Introduction to cache attacks

Yuval Yarom

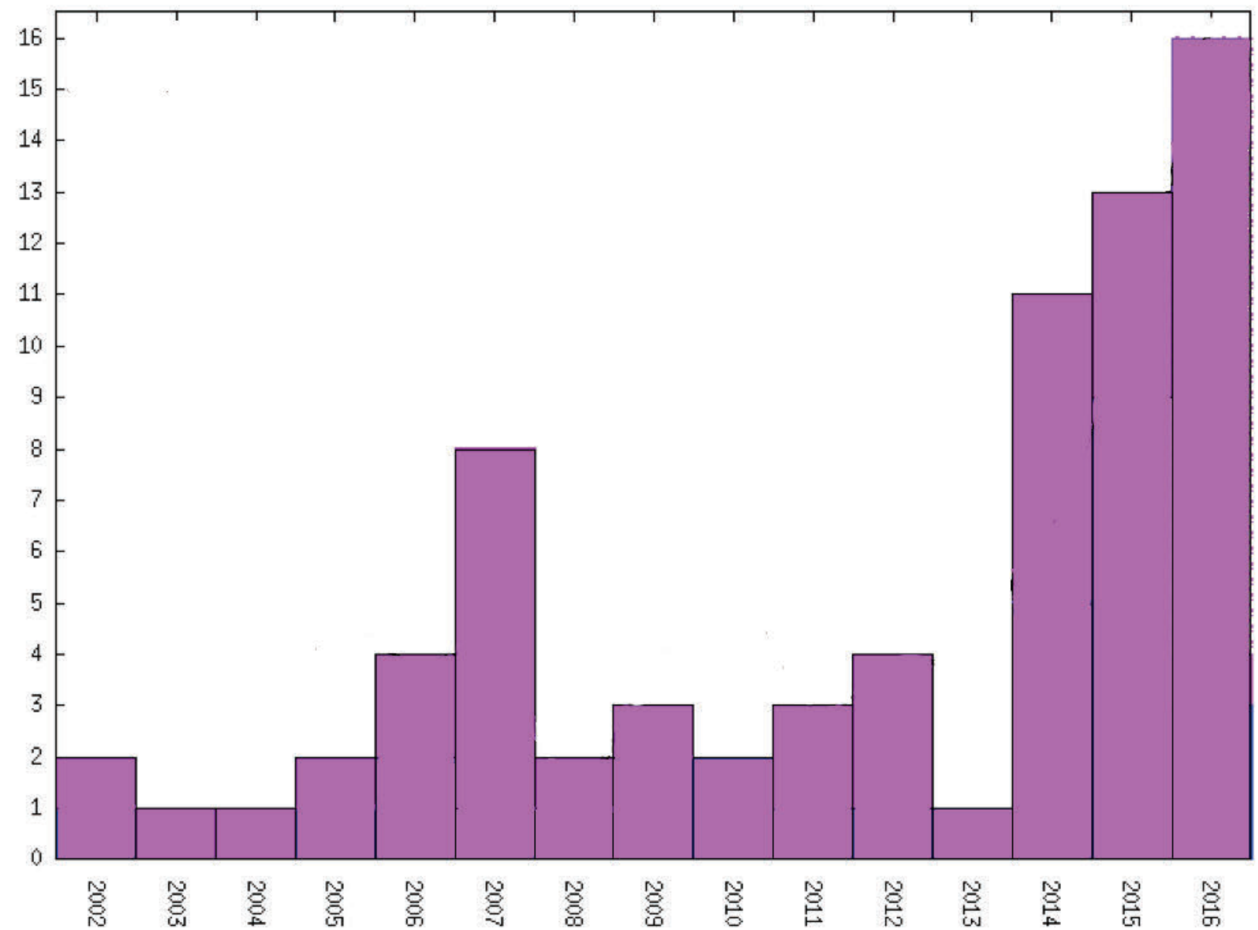
Summer School on Real-World Crypto and Privacy

6 June 2017

THE UNIVERSITY of ADELAIDE

DATA 61

Publications on Cache Attacks



Data from [GYCH16]

