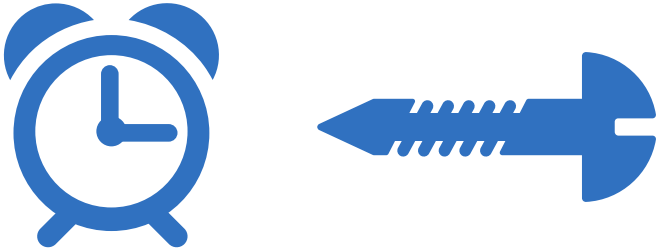
Flipping Bits from Software without Rowhammer

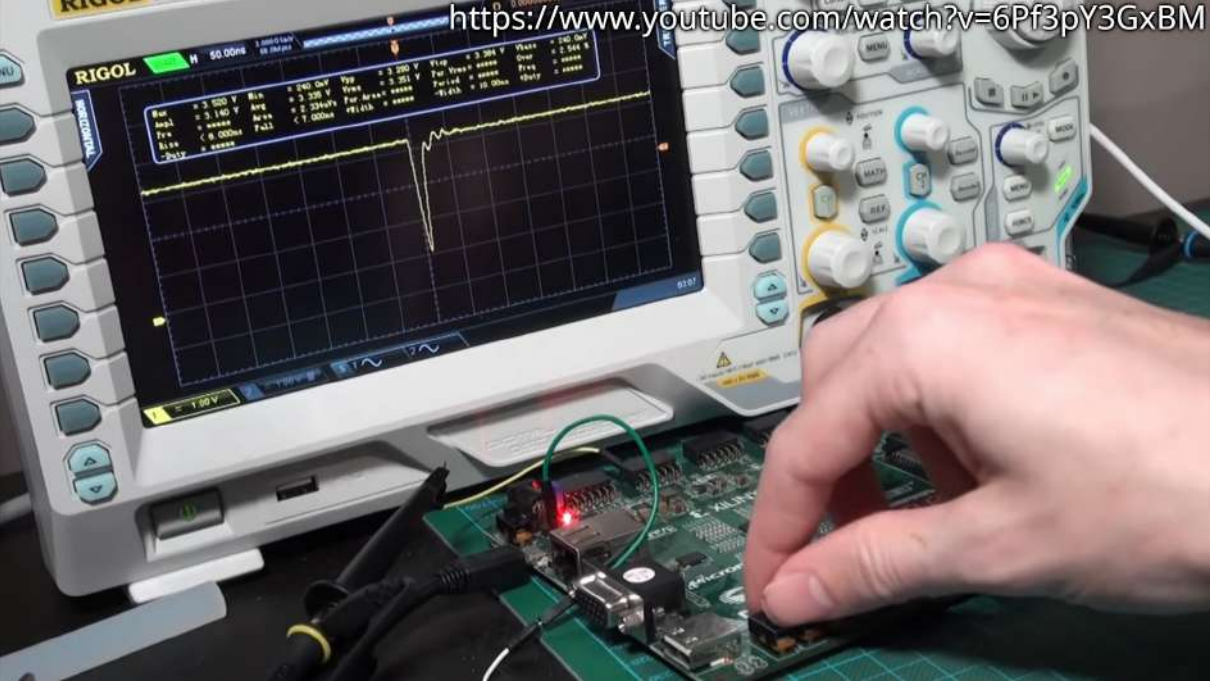
Kit Murdock, Daniel Gruss

December 27, 2019

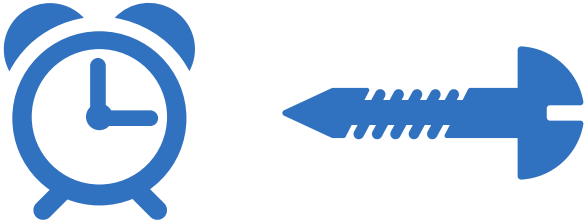
In the beginning...



A NEW CLASS OF FAULT ATTACKS


https://www.youtube.com/watch?v=6Pf3pY3GxBM
RIGOL

Our starting point...



Adrian Tang et al. “CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management”. In: *USENIX Security Symposium*. 2017

DVFS

Dynamic Voltage and Frequency Scaling

Changing the voltage and frequency of the CPU

Dynamic Voltage and Frequency Scaling

Changing the voltage and frequency of the CPU

- Gamers want fast responses

Dynamic Voltage and Frequency Scaling

Changing the voltage and frequency of the CPU

- Gamers want fast responses
- Cloud servers want high-assurance and low running costs

Dynamic Voltage and Frequency Scaling

Changing the voltage and frequency of the CPU

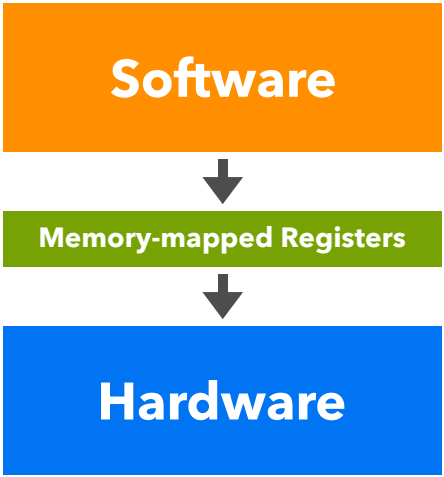
- Gamers want fast responses
- Cloud servers want high-assurance and low running costs
- What if the hardware gets hot?

Dynamic Voltage and Frequency Scaling

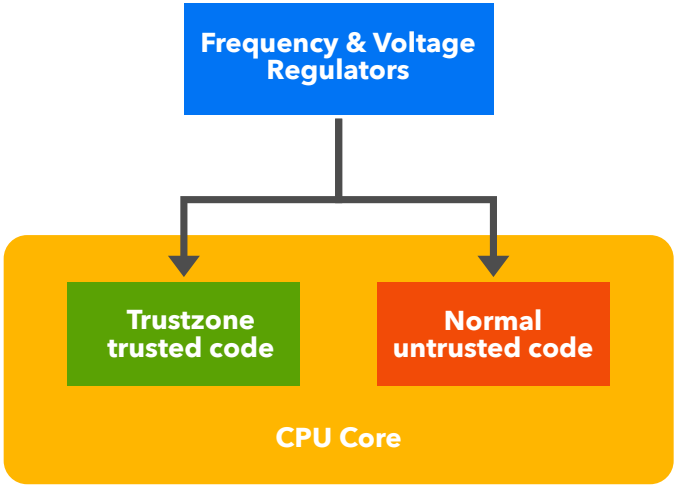
Changing the voltage and frequency of the CPU

- Gamers want fast responses
- Cloud servers want high-assurance and low running costs
- What if the hardware gets hot?
- Optimal voltage & frequency is difficult!

Memory Mapped Registers...

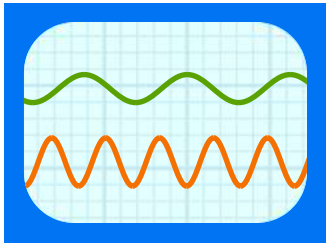


Memory Mapped Registers



CLKscrew attack

```
add.w   (a0)+,d1
cmp.l   a0,d0
bcc.s   loop
movea.l #$18E,a1
cmp.w   (a1),d1
bne.w   WrongChecksum
```



2 + 2 = 5

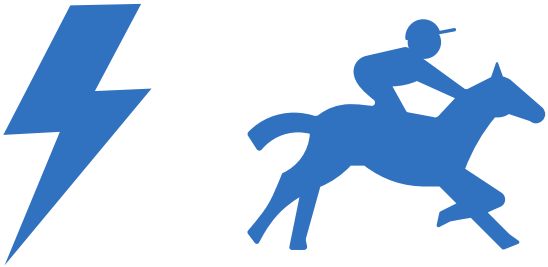
CLKscrew - Attacks

- Infer secret AES key that was stored within Trustzone

- Infer secret AES key that was stored within Trustzone
- Trick Trustzone into loading a self-signed app

ARM Processors

More are joining the revolution...

