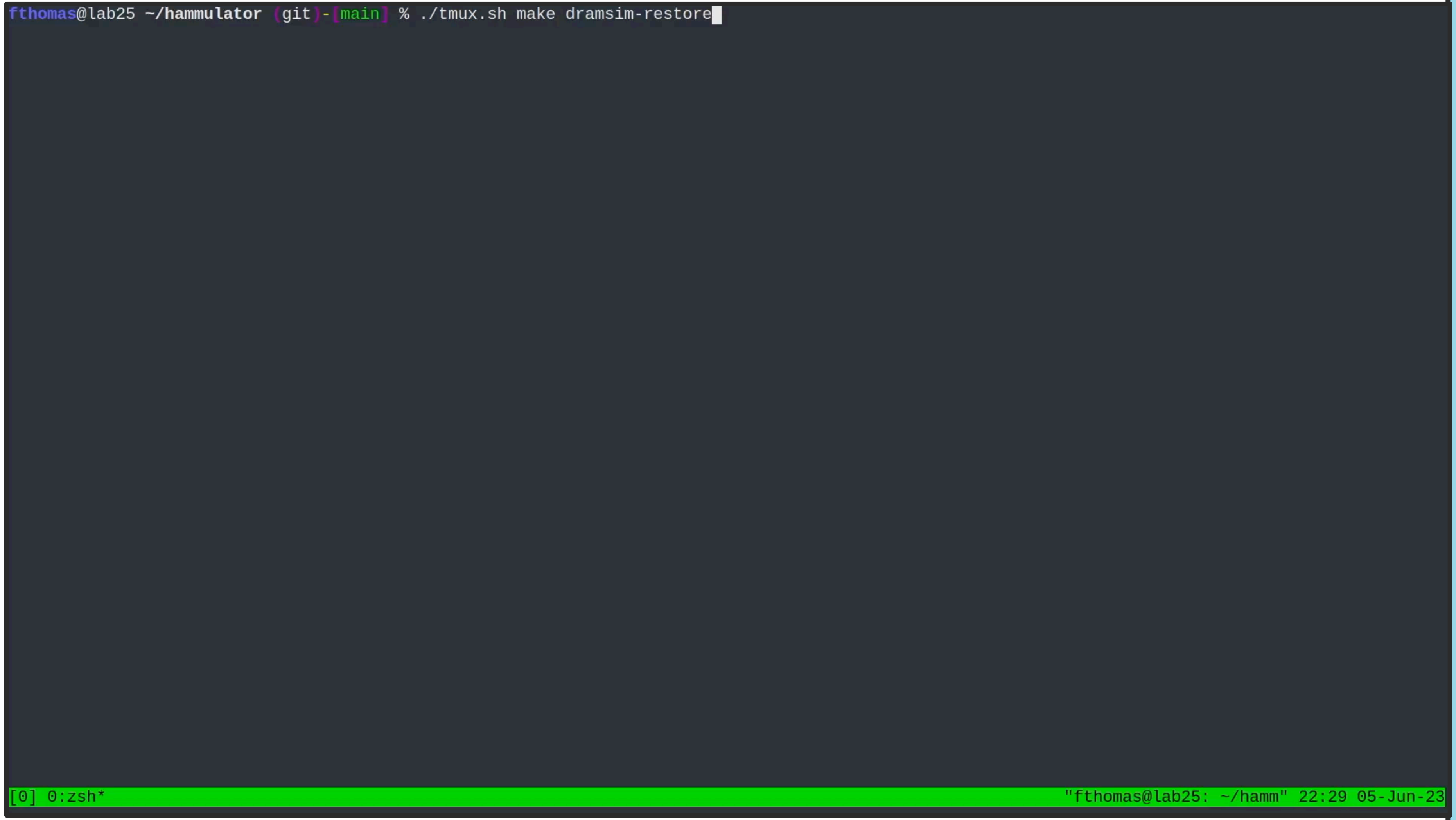
CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Rowhammer Revisited

From Exploration to Exploitation and Mitigation

Lukas Gerlach, Daniel Weber | m0leCon 2023 | 02.12.2023

Why Rowhammer?



```
fthomas@lab25 ~/hammulator (git)-[main] % ./tmux.sh make dramsim-restore
```

```
[0] 0:zsh*                                                    "fthomas@lab25: ~/hamm" 22:29 05-Jun-23
```

Code execution

↓

Privilege escalation

Who are we?