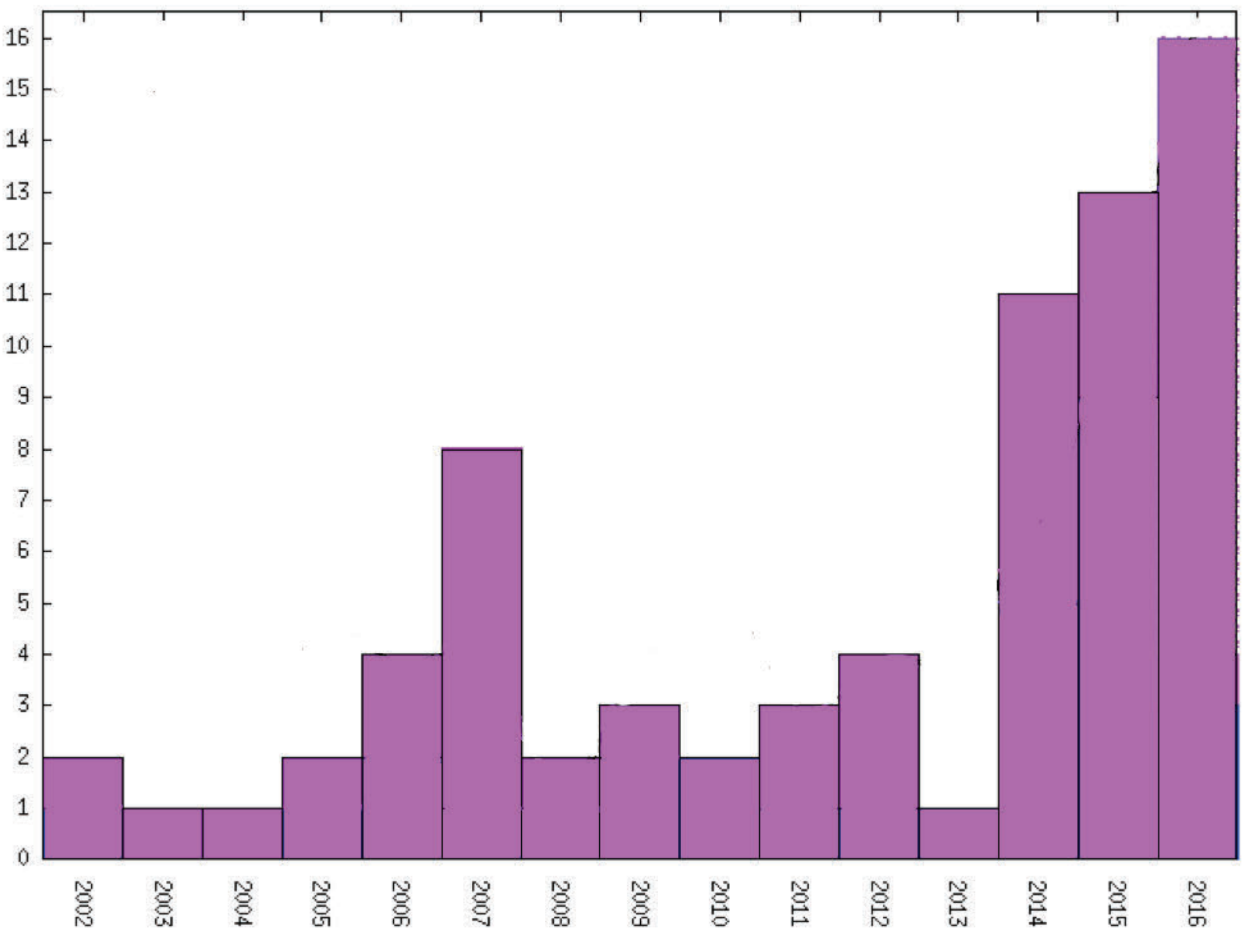
Some Targets

- RSA
  - Percival, 2005
  - Yarom and Falkner USENIX Security 2014
  - Bernstein, Genkin, Groot Bruinderink, Heninger, Lange, van Vredendaal and Yarom, CHES 2017
- AES
  - Osvik, Shamir and Tromer, CT-RSA 2006
  - Gullasch, Bangerter and Krenn, IEEE S&P 2011
  - Irazoqui, Inci, Eisenbarth and Sunar, RAID 2014
- ElGamal
  - Zhang, Juels, Reiter and Ristenpart, CCS 2012
  - Liu, Yarom, Ge, Heiser and Lee, IEEE S&P 2015

Some Targets

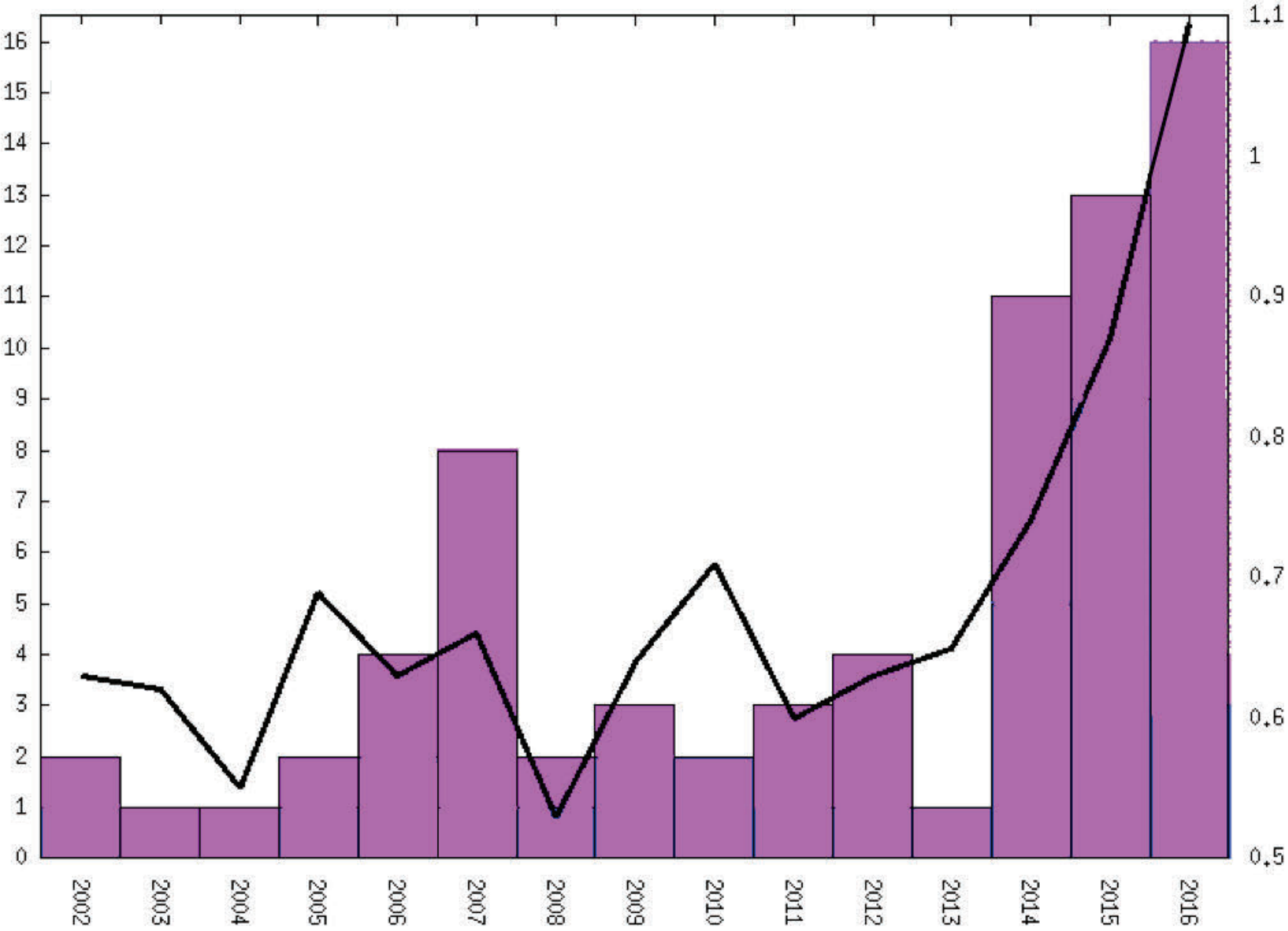
- DSA / ECDSA
  - Benger, van de Pol, Smart and Yarom, CHES 2014
  - Pereida, Brumley and Yarom, CCS 2016
  - Pereida and Brumley, USENIX Security 2017
- BLISS
  - Groot Bruinderink, Hülsing, Lange and Yarom, CHES 2016
  - Pessl, Groot Bruinderink and Yarom, ePrint 2017/490
- ECDH on Curve25519
  - Genkin, Valenta and Yarom, 2017 (in submission)