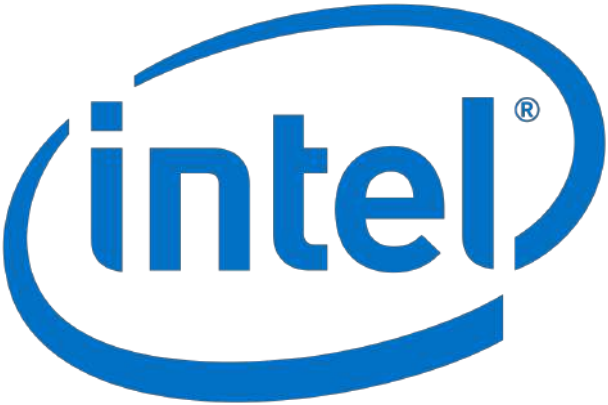


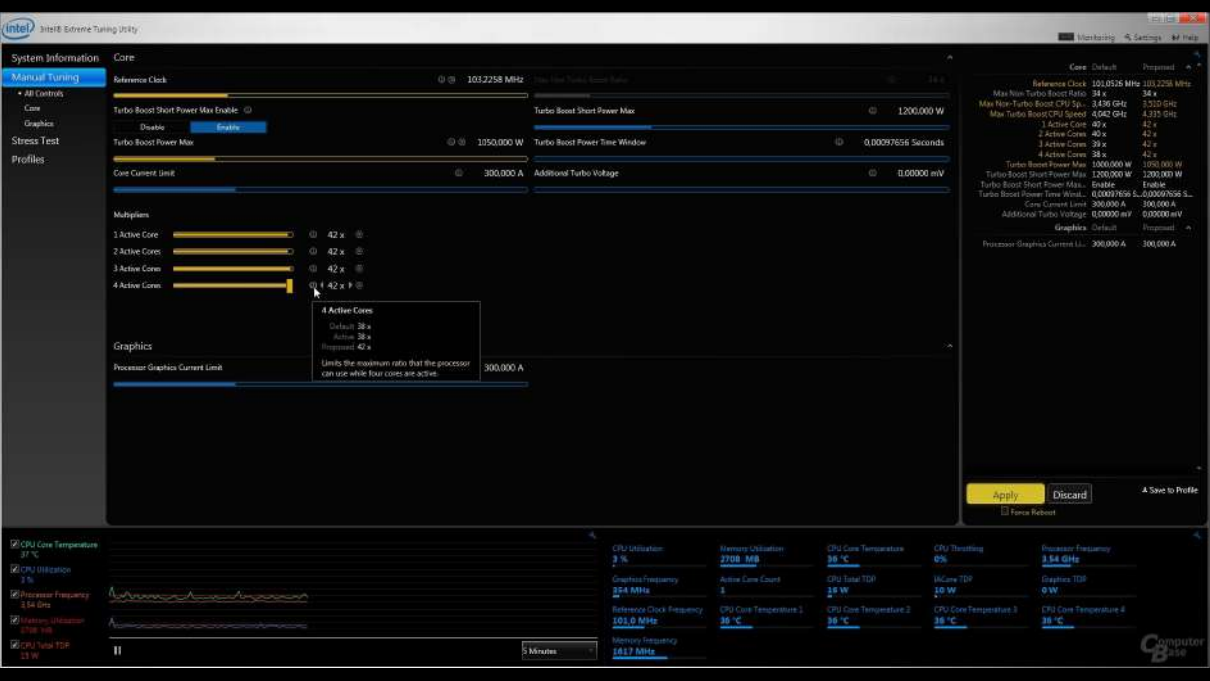
Pengfei Qiu et al. “VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies”. In: *CCS*. 2019

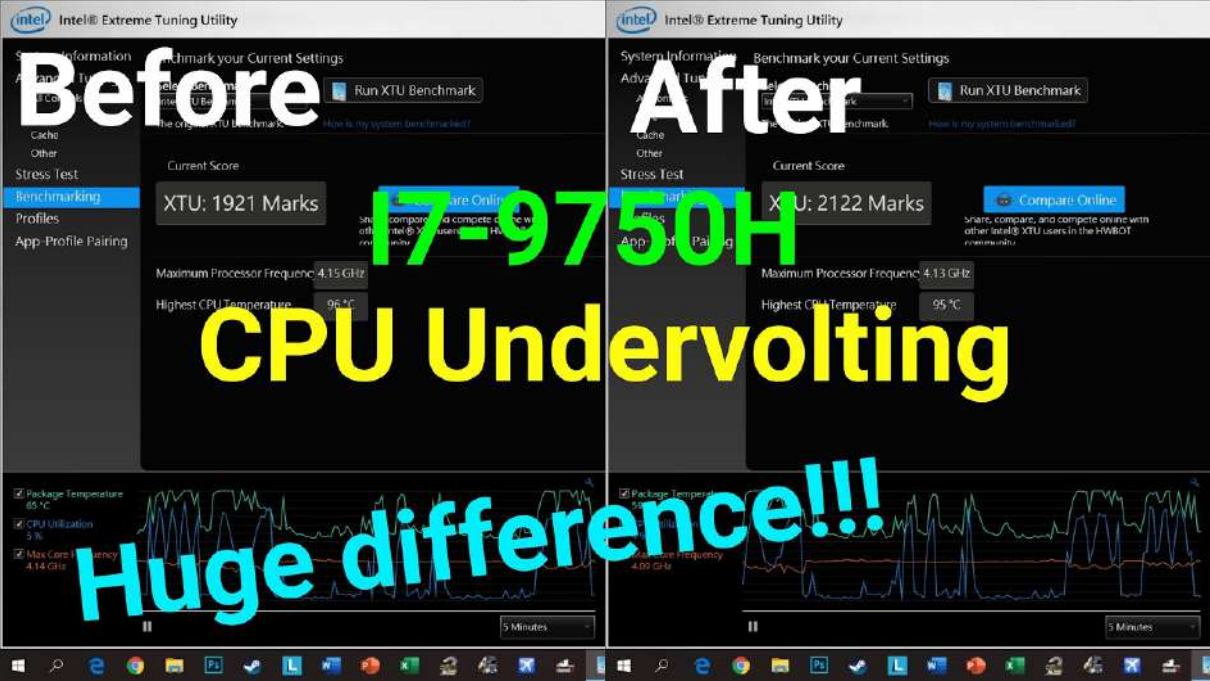
ARM Processors

ARE WE ALL THINKING THE SAME THING?

What about Intel?




Intel® Extreme Tuning Utility
Monitoring
Settings
Help
System Information
Manual Tuning
• All Controls
Core
Graphics
Stress Test
Profiles
Core
Reference Clock
103.2258 MHz
Turbo Boost Short Power Max Enable
Disable
Enable
Turbo Boost Short Power Max
1200,000 W
Turbo Boost Power Max
1050,000 W
Turbo Boost Power Time Window
0,00097656 Seconds
Core Current Limit
300,000 A
Additional Turbo Voltage
0,00000 mV
Multipliers
1 Active Core
42 x
2 Active Cores
42 x
3 Active Cores
42 x
4 Active Cores
42 x
4 Active Cores
Default 38 x
Active 38 x
Proposed 42 x
Limits the maximum ratio that the processor can use while four cores are active.
Graphics
Processor Graphics Current Limit
300,000 A
Core Default Proposed
Reference Clock 101,0526 MHz 103,2258 MHz
Max Non-Turbo Boost Ratio 34 x 34 x
Max Non-Turbo Boost CPU Sp... 3,436 GHz 3,510 GHz
Max Turbo Boost CPU Speed 4,042 GHz 4,235 GHz
1 Active Core 40 x 42 x
2 Active Cores 40 x 42 x
3 Active Cores 39 x 42 x
4 Active Cores 38 x 42 x
Turbo Boost Power Max 1000,000 W 1050,000 W
Turbo Boost Short Power Max 1200,000 W 1200,000 W
Turbo Boost Short Power Max... Enable Enable
Turbo Boost Power Time Wind... 0,00097656 S... 0,00097656 S...
Core Current Limit 300,000 A 300,000 A
Additional Turbo Voltage 0,00000 mV 0,00000 mV
Graphics Default Proposed
Processor Graphics Current Li... 300,000 A 300,000 A
Apply
Discard
Force Reboot
Save to Profile
CPU Core Temperature
37 °C
CPU Utilization
3 %
Processor Frequency
3.54 GHz
Memory Utilization
2708 MB
CPU Total TDP
10 W
5 Minutes
CPU Utilization
3 %
Memory Utilization
2708 MB
CPU Core Temperature
36 °C
CPU Throttling
0%
Processor Frequency
3.54 GHz
Graphics Frequency
354 MHz
Active Core Count
1
CPU Total TDP
10 W
IACore TDP
10 W
Graphics TDP
0 W
Reference Clock Frequency
101.0 MHz
CPU Core Temperature 1
36 °C
CPU Core Temperature 2
36 °C
CPU Core Temperature 3
36 °C
CPU Core Temperature 4
36 °C
Memory Frequency
1417 MHz


