure OS is usually Linux
- Write a kernel module that uses performance counters and interrupts to execute the attacks
- Limit collection to secure world by hooking victim calls to TrustZone driver
- Result: synchronized trace for each attack


cachegrab
File
Scope
Datasets
Analysis
Collection
Scope is Enabled
Scope Server
localhost:8000
Connect
Target CPU
Core 4
Scope CPU
Core 5
Target Command
Target App
Target Trigger Buffer
Max # Samples
3000
Time Step
10000
Debug TA Calls
Disable
Collect
Probe L1d is Enabled
Associativity
2
Number of Sets
256
Size of Line
64
Capture Limits:
[0
256)
Configure
Disable
Probe L1i is Enabled
Probe l1d
0
50
100
150
200
250
Probe l1i
0
50
100
150
200
250
0
200
400
600
800
1000
STDOUT
STDERR

Can we get even better spatial resolution?

Branch Shadowing (SGX)

(Lee et al. 2016)

- Processor uses Branch Target Buffer (BTB) branch predictor
- BTB is similar to a cache, but doesn’t compare full address
- Attacker manipulates address of SGX enclave to cause collisions in the BTB
- Attacker and victim share contents of cache, like in F+R
- Use attacker branch mispredictions to infer about victim

Spatial Resolution: Individual branches
Temporal Resolution: Almost unlimited
Noise: None

Improving TrustZone Attacks

- Victim and Attacker don’t share entries in BTB
- Use Prime+Probe style attack
- Prime BTB by executing many attacker branches
- Victim executes branch, evicting attacker BTB entry
- Attacker re-executes branches, monitors for mispredictions
- 2048 sets in BTB, 16 byte granularity

Spatial Resolution: 16 bytes
Temporal Resolution: Almost unlimited
Noise: Virtually None

Countermeasures

(Gulati, Smith & Yu 2014)
(Xin 2017)

- Use separate hardware for sensitive operations
  - Apple SoCs
  - Pixel 2
- But not all applications will be run in these environments
- Write side-channel free software
  - Performance is often at odds with security
  - Trusted Execution Environments don’t protect it all
  - Microarchitectural attacks are powerful
  - It only takes one small error

(Intel 2015)

“[We do] not defend against this adversary”

- Intel

Thank You

Keegan Ryan
Keegan.Ryan@nccgroup.trust
@inf_0_

References

2015. Intel Software Guard Extensions (Intel SGX). https://software.intel.com/sites/default/files/332680-001.pdf. (2015). Accessed December 2017.

ARM Limited. “ARM Architecture Reference Manual: ARMv7-A and ARMv7-R edition.” 2012.

ARM Limited. “ARM Architecture Reference Manual: ARMv8, for ARMv8-A architecture profile.” Retrieved from https://static.docs.arm.com/ddi0487/bb/DDI0487B_b_armv8_arm.pdf. 2017.

Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." IACR Cryptology ePrint Archive 2016/086 (2016).

Gruss, Daniel, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. "Flush+Flush: a fast and stealthy cache attack." In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 279-299. Springer International Publishing, 2016.

References

Guanciale, Roberto, Hamed Nemati, Christoph Baumann, and Mads Dam. "Cache storage channels: Alias-driven attacks and verified countermeasures." In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 38-55. IEEE, 2016.

Gulati, Manu, Michael J. Smith, and Shu-Yi Yu. "Security enclave processor for a system on a chip." U.S. Patent 8,832,465, issued September 9, 2014.

Lee, Sangho, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. "Inferring fine-grained control flow inside SGX enclaves with branch shadowing." *arXiv preprint arXiv:1611.06952* (2016).

Moghimi, Ahmad, Gorka Irazoqui, and Thomas Eisenbarth. "Cachezoom: How SGX amplifies the power of cache attacks." *arXiv preprint arXiv:1703.06986* (2017).

Osvik, Dag Arne, Adi Shamir, and Eran Tromer. "Cache attacks and countermeasures: the case of AES." In *Cryptographers' Track at the RSA Conference*, pp. 1-20. Springer, Berlin, Heidelberg, 2006.

Tang, Adrian, Simha Sethumadhavan, and Salvatore Stolfo. "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management." (2017).

References

Xin, Xiaowen. "Lock it up! New hardware protections for your lock screen with the Google Pixel 2." Google Online Security Blog. November 14, 2017. Accessed December 22, 2017. https://security.googleblog.com/2017/11/lock-it-up-new-hardware-protections-for.html.

Xu, Yuanzhong, Weidong Cui, and Marcus Peinado. "Controlled-channel attacks: Deterministic side channels for untrusted operating systems." In *Security and Privacy (SP), 2015 IEEE Symposium on*, pp. 640-656. IEEE, 2015.

Yarom, Yuval, and Katrina Falkner. "FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack." In *USENIX Security Symposium*, pp. 719-732. 2014.

Zhang, Ning, Kun Sun, Deborah Shands, Wenjing Lou, and Y. Thomas Hou. "TruSpy: Cache Side-Channel Information Leakage from the Secure World on ARM Devices." *IACR Cryptology ePrint Archive* 2016 (2016): 980.