Hot Research Area



Data from [GYCH16]

Causes Global Warning



CPU vs. Memory

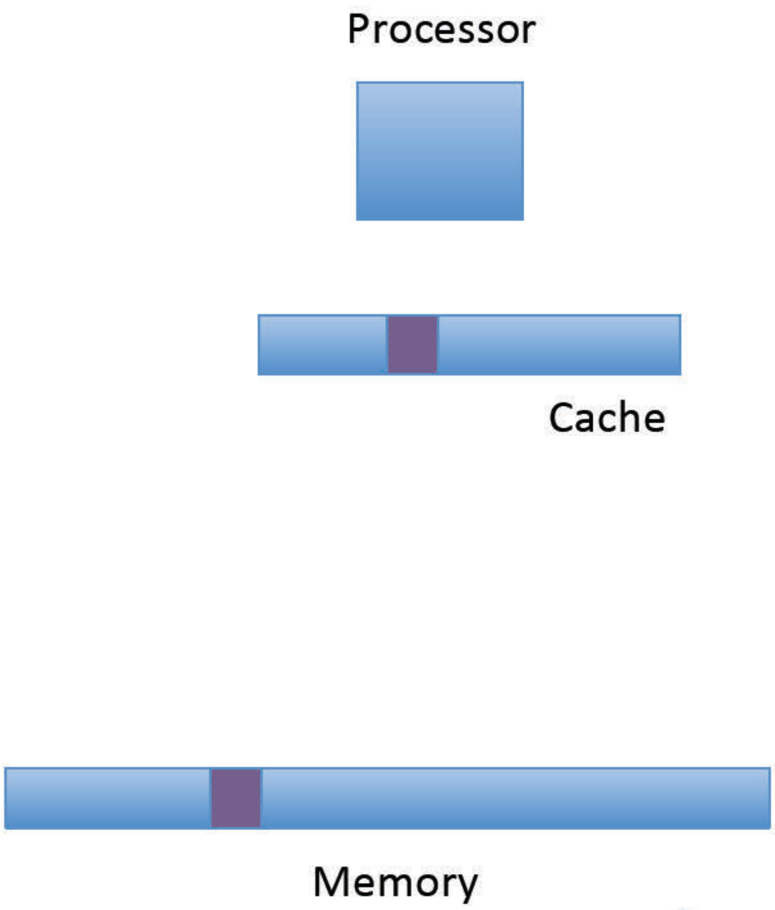
	Processor Speed	**Memory Latency**
	1 MHz	500 ns
	8*2600 MHz	63 ns