Intel® Extreme Tuning Utility
Before
Run XTU Benchmark
Stress Test
Benchmarking
Profiles
App-Profile Pairing
Current Score
XTU: 1921 Marks
Maximum Processor Frequency 4.15 GHz
Highest CPU Temperature 96 °C
Intel® Extreme Tuning Utility
After
Run XTU Benchmark
Stress Test
Current Score
XTU: 2122 Marks
Compare Online
Maximum Processor Frequency 4.13 GHz
Highest CPU Temperature 95 °C
I7-9750H
CPU Undervolting
Huge difference!!!
5 Minutes
5 Minutes

Intel Power Management

- RMClock

Intel Power Management

- RMClock
- Throttlestop

Intel Power Management

- RMClock
- Throttlestop
- linux-intel-undervolt

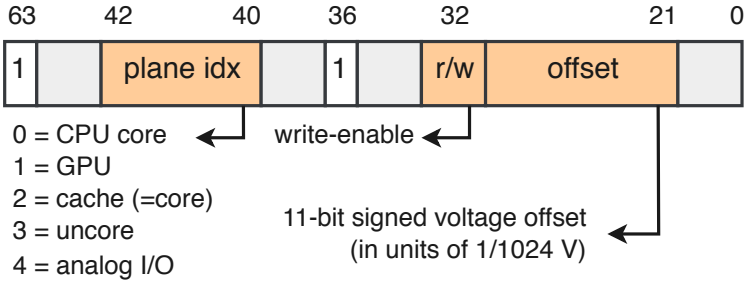
Intel Power Management

- RMClock
- Throttlestop
- linux-intel-undervolt
- Adrian Tang's PhD dissertation

Intel Power Management

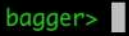
Intel Power Management

`msr 0x150`

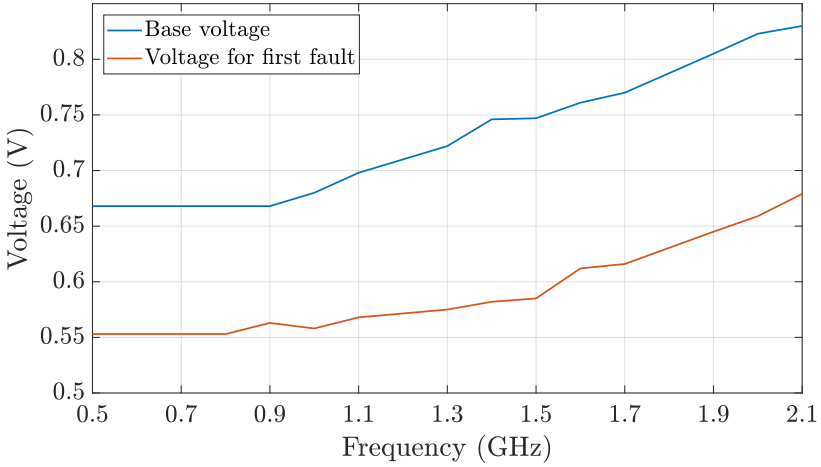

63
42
40
36
32
21
0
1
plane idx
1
r/w
offset
0 = CPU core
1 = GPU
2 = cache (=core)
3 = uncore
4 = analog I/O
write-enable
11-bit signed voltage offset
(in units of 1/1024 V)

Will it fault?

```
uint64_t multiplier = 0x1122334455667788;
uint64_t correct    = 0xdeadbeef * multiplier;
uint64_t var        = 0xdeadbeef * multiplier;

while (var == correct)
{
    var = 0xdeadbeef * multiplier;
}
uint64_t flipped_bits = var ^ correct;
```


bagger>

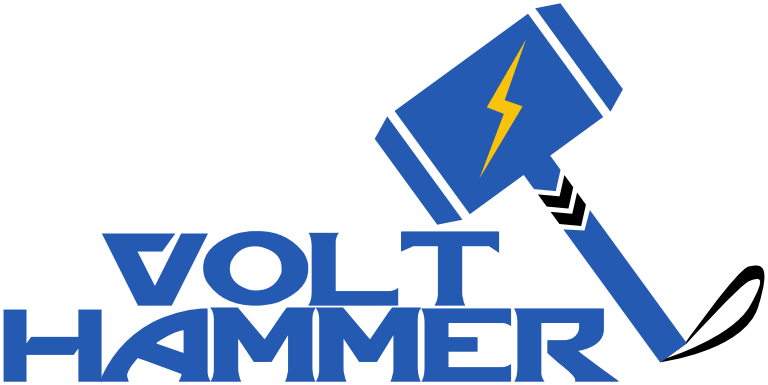
Safe voltages



Aristotle Tzafalias
@Aristot73

thesis: given a cool name, researchers will be able to find a new vulnerability to match it, in a finite period of time; cool name-logo combination requirements have no measurable effect on aforementioned period.

We need a logo!




CTV

ROWHAMMER.JS

ROOT privileges for web apps!

Test - Mozilla Firefox (on lab02)

Test

file:///home/dgruss/rowhammerjs/rowhammer.html

Search

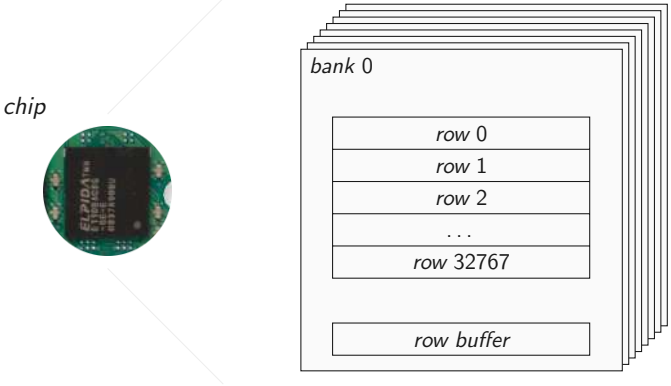
320: 12
330: 9
340: 1
350: 0
360: 1
370: 2
380: 199
390: 76
400: 72
410: 231
420: 572
1250
[!] Found flip (254 != 255) at array index 340021386 when hammering indices 339881984 and 340156416

[!] Found flip (239 != 255) at array index 340022176 when hammering indices 339881984 and 340156416

[!] Found flip (191 != 255) at array index 340023138 when hammering indices 339881984 and 340156416

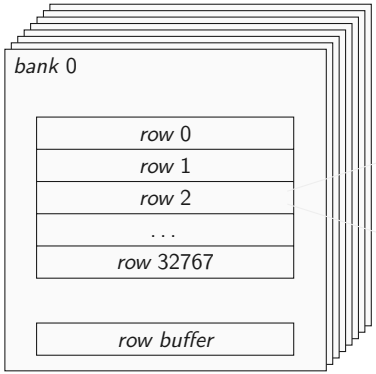
[!] Found flip (254 != 255) at array index 340025146 when hammering indices 339881984 and 340156416

DRAM organization



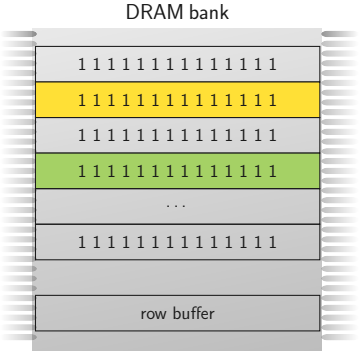
DRAM organization

chip



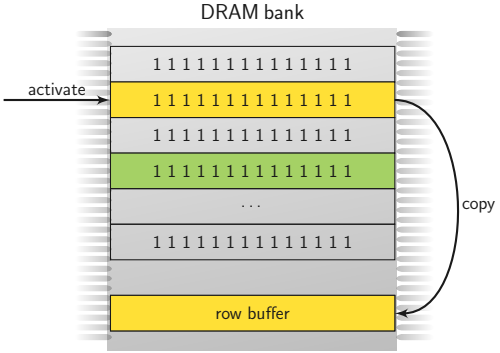
64k cells
1 capacitor,
1 transitor each

Rowhammer



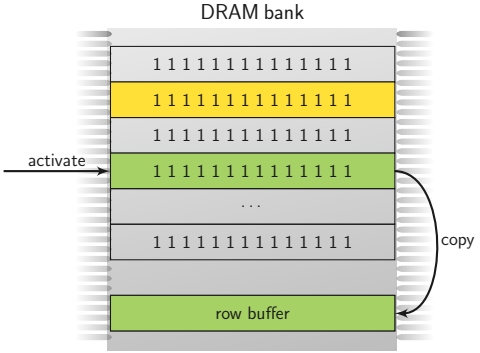
- Cells leak $\rightarrow$ need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Rowhammer