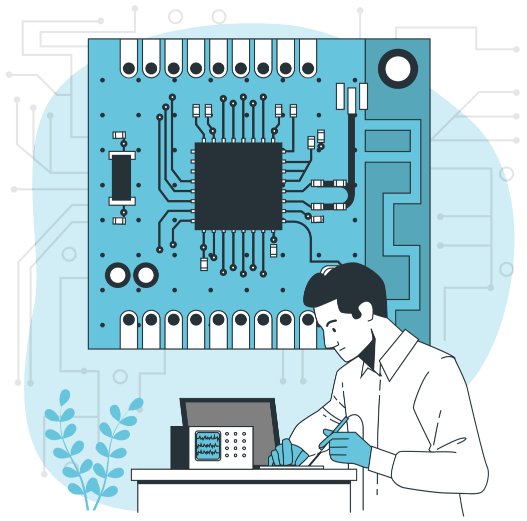
2 PhD Students

CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Lukas Gerlach

Daniel Weber

Rowhammer Revisited — Agenda



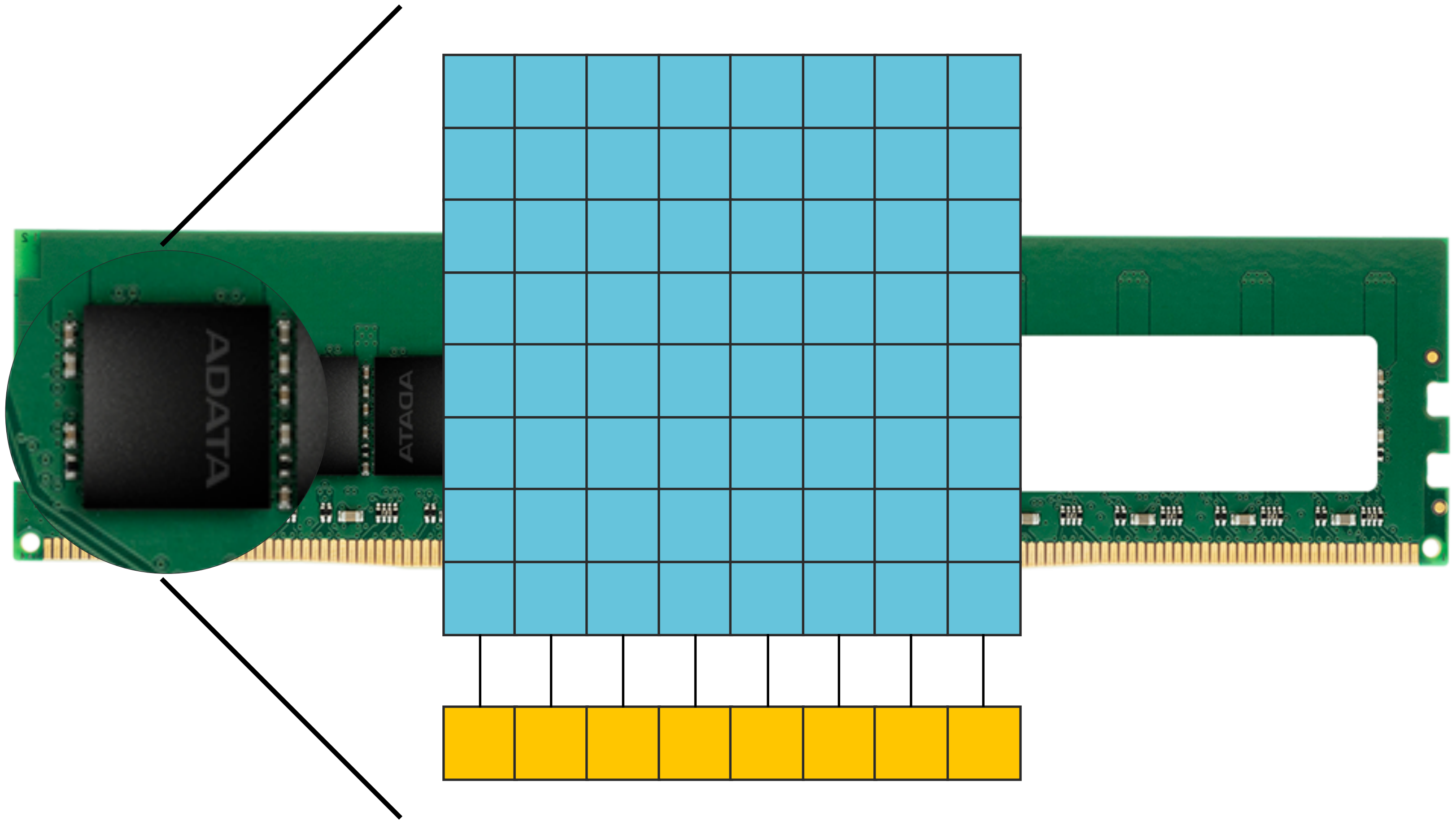
Exploration

Exploitation

Mitigations

Uhm… but what is Rowhammer?