Bridging the gap

Cache utilises locality to bridge the gap

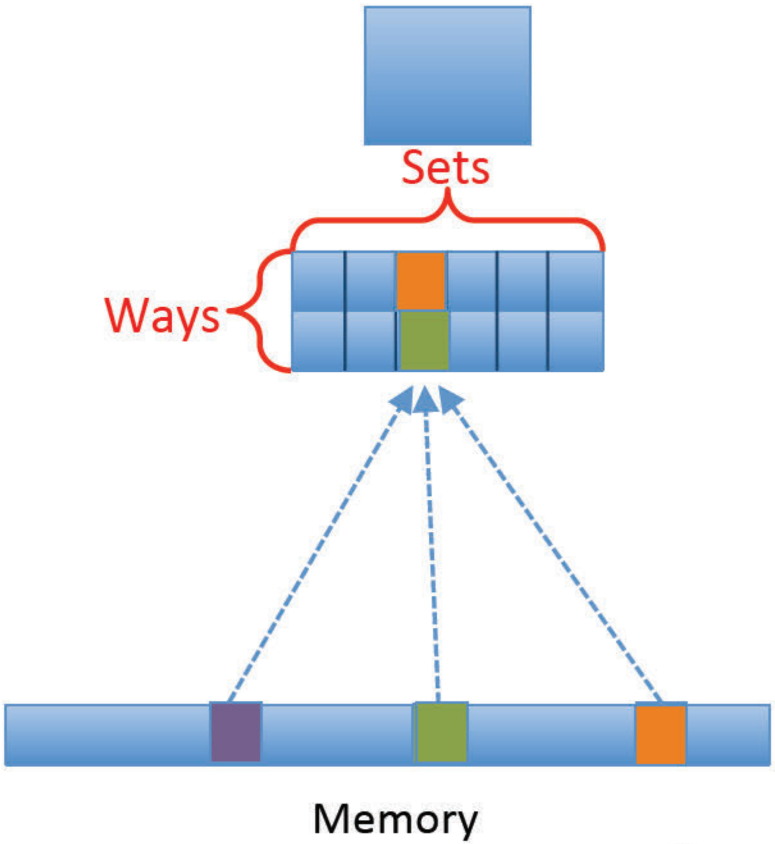
- Divides memory into *lines*
- Stores recently used lines

- In a *cache hit*, data is retrieved from the cache
- In a *cache miss*, data is retrieved from memory and inserted to the cache



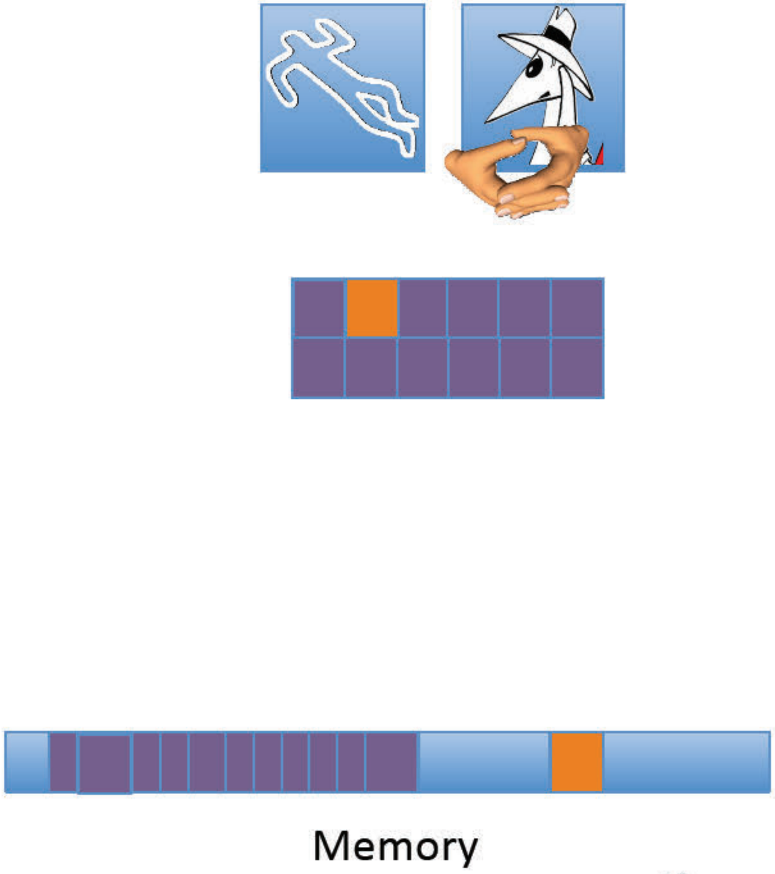
Set Associative Caches

- Memory lines map to *cache sets*. Multiple lines map to the same set.
- Sets consist of *ways*. A memory line can be stored in **any** of the ways of the set it maps to.
- When a cache miss occurs, one of the lines in the set is *evicted*.



The Prime+Probe Attack [Per05, OST06]

- Allocate a cache-sized memory buffer
- *Prime:* fills the cache with the contents of the buffer
- *Probe:* measure the time to access each cache set
  - Slow access indicates victim access to the set



Implementation Problems

- The observer effect
  - The spy also modifies the state of the cache
  - Cache thrashing
- Optimising compiler
  - Tries to mask memory latency
  - Delete dead code
- Hardware optimisations
  - Prefetch data predicted to be needed soon

Sample Victim: Data Rattle

```
volatile char buffer[4096];

int main(int ac, char **av) {
  for (;;) {
    for (int i = 0; i < 64000; i++)
      buffer[800] += i;

    for (int i = 0; i < 64000; i++)
      buffer[1800] += i;
  }
}
```

Mastik

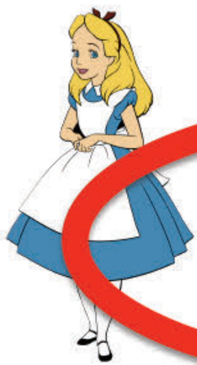
- A side channel toolkit
- Implements 6 attack techniques (more to follow)
  - Almost zero documentation, little testing
- Both API and command line utilities
- Available at
  http://cs.adelaide.edu.au/~yval/Mastik/