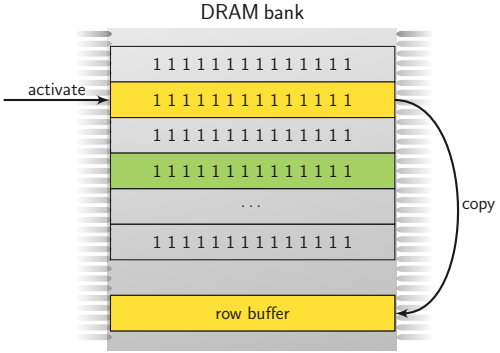
- Cells leak $\rightarrow$ need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Rowhammer



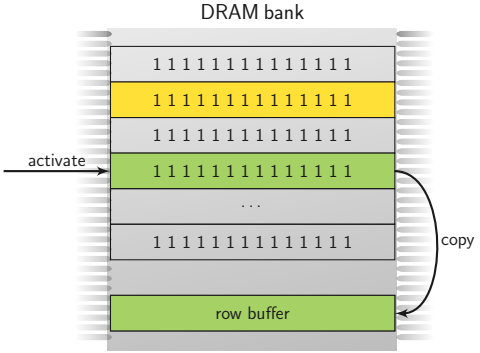
- Cells leak → need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses → Rowhammer

Rowhammer



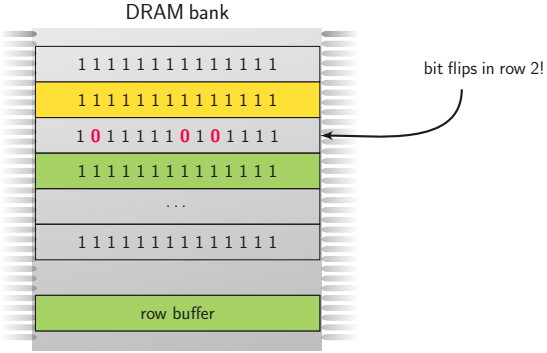
- Cells leak $\rightarrow$ need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Rowhammer



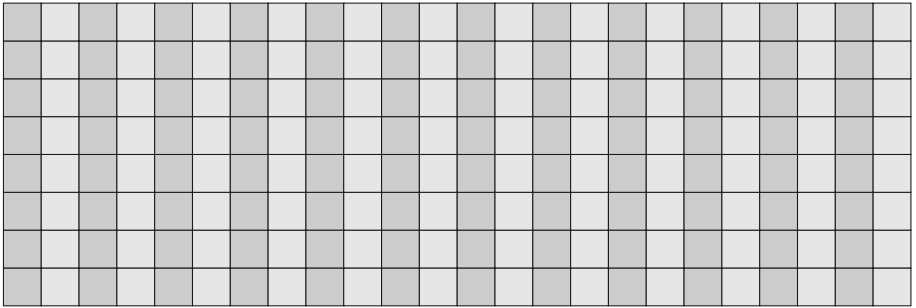
- Cells leak → need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses → Rowhammer

Rowhammer



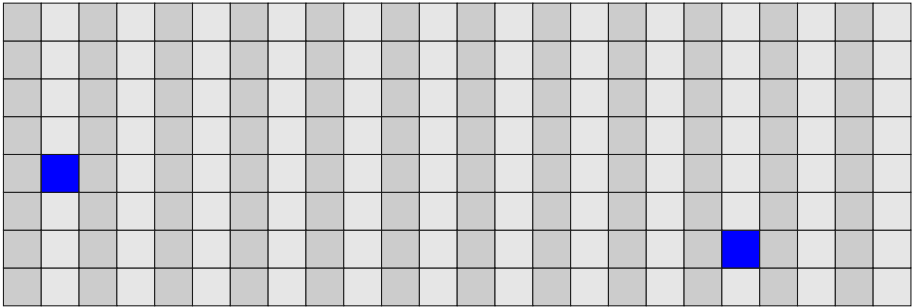
- Cells leak $\rightarrow$ need refresh
- Max. refresh interval to guarantee data integrity
- Cells leak faster upon proximate accesses $\rightarrow$ Rowhammer

Search for page with flip



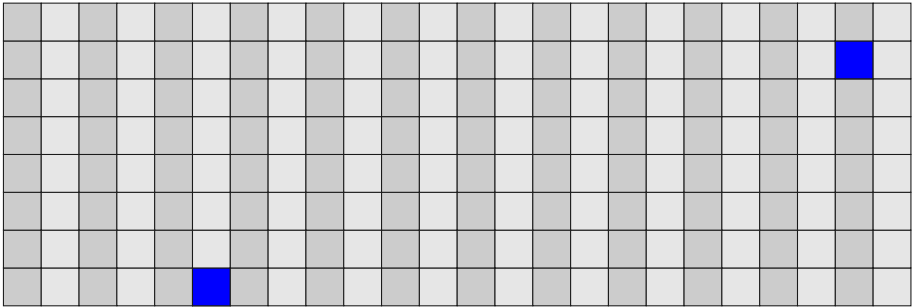
Hammering memory locations in different rows

Search for page with flip



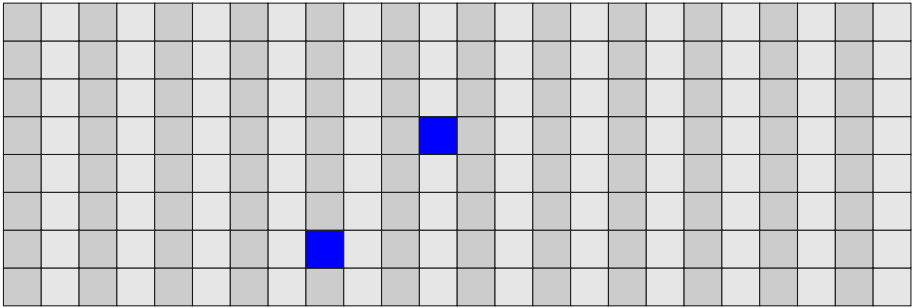
Hammering memory locations in different rows

Search for page with flip



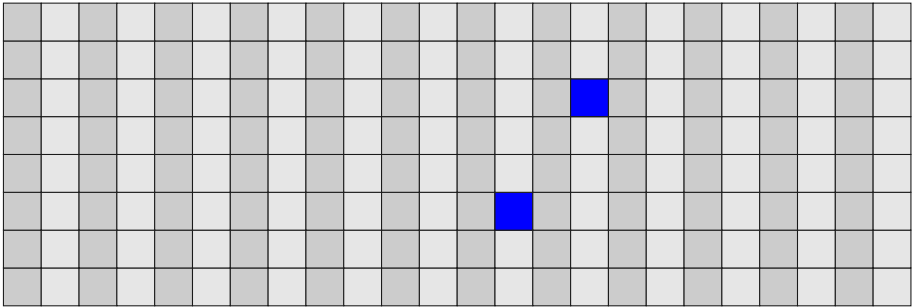
Hammering memory locations in different rows

Search for page with flip



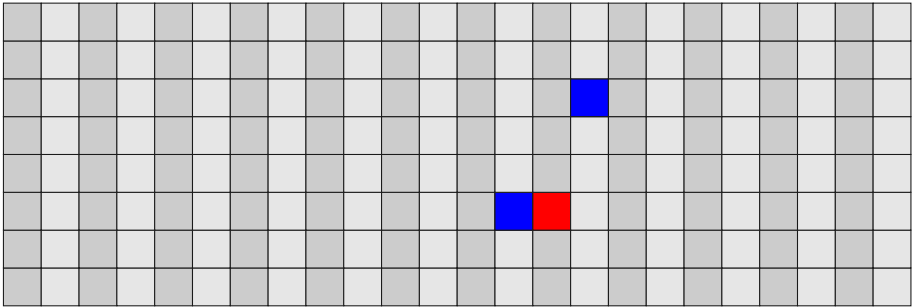
Hammering memory locations in different rows