Let’s Talk about DRAM



Inner Workings of DRAM

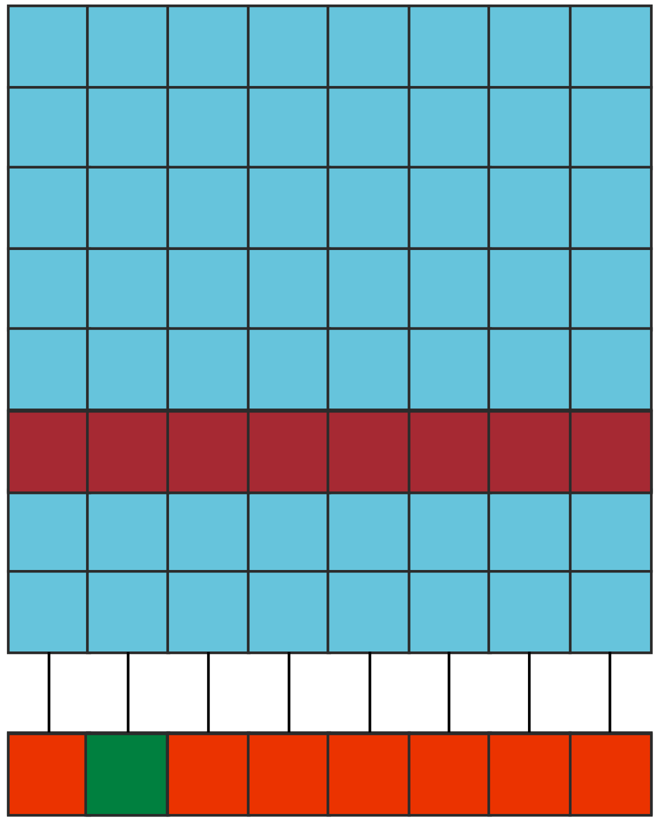
- DRAM consists of **memory cells** and a **row buffer**
- DRAM is **recharged periodically**
- **Read and write** on a DRAM chip is always **done per row**

Access:

I. Copy **the row** to the **row buffer**

II. Read **requested memory** from the row buffer

→ **Problem:** Rapid **row activations** drain the capacitors fasters



What happens if we rapidly access memory?

Experiment: Rapid Row Activation

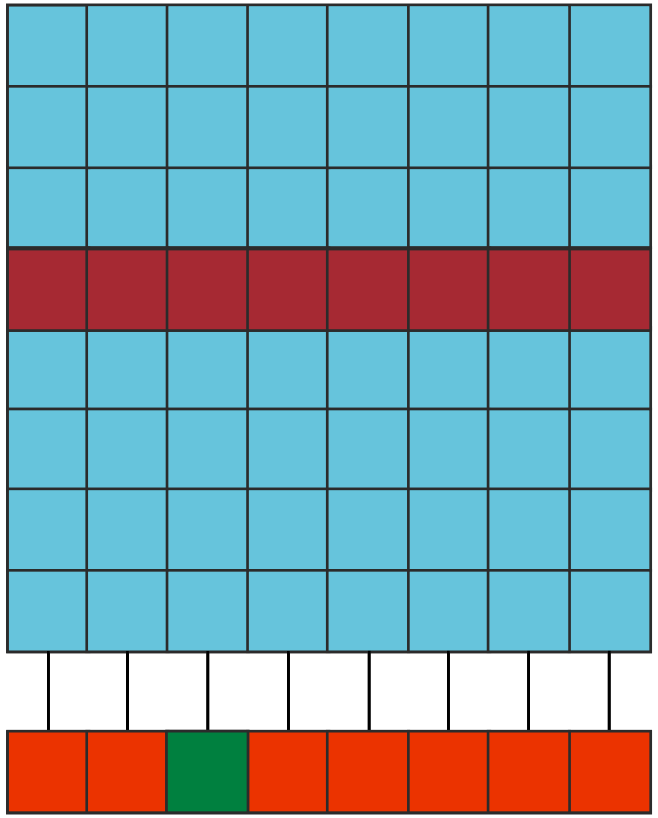
1. Memory Read:

I. Copy to **row buffer**

II. Read **memory** from **row buffer**

2. Memory Read:

I. Read **memory** from **row buffer**



How can we solve that?

