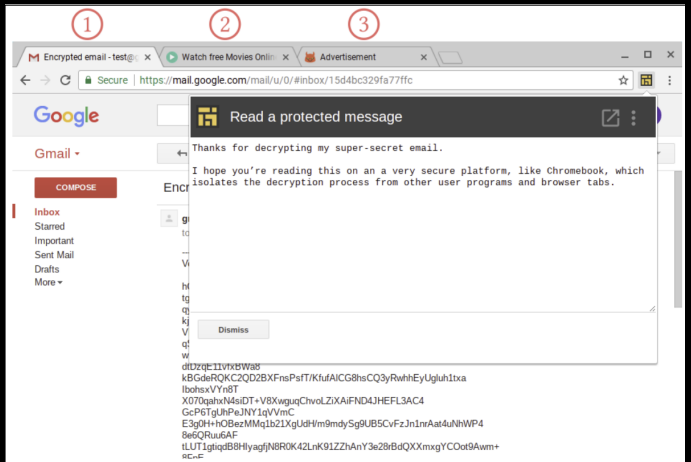
OSY.SSI[2019][7]

Tromer et al. 2018



Screenshot of the attack scenario. The target user opens an online streaming web-site in Tab ②. Pressing somewhere in this tab (for example to start a movie), causes a pop-under to open up as Tab ③. The malicious advertisement in Tab ③ then monitors the cache activity on the target machine. When an encrypted email is received and decrypted using Google's encrypted email extension (in Tab ①), the malicious advertisement in Tab ③ learns information about the user's secret key.

Exercise

How to detect if you are surveilled by a drone?

Side channels are a nice 'read' primitive

Using side-channel analysis we can extract data.

But designers may be aware of this. So what do?

A metaphor

- Alice likes 'toys'. She orders a few on some website, to be delivered.
- Bob sells the toys: $10 for a medium, $31 for a big one. Broken toys are not charged to his clients.
- Charlie is a lowly (and pervy) employee from the postal service.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.
- Charlie can read the bills and cheques, but can't open the sealed package.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.
- Charlie can read the bills and cheques, but can't open the sealed package.
- Charlie kicks the box and breaks things.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.
- Charlie can read the bills and cheques, but can't open the sealed package.
- Charlie kicks the box and breaks things.
- Alice receives the toys, some of which are broken. She pays for the unbroken ones.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.
- Charlie can read the bills and cheques, but can't open the sealed package.
- Charlie kicks the box and breaks things.
- Alice receives the toys, some of which are broken. She pays for the unbroken ones.
- Exercice: given the bill amount x and the paid amount y, deduce what Alice bought.

A metaphor

- Alice orders toys, she'll pay Bob by cheque upon reception.
- Bob prepares the package and sends it through the post, with a bill.
- Charlie can read the bills and cheques, but can't open the sealed package.
- Charlie kicks the box and breaks things.
- Alice receives the toys, some of which are broken. She pays for the unbroken ones.
- Exercice: given the bill amount x and the paid amount y, deduce what Alice bought.

This is an example of a fault attack.

Kicking the box

In practice, a fault injection can be triggered by

- Literally kicking the thing
- Electromagnetic interaction (magnets, pulsed EM fields, lasers, tension glitch)
- Temperature extremes
- Clock skewing / Bus delaying

Until 2014, fault attacks required physical access to the device, then...

Rowhammer

- Physical design of high-density DRAM
- Refresh mechanism due to leak (row buffer == cache)
- Cells leak faster upon proximal access
- "Merely a reliability issue"

Rowhammer

- Physical design of high-density DRAM
- Refresh mechanism due to leak (row buffer == cache)
- Cells leak faster upon proximal access
- "Merely a reliability issue"
- In 2015 a full priviledge escalation attack was mounted from it

"It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after"

Rowhammer

- Essentially a Flush+Reload/Prime+Probe loop, because data isn't read if it's cached.
- Can be run from JavaScript (done here last year)
- Doesn't work on mobile
  - Flush instruction is priviledged
  - Cache eviction too slow
  - Non-temporal stores may still be cached

  Lol jk still feasible through `/dev/ion` on Android

Kim et al. (2014): 110/129 DDR3 modules affected.
Seaborn and Dullien (2016): 15/29 laptops affected.

What do we do with bitflips?

- Change data structures (e.g., permissions, become root)
- Change opcodes (e.g., conditional jump, become root)
  - sudo: 29 possible bit flips to bypass password check

Am I worthy?