Search for page with flip



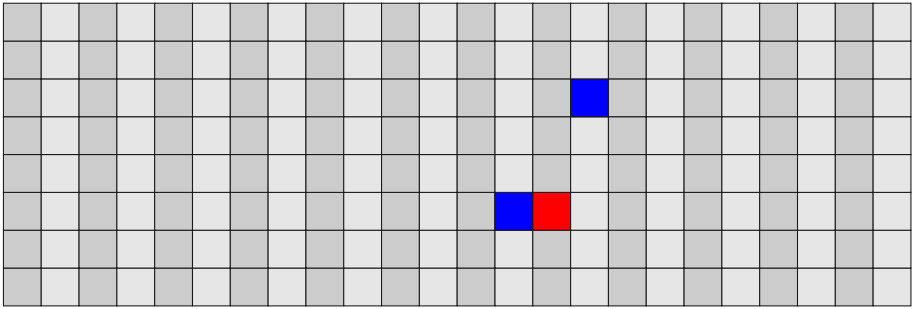
Hammering memory locations in different rows

Search for page with flip

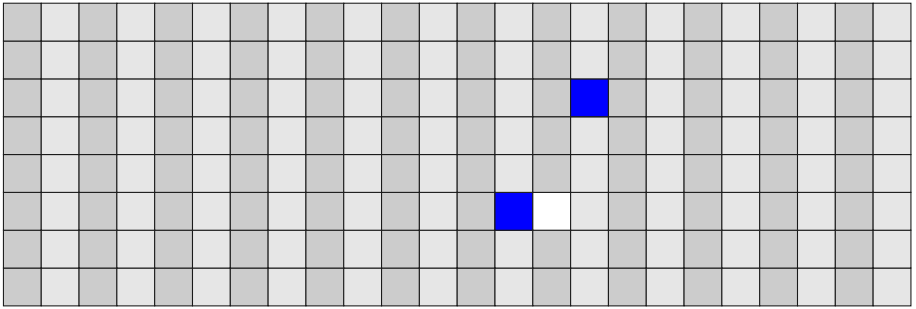


Hammering memory locations in different rows

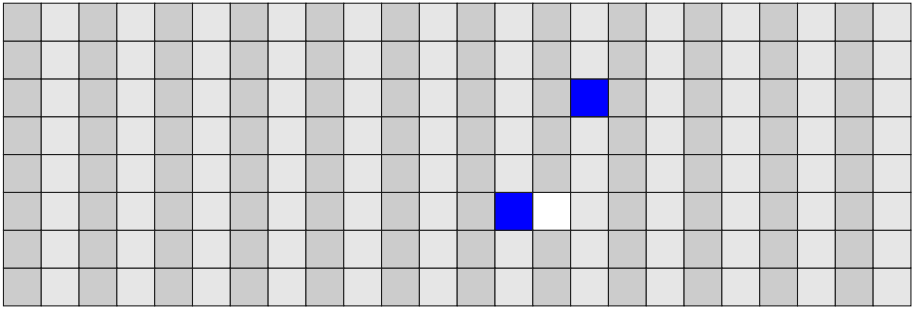
Release page with flip



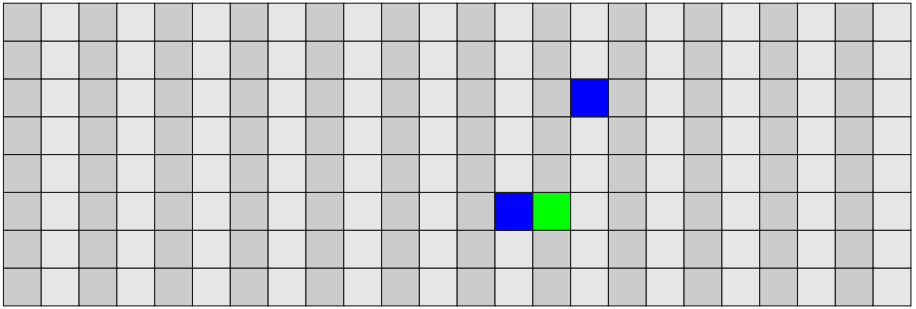
Release page with flip



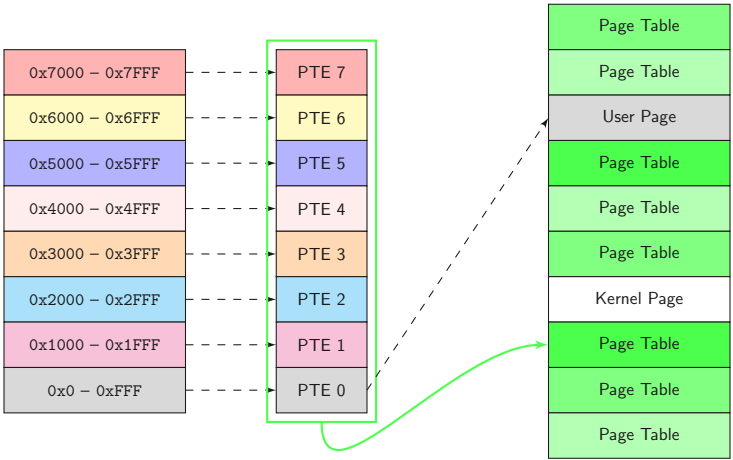
Fill all remaining memory with target pages



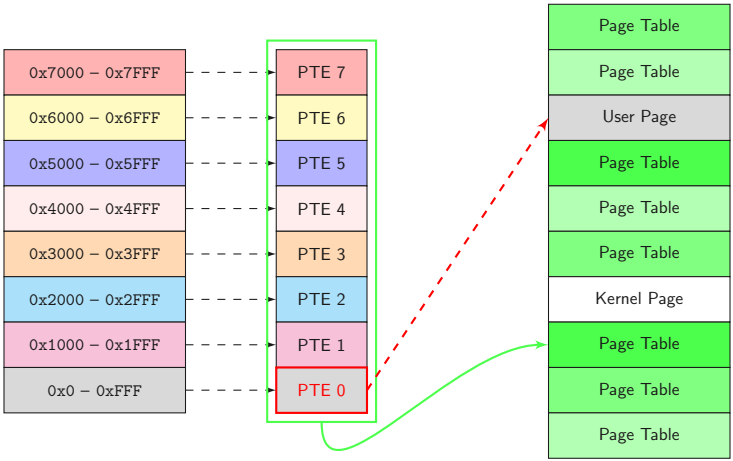
Fill all remaining memory with target pages



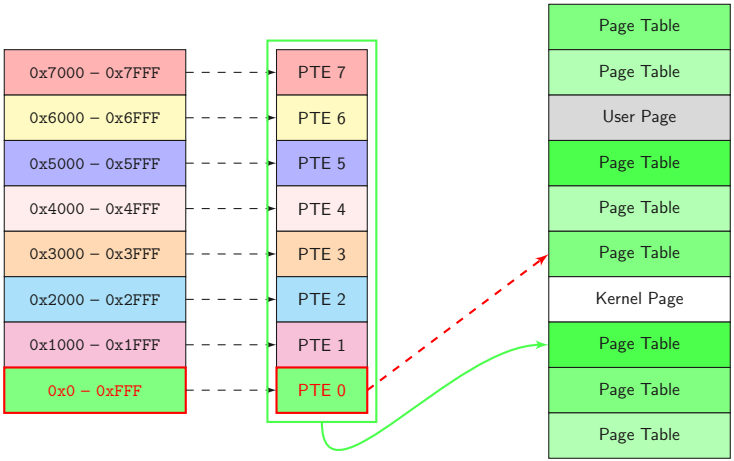
Page Table Example



Page Table Example



Page Table Example



Code Page Example

Code Page Example

Code Page Example

Code Page Example

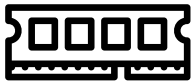
Code Page Example

Code Page Example

Code Page Example

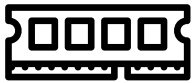
Code Page Example

Rowhammer on x86



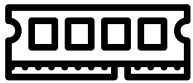
- DDR3 affected

Rowhammer on x86



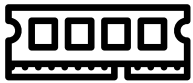
- DDR3 affected
- DDR4 affected

Rowhammer on x86



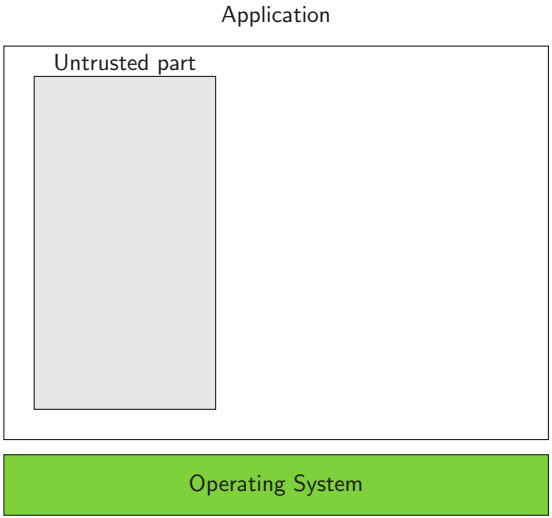
- DDR3 affected
- DDR4 affected
- Even ECC affected despite error correction!