Experiment: Rapid Row Activation (2. Try)

1. Memory Read (Row X):

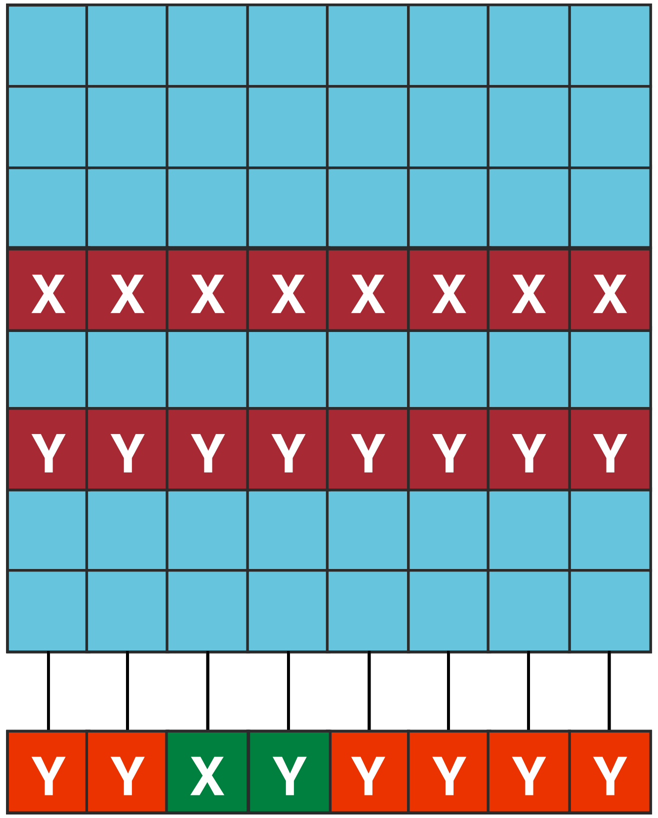
I. Copy to **row buffer**

II. Read **memory** from **row buffer**

2. Memory Read (Row Y):

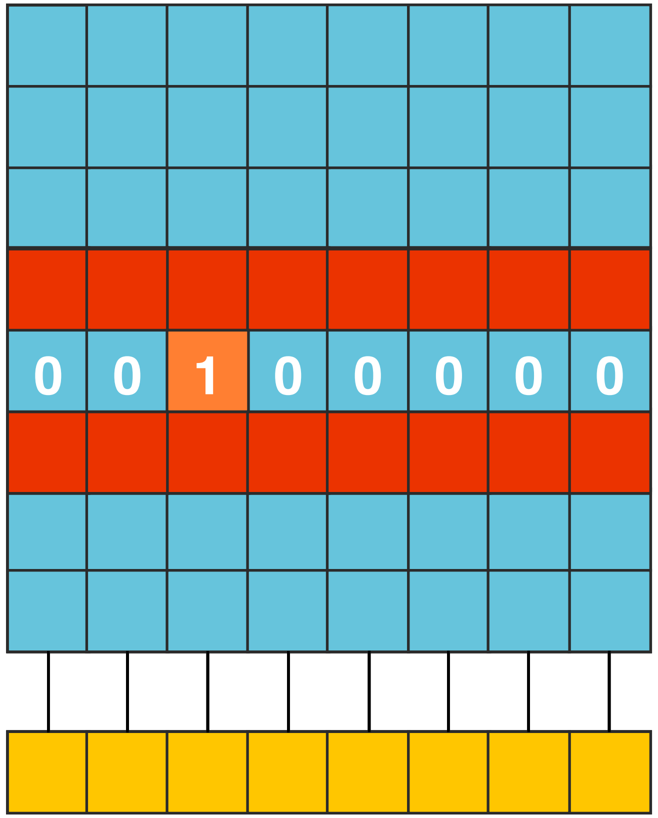
I. Copy to **row buffer**

II. Read **memory** from **row buffer**



Experiment: Rapid Row Activation (2. Try)