Demo

L1-Data Rattle

The RSA Encryption System

- The RSA encryption is a public key cryptographic scheme



$M = C^d \bmod N$

M

$C = M^e \bmod N$

Key Generation:

- Select random primes p and q
- Calculate $N = pq$
- Select a public exponent $e (=65537)$
- Compute $d = e^{-1} \bmod \varphi(N)$
- (N, e) is the public key
- (p, q, d) is the private key

GnuPG 1.4.13 Decryption

$x \leftarrow 1$
for $i \leftarrow |d|-1$ **downto** 0 do
  $x \leftarrow x^2 \bmod n$
  if $(d_i = 1)$ **then**
    $x = xC \bmod n$
  endif
done
return x

Example:
$11^5 \bmod 100 =$
$161{,}051 \bmod 100 = 51$

Operation	x	i	d_i

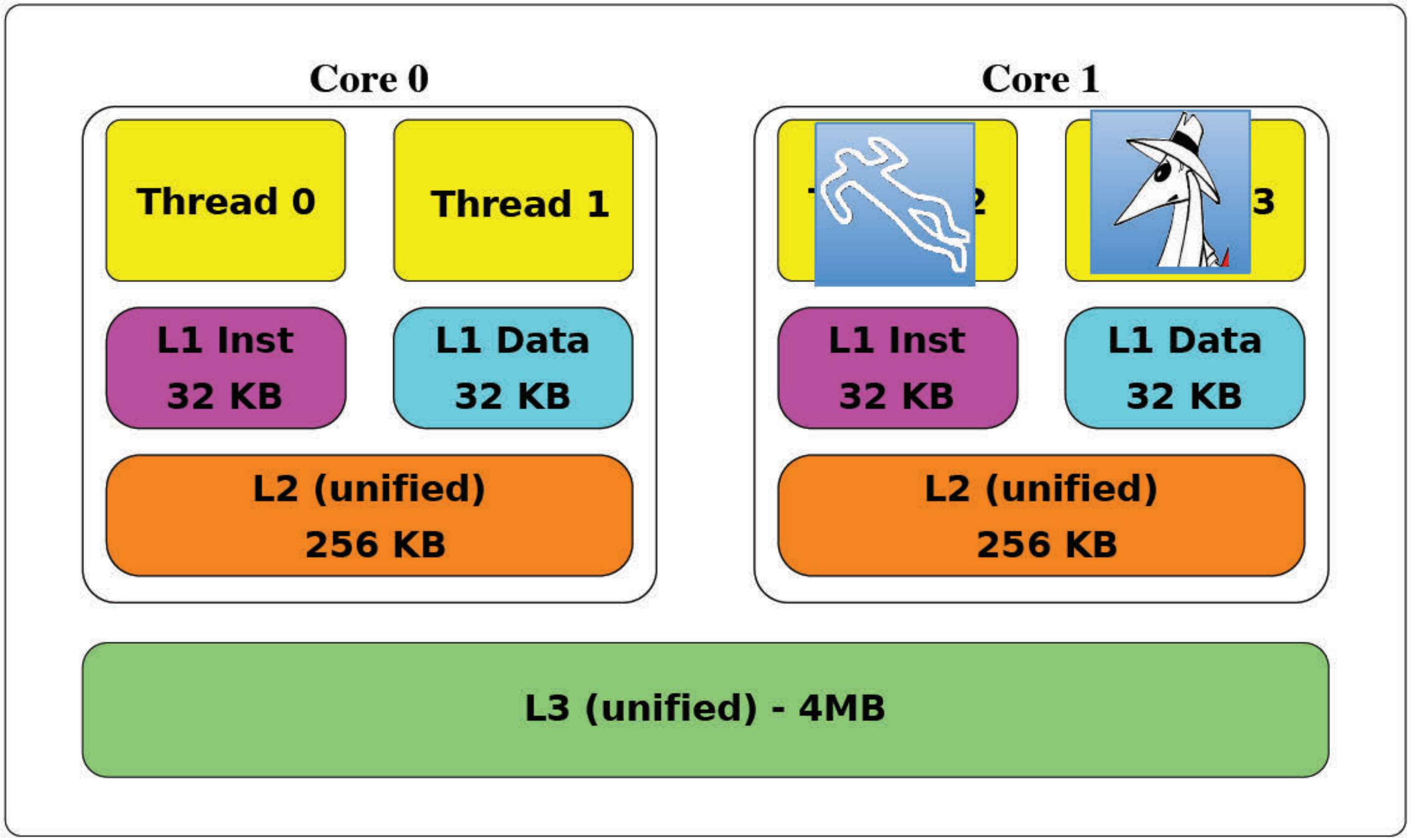
The private key is encoded in the sequence of operations !!!