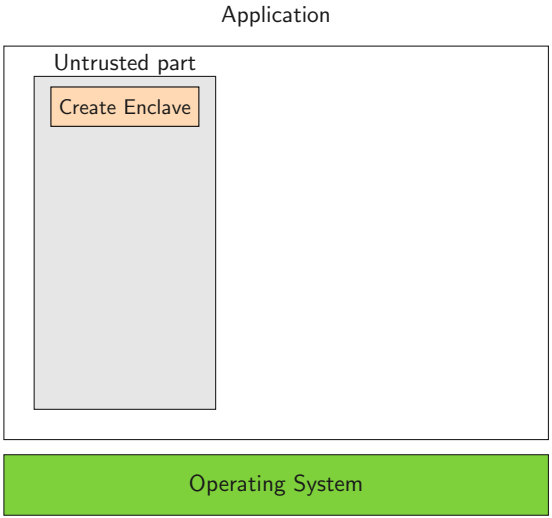
Rowhammer on x86

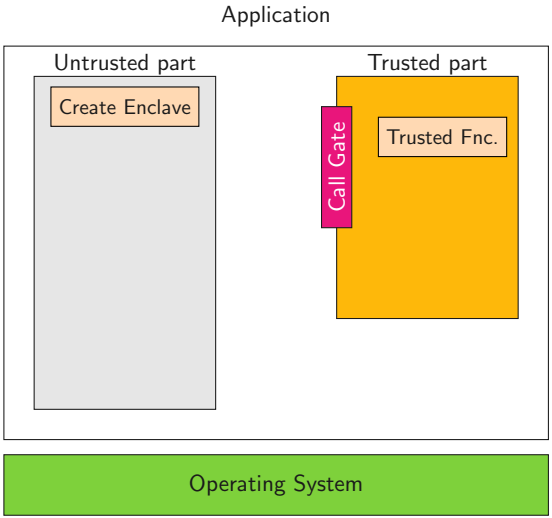


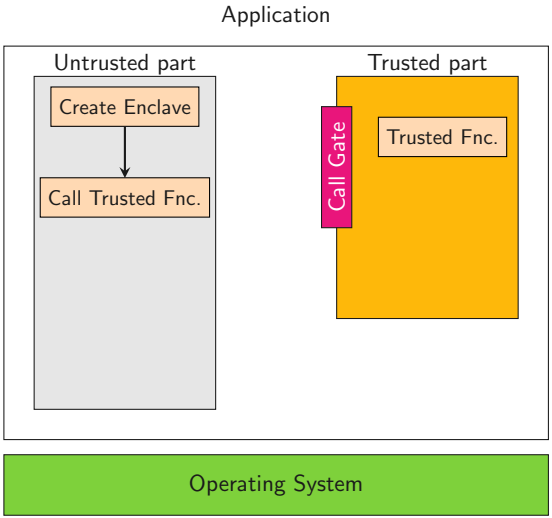
- DDR3 affected
- DDR4 affected
- Even ECC affected despite error correction!

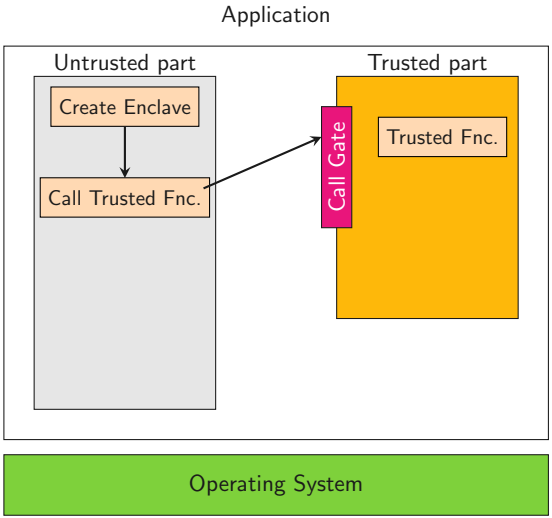
$\rightarrow$ Can SGX's integrity protection prevent Rowhammer?

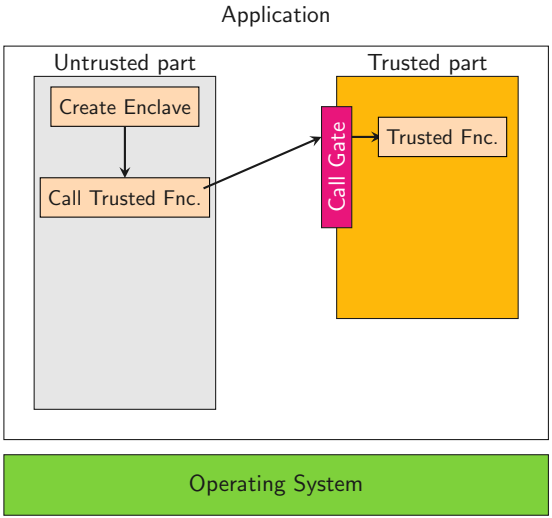

Application
Untrusted part
Operating System

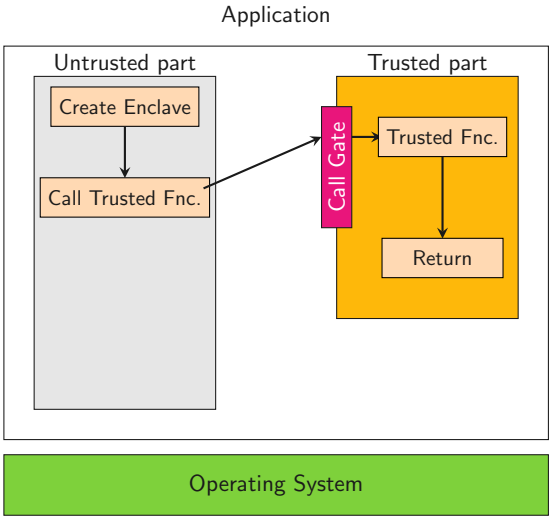

Application
Untrusted part
Create Enclave
Operating System

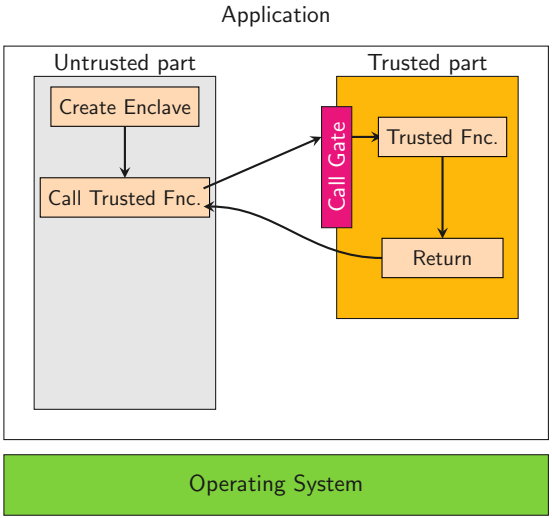

Application
Untrusted part
Create Enclave
Trusted part
Call Gate
Trusted Fnc.
Operating System

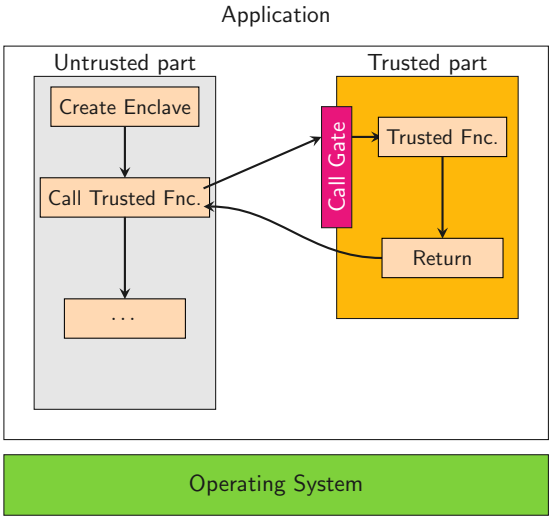

Application
Untrusted part
Create Enclave
Call Trusted Fnc.
Trusted part
Call Gate
Trusted Fnc.
Operating System