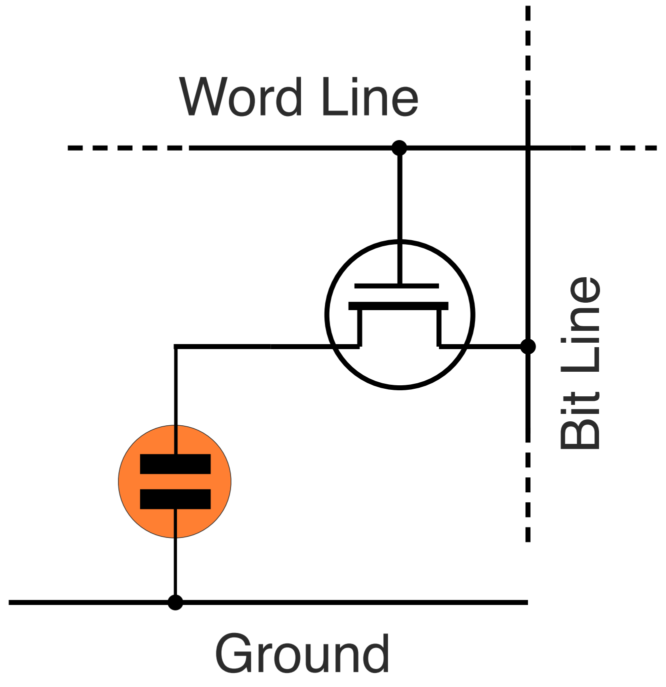
```
while (1) {
    memory_read(row_x);
    memory_read(row_y);
}
```



Root Cause of the Bitflips

- Rowhammer is a physical property of DRAM modules
- Influenced by how quickly **capacitors** in the module discharge

Depends on manufacturing differences and external factors



Exploration

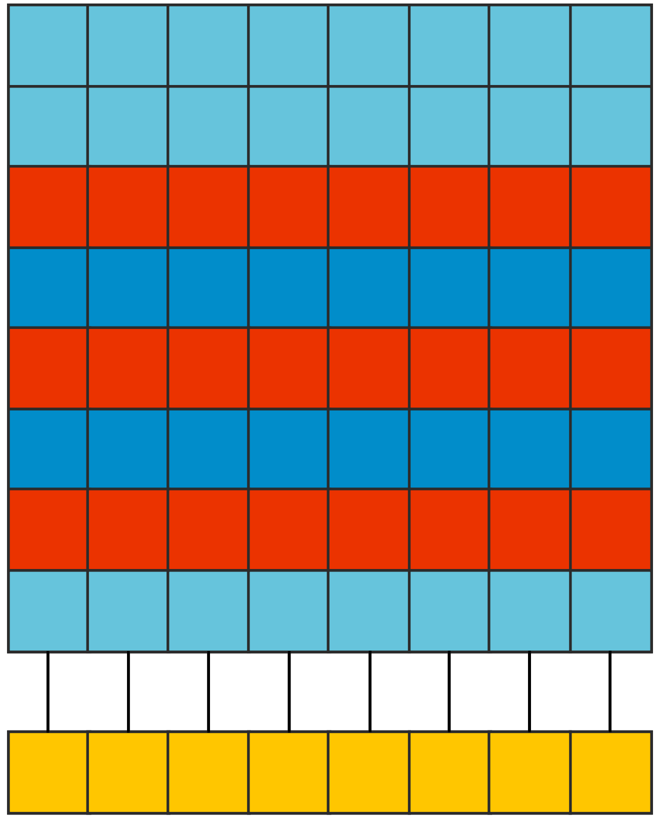
Do New Memory Modules Save Us?

DDR3: vulnerable

But I’m using DDR4!

Hammering Strategies

- Multiple strategies work
- Classified by the distribution and number of **attacker rows** and **victim rows**
  - Single Sided
  - Double Sided
  - N-Sided



Do New Memory Modules Save Us?