Demo

Attacking GnuPG

Limitations

- Victim and spy run on the same core
  - Easy to mitigate in the operating system

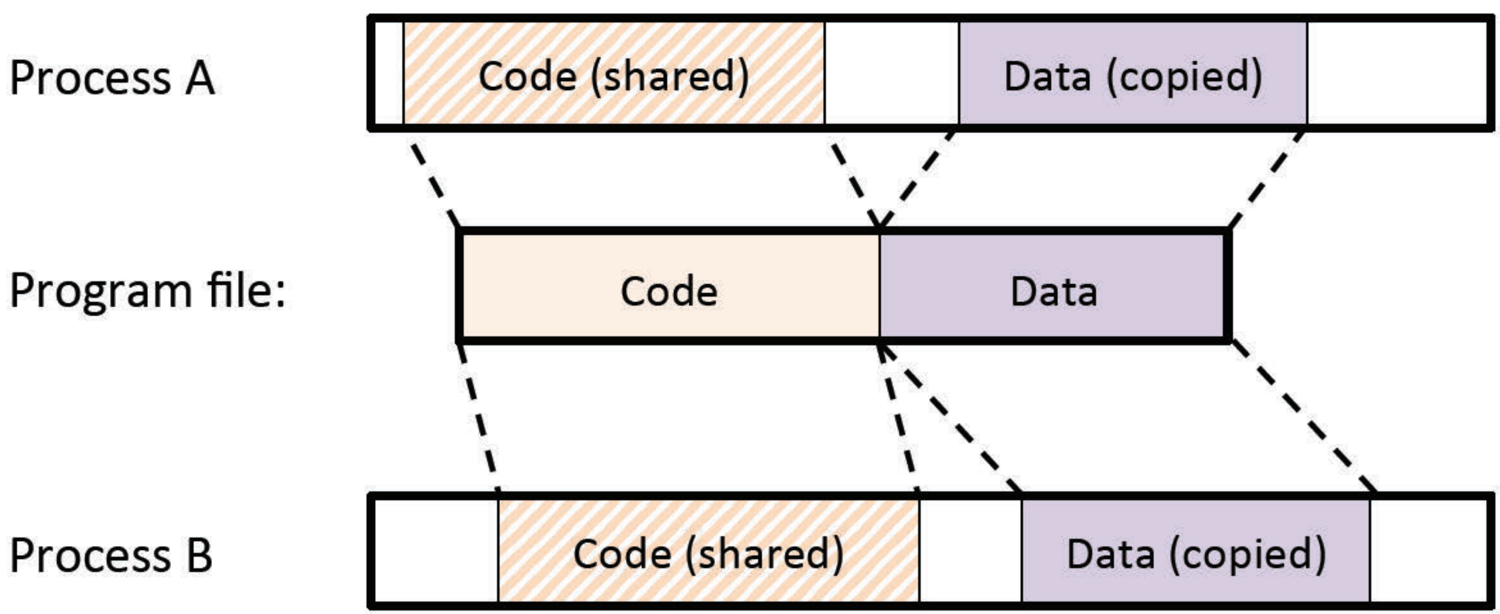


The FLUSH+RELOAD Technique

- Leaks information on victim access to shared memory.
- Spy monitors victim's access to shared code
  - Spy can determine what victim does
  - Spy can infer the data the victim operates on

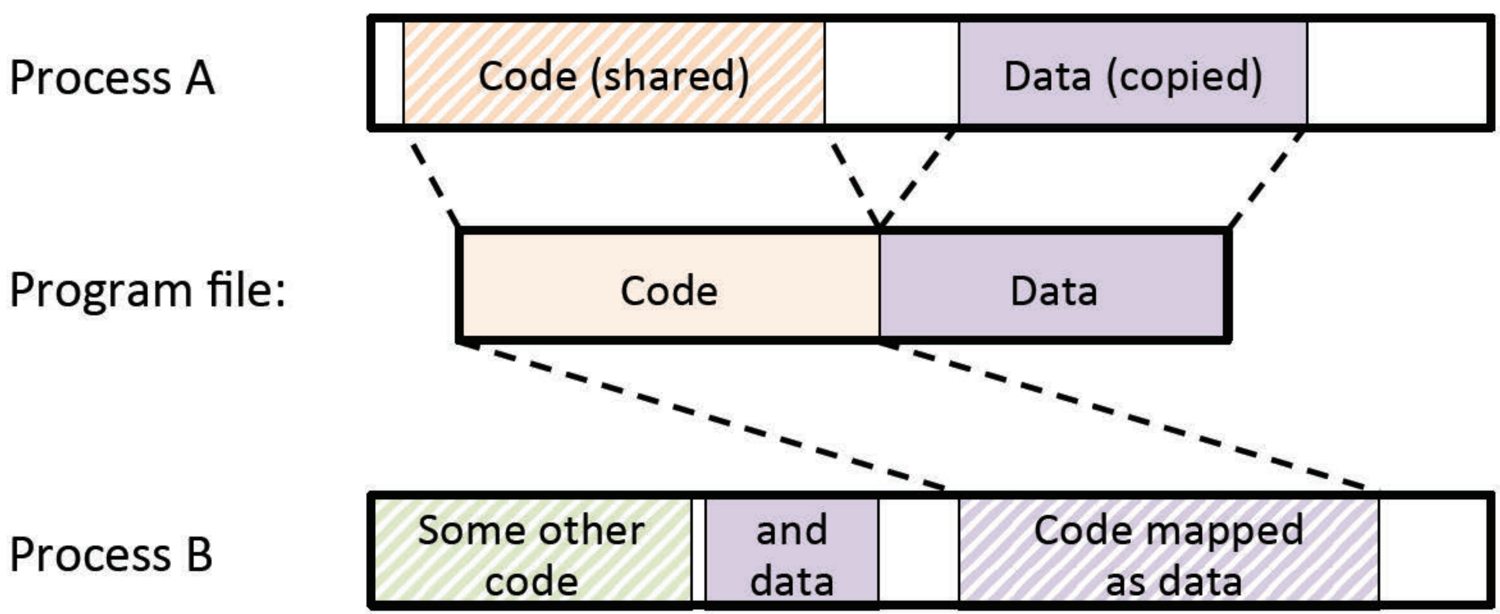
Code Sharing

- To reduce its memory footprint, the operating system shares code between processes