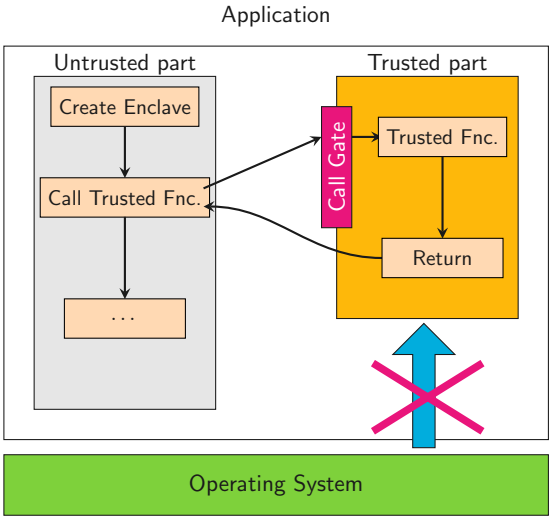

Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
Call Gate
Trusted Fnc.
Operating System


Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
Call Gate
Trusted Fnc.
Operating System


Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
Call Gate
Trusted Fnc.
Return
Operating System

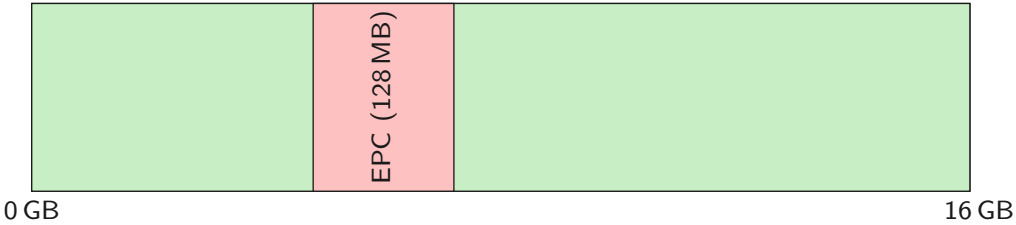

Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
Call Gate
Trusted Fnc.
Return
Operating System


Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
...
Call Gate
Trusted Fnc.
Return
Operating System

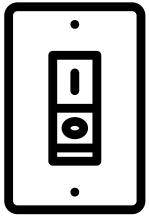

Application
Untrusted part
Trusted part
Create Enclave
Call Trusted Fnc.
...
Call Gate
Trusted Fnc.
Return
Operating System

SGX Encrypted Memory

SGX Encrypted Memory

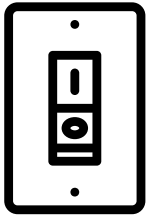


Bit Flips in the EPC



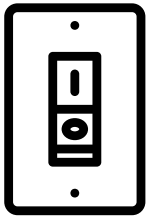
- Bit flips in the EPC?

Bit Flips in the EPC



- Bit flips in the EPC?
- Integrity check fails!

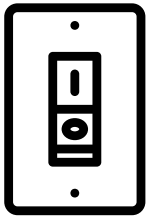
Bit Flips in the EPC



- Bit flips in the EPC?
- Integrity check fails!

→ Lock up memory controller

Bit Flips in the EPC



- Bit flips in the EPC?
- Integrity check fails!

→ Lock up memory controller

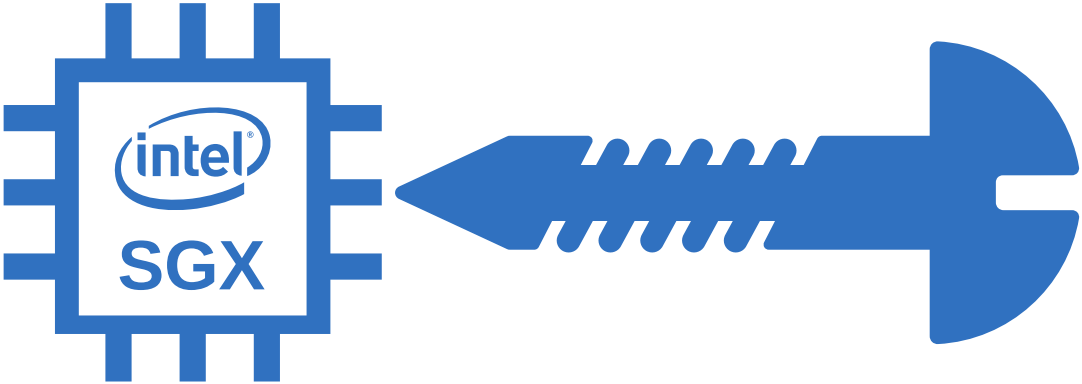
→ System halts immediately (no exploit, but DoS!)

Will SGX save us?

[Userspace] Voltage: -261 mV

```
[Enclave] Multiplying 0xdeadbeef 0x1122334455667788
[Enclave] Multiply_it. result: 0.
```

SGXScrew?



RSA Basics

- Public Key Crypto
- Encrypt/Sign messages
- Untrusted channel
- Encrypt/Verify messages with public key
- Decrypt/Sign messages with private key

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = H^{d} \bmod n$$

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = H^{d} \bmod n$$

$$d_p = d \bmod p - 1$$

$$d_q = d \bmod q - 1$$

$$s_1 = H^{d_p} \bmod p$$

$$s_2 = H^{d_q} \bmod q$$

$$n = \mathrm{p} \times q$$

$$S = ((q^{-1} \bmod p)(s_1 - s_2) \bmod p) \times q + s_2$$

$$d_p = d \bmod p - 1$$

$$d_q = d \bmod q - 1$$

$$s_1 = H^{d_p} \bmod p$$

$$s_2 = H^{d_q} \bmod q$$

$$n = \mathrm{p} \times q$$

$$S = ((q^{-1} \bmod p)(s_1 - s_2) \bmod p) \times q + s_2$$

$$d_p = d \bmod p - 1$$

$$d_q = d \bmod q - 1$$

$$s_1 = H^{d_p} \bmod p$$

$$s_2 = H^{d_q} \bmod q$$

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = ((q^{-1} \bmod p)(s_1 - s_2) \bmod p) \times q + s_2$$

$$S' = ((q^{-1} \bmod p)(s_1 - s_2) \bmod p) \times q + s_2$$

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = (\text{something}) \times q + s_2$$

$$S' = (\text{something else}) \times q + s_2$$

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = (\boxed{\text{something}}) \times q + s_2$$

$$S' = (\boxed{\text{something else}}) \times q + s_2$$

$$S - S' = \text{something} \times q$$

RSA Signature with Chinese Remainder Theorem

$$n = \mathrm{p} \times q$$

$$S = (\boxed{\text{something}}) \times q + s_2$$

$$S' = (\boxed{\text{something else}}) \times q + s_2$$

$$S - S' = \text{something} \times q$$

$$q = \gcd(S - S', n)$$

RSA decryption

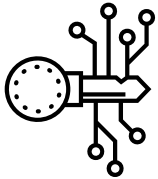
```
uint8_t rsa_dec_ecall(int iterations)
{
    // Wait for first fault
    trigger_fault(iterations);

    // Actual decryption
    ippsRSA_Decrypt(ct, dec, pPrv, scratchBuffer);
}
```

```
bagger> dog Enclave/encl
```


MUFARSA
My Undervolting
Fault Attack on RSA
imgflip.com

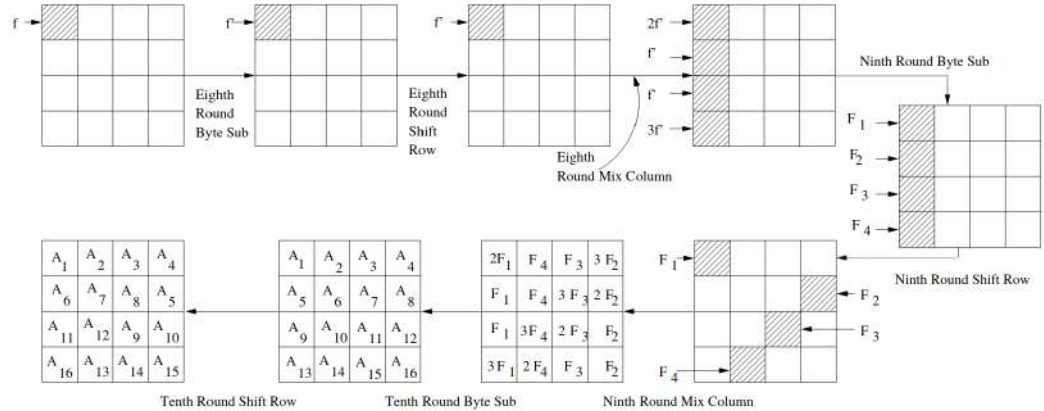
WHAT ELSE CAN WE BREAK?!