- Not an easy attack but feasible (done 2 years in a row here)
- Refresh rate would need to be $\times 7$ to stop the attack
- Blocking flush instruction isn't enough (JS eviction attacks)
- ECC RAM is costly!

OK, back to faults

Tension/Power supply glitches

- Can cause the CPU to skip an instruction (oops no password check)
- Can cause the CPU to misread an instruction (error -> no error)

Clock skew

- Attempt to read data before it is loaded = stale input
- Start executing next instruction before the current one is finished

Temperature extremes

- Read and write thermal limits are different: e.g. can read, cannot write
- RAM leakage increases with temperature: selective erasure

More faults

Light and EM induction

- Electrical induction allows to change bits from 1 to 0 and vice versa
- Laser is very precise, can simulate many faults (e.g. particle accelerators)
- X-ray and ion beams can be used even with packaged circuits

Some faults are transient, destructive, or eventually destructive.

Example: Single Event Latch-up faults (SELs) are propagated in an electronic circuit by the creation of a self-sustained current with the releasing of PNPN parasitic bipolar transistors.

In practice we prefer transient faults, from which the target recovers.

An example target

AES is the standard for data encryption.

- KeyExpansion: derive 128-bit 'round keys' for each round
- Initialisation:
  - AddRoundKey: state = state XOR round key.
- Repeat (9, 11 or 13 rounds):
  - SubBytes: inversion in a finite field
  - ShiftRows: transposition step
  - MixColumns: linear mixing operation
  - AddRoundKey: state = state XOR round
- Final round (making 10, 12 or 14 rounds in total):
  - SubBytes
  - ShiftRows
  - AddRoundKey

Questions: how to attack this?

An example target

AES is the standard for data encryption.

- KeyExpansion: derive 128-bit 'round keys' for each round
- Initialisation:
  - AddRoundKey: state = state XOR round key.
- Repeat (9, 11 or 13 rounds):
  - SubBytes: inversion in a finite field
  - ShiftRows: transposition step
  - MixColumns: linear mixing operation
  - AddRoundKey: state = state XOR round
- Final round (making 10, 12 or 14 rounds in total):
  - SubBytes
  - ShiftRows
  - AddRoundKey

Questions: how to attack this? One approach: Differential Fault Analysis (Biham–Shamir 1998)

Timeline

- First documented use of logical faults to break cryptography: RSA, Bellcore team (DeMillo–Boneh–Lipton), 1997.
- Biham–Shamir: any block cipher, 1998.
- Piret et al.: retrieve an AES key with only one pair of correct/faulty ciphertexts with a fault on a single byte of the state before the penultimate MixColumn, 2003.
- Sasaki et al.: AES, disturb 1 byte, key can be revealed even with a large number of noisy fault injections, 2013.

A note on AES-GCM

From a 'traditional' cryptographic standpoint, AES-GCM is unbroken, and used all over the place.

- "Do not use GCM" (Niels Ferguson)
- "common implementations of GCM are potentially vulnerable to [...] cache timing attacks." (Emilia Käsper, Peter Schwabe, 2009)
- "AES-GCM so easily leads to timing side-channels that I'd like to put it into Room 101." (Adam Langley, 2013)
- "The fragility of AES-GCM authentication algorithm" (Shay Gueron, Vlad Krasnov, 2013)
- "GCM is extremely fragile" (Kenny Paterson, 2015)

"Practical Forgery Attacks on GCM in TLS", Böck et al., 2016.

Wait a second what is going on?

Wait a second what is going on?

- Cryptographers: it's the implementer's fault!

Wait a second what is going on?

- Cryptographers: it's the implementer's fault!
- Software engineer: it's the hardware engineers' fault!

Wait a second what is going on?

- Cryptographers: it's the implementer's fault!
- Software engineer: it's the hardware engineers' fault!
- Hardware engineer: it's your adversarial model's fault!

Wait a second what is going on?

- Cryptographers: it's the implementer's fault!
- Software engineer: it's the hardware engineers' fault!
- Hardware engineer: it's your adversarial model's fault!
- Adversary: someone said fault?

Fault attacks allow an attacker to read/write data even in programs that are perfectly correct and run on perfectly fine hardware.

Demo

Wait a second, what is going on?

- The fault injection gave the 8F and TR items
- But what about the rest?
- And how do we find these things?

Well...

Come next time and you'll see :3