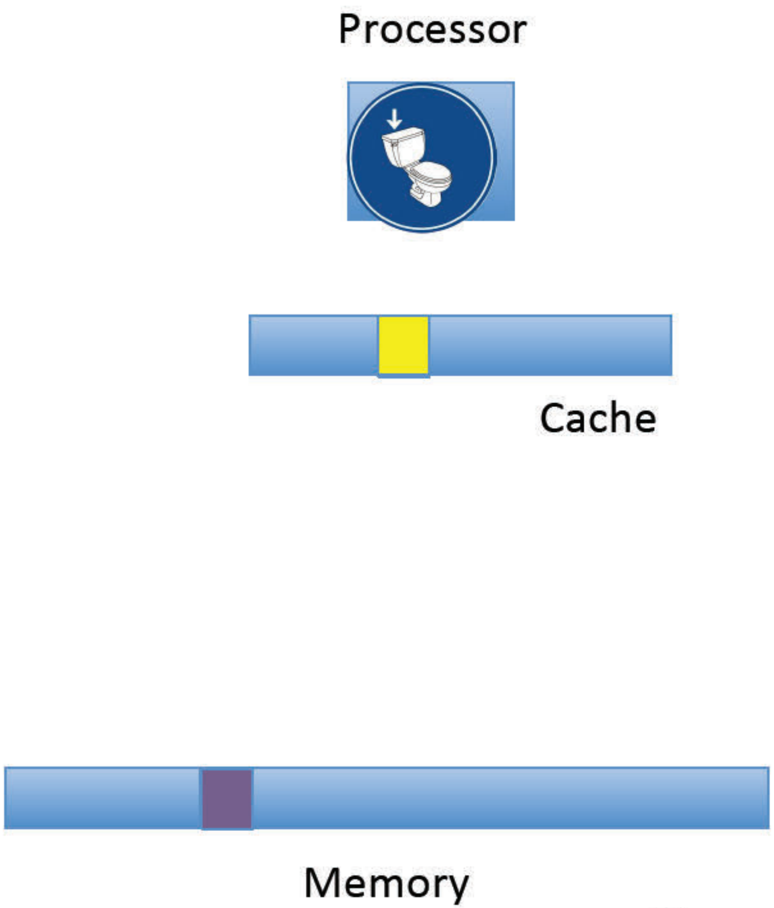
Code is Data

- In Von Neumann architectures code is a type of data



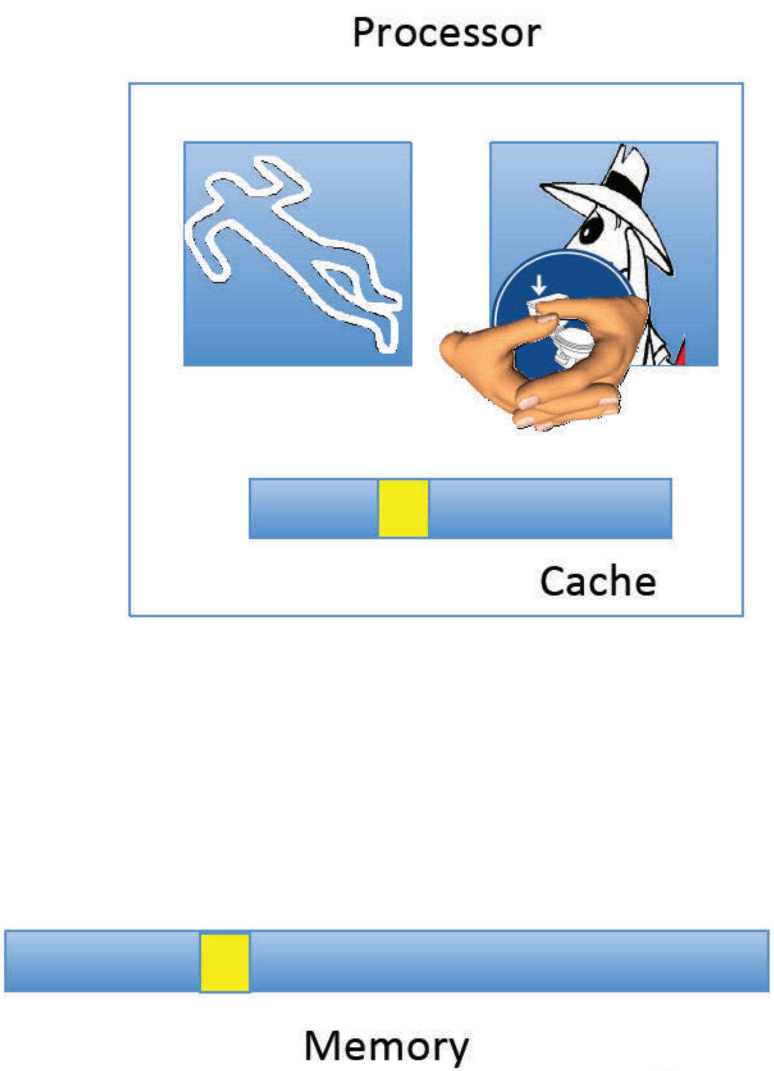
Cache Consistency

- Memory and cache can be in inconsistent states
  - Rare, but possible
- Solution: Flushing the cache contents
  - Ensures that the next load is served from the memory