AES Basics



- Symmetric Key Crypto
- Encrypt messages for transfer over public channel
- Encrypt data for (untrusted) storage
- 4×4 byte state
- 10 rounds: S-Box, ShiftRows, MixColumns, AddRoundKey

Differential Fault Attack on AES



Michael Tunstall et al. "Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault". In: *International Workshop on Information Security Theory and Practices*. 2011

AES-NI (New Instructions)

Instruction	Description
`AESENC`	Perform one round of an AES encryption flow
`AESENCLAST`	Perform the last round of an AES encryption flow
`AESDEC`	Perform one round of an AES decryption flow
`AESDECLAST`	Perform the last round of an AES decryption flow
`AESKEYGENASSIST`	Assist in AES round key generation
`AESIMC`	Assist in AES Inverse Mix Columns
`PCLMULQDQ`	Carryless multiply (CLMUL)

AES New Instructions - inside SGX

```
do
{
  i++;
  plaintext = <randomly generated>

  result1 = aes128_enc(plaintext);
  result2 = aes128_enc(plaintext);
} while (vec_equal_128(result1,result2) && i<iterations);
```

```
bagger> sudo ./aes-encrypt 100000 -262
```

I have an idea for a logo!

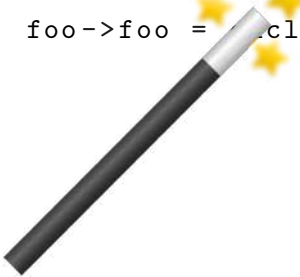
THIS IS MORE THAN JUST CRYPTO

But what about memory corruption?

```
struct_foo_t *foo = &arr[offset];
foo->foo = enclave_secret;
```

But what about memory corruption?

```
struct_foo_t *foo = &arr[offset];
foo->foo = enclave_secret;
```

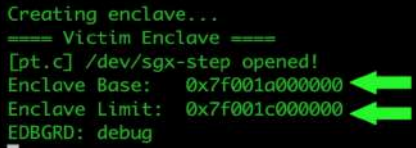


But what about memory corruption?

```
foo = arr + offset * 0x24
```

But what about memory corruption?

```
foo = arr + offset * 0x24
```

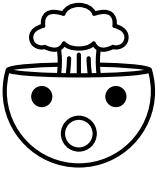


Voltage
0.584V

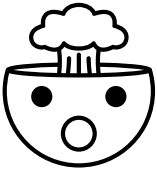
Undervolting
-235mV

More than just crypto?

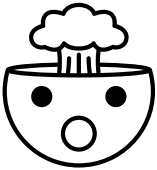
More than just crypto?



More than just crypto?

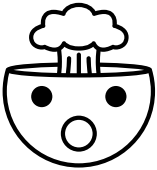


More than just crypto?



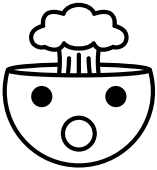
- RSA keys

More than just crypto?



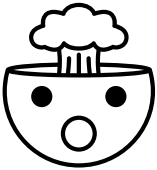
- RSA keys
- AES keys

More than just crypto?



- RSA keys
- AES keys
- Memory corruption

More than just crypto?



- RSA keys
- AES keys
- Memory corruption

$\rightarrow$ CryptoVaultScewHammer is not the ideal name!?

A better name

- Should be related to undervolting

A better name

- Should be related to undervolting
- From protected TEE vaults

A better name

- Should be related to undervolting
- From protected TEE vaults
- Steal

A better name

- Should be related to undervolting
- From protected TEE vaults
- Steal, corrupt

A better name

- Should be related to undervolting
- From protected TEE vaults
- Steal, corrupt, plunder, ...

A better name

Bob McGingerbread
@j77tbw

Replying to @shuckle

Plundervolt is such a great name

12:10 AM · Dec 11, 2019 · Twitter for iPhone

Trey Forgety @cincvolfIt · Dec 11

"**Plundervolt**" is the best **vulnonym** of 2019, and we had to wait until mid-December to get it.

betam4x / Ars Scholae Palatinae

POPULAR DEC 10, 2019 10:45 PM

Plundervolt? One of the better named exploits of the decade.

+42 (+42 / 0)

1042 posts | registered 12/23/2013

Jeffrey Joslin
@jeffemera

just pondering the genius behind the name “Plundervolt”

Ravi @ravikumark815 · 6h

Funky **name** for a funky exploit :D

#networksecurity #exploit #cybersecuirty #reverseengineering #technology

smistephen / Ars Praetorian / *et Subscriptor* DEC 10, 2019 11:24 PM

I'm wondering if it took longer to actually create Plundervolt, or come up with that double-pun of a name, haha.

+3 (+3 / 0) 457 posts | registered 7/31/2014

Daniel Cuthbert @dcuthbert · Dec 11

It might be me but I feel most vulnerability **names** today sound like bad 1980s porno titles

Plundervolt
KNOB attack
Spectre

8 2 28

Victor Fernando R. Ocampo @VictorOcampo · Dec 11

Plundervolt (sounds like a supervillain **name**) can steal all your chips' secure data -

Alex Ionescu @aionescu · Dec 11

I have it on good authority that **PlunderVolt** is the **name** of the villain in Incredibles 3. Missed opportunity for "The Underrrvolter!" (Shamelessly stolen from @bfosterjr)

Michael Kohl @citizen428 · 16h

I love how new exploits nowadays get cool **names**, landing pages etc. Anyway, **Plundervolt** is pretty interesting. plundervolt.com

1 1

mindovermiles262 19 points · 1 day ago

Can we take a moment and appreciate the quality of those videos? Wow 🤩

ObscureCulturalMeme 12 points · 1 day ago

Shit, even the website design is easy to read, and on mobile at that. Professional across the board.

Plundervolt TLDR

- A new type of attack

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX
- Within SGX we:

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX
- Within SGX we:
  - Retrieve keys from AES-NI

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX
- Within SGX we:
  - Retrieve keys from AES-NI
  - Retrieve RSA signature key

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX
- Within SGX we:
  - Retrieve keys from AES-NI
  - Retrieve RSA signature key
  - Induce memory corruption in bug free code

Plundervolt TLDR

- A new type of attack
- Breaks the integrity of SGX
- Within SGX we:
  - Retrieve keys from AES-NI
  - Retrieve RSA signature key
  - Induce memory corruption in bug free code
  - Make enclave write secrets to untrusted memory

Kit Murdock, David Oswald, Flavio D. Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. “Plundervolt: Software-based Fault Injection Attacks against Intel SGX”. In: *S&P*. 2020

Plundervolt

Flipping Bits from Software without Rowhammer

Kit Murdock, Daniel Gruss

December 27, 2019

Susan @SusanMaret · Dec 16

#Plundervolt "is a slightly ponderous pun on Thunderbolt...& the new vulnerability that has its own domain/website, HTTPS certificate, pirate-themed logo, & media-friendly strapline." #cybersecurity

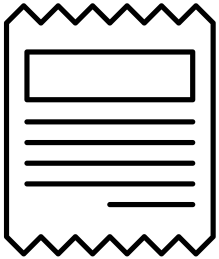
Plundervolt

Flipping Bits from Software without Rowhammer

Kit Murdock, Daniel Gruss

December 27, 2019

Acknowledgements



This research is partially funded by the Research Fund KU Leuven, and by the Agency for Innovation and Entrepreneurship (Flanders). Jo Van Bulck is supported by a grant of the Research Foundation – Flanders (FWO). This research is partially funded by the Engineering and Physical Sciences Research Council (EPSRC) under grants EP/R012598/1, EP/R008000/1, and by the European Union's Horizon 2020 research and innovation programme under grant agreements No. 779391 (FutureTPM) and No. 681402 (SOPHIA).