FLUSH+RELOAD [GBK11,YF14]

- **FLUSH** memory line
- Wait a bit
- Measure time to **RELOAD** line
  - slow-> no access
  - fast-> access
- Repeat



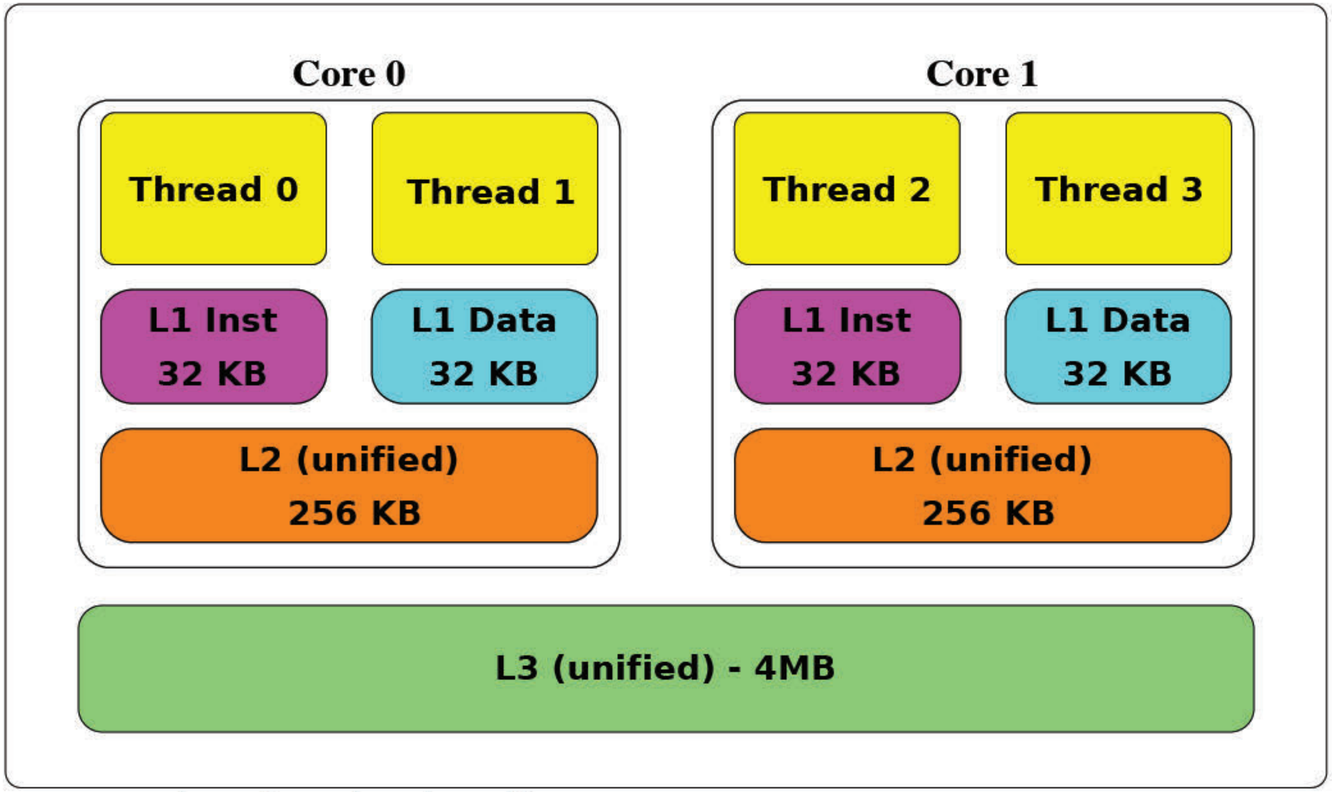
Demo

Attacking GnuPG 1.4.13

Limitations

- Requires shared memory
  - Easy to mitigate in virtualised environment
    - Modern hypervisors do not share across VMs
  - Harder to mitigate within the operating system or in PaaS platform
- Cannot monitor access to data

Prime+Probe on the Last Level Cache



- Some technical challenges
  - See Liu et al. IEEE S&P 2015
  - Or just use Mastik

Countermeasures - Hardware

- Re-design the cache
  - Random replacement
  - Cache partitioning
- Don't hold your breath…

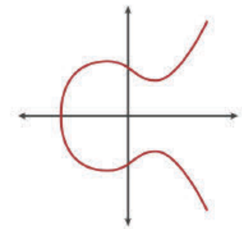
Countermeasures - System

- Detection
  - May be circumvented
- Prevention
  - All suggested methods have subtle limitations

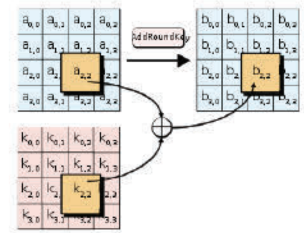
Countermeasures - Software

- Blinding
  - Not always applicable
  - Not always work
- Constant-time programming
  - Fragile

Summary



- Cache attacks are a threat to security
  - Multiple ciphers
  - Multiple system models



- (Almost) easy to mount
  - Mastik
- Hard to mitigate
  - No silver bullet