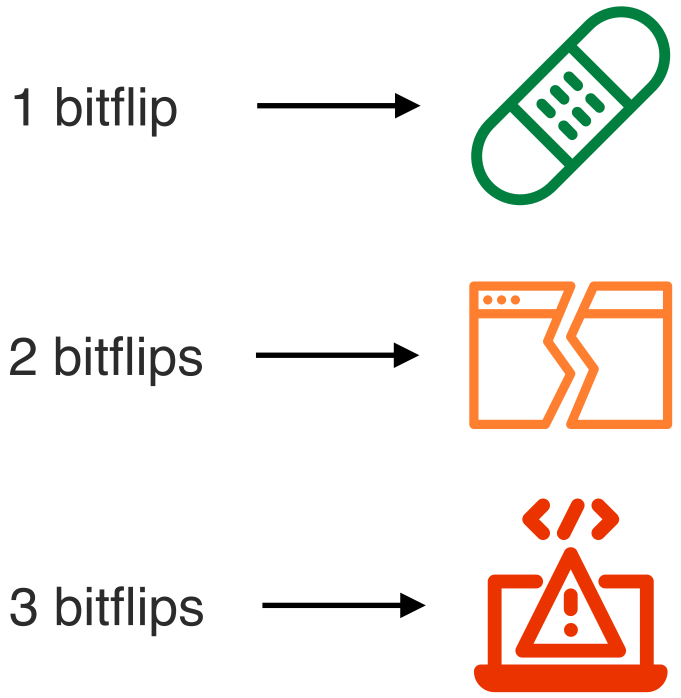
DDR4: vulnerable
DDR3: vulnerable

Ok ok, let’s just use ECC?

ECCploit

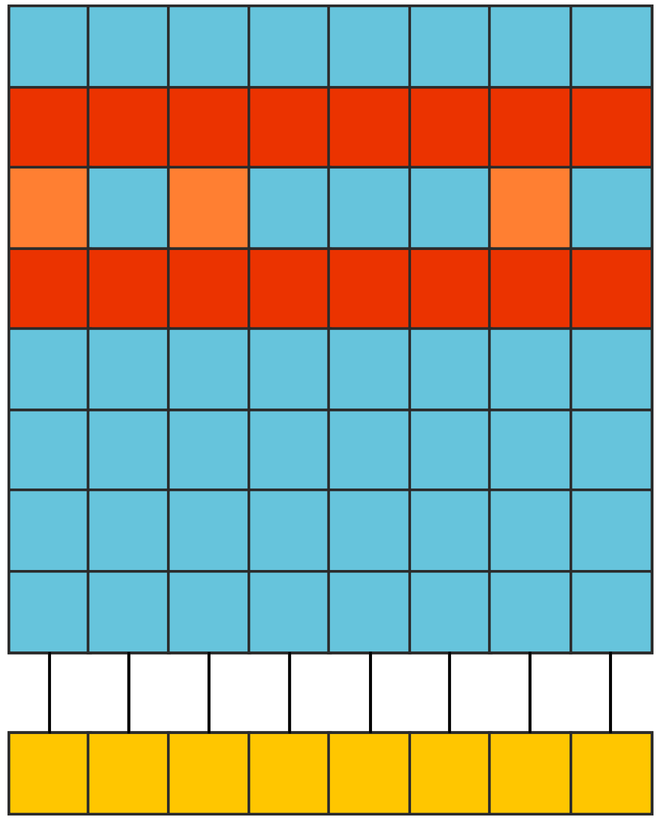
- ECC designed to **protect against single** random bitflips
- ECC can:
  - **Correct** 1 bitflip
  - **Detect** and **Crash** on 2 bitflips
  - **Fail** with 3 bitflips

Triple bitflips make Rowhammer on ECC possible



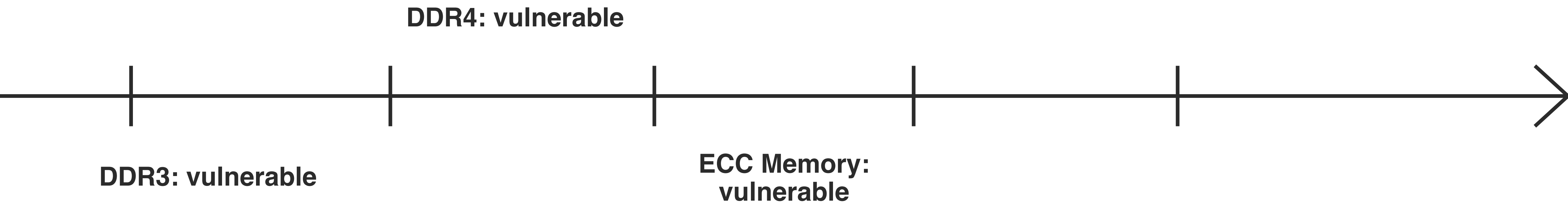
ECCploit details

- But how to trigger 3 bitflips at once?
- Find single bitflips that are silently corrected
- Combine them to overwhelm ECC



Cojocar, Lucian, et al.
"Exploiting correcting codes: On the effectiveness of ecc memory against rowhammer attacks."
S&P 2019

Do New Memory Modules Save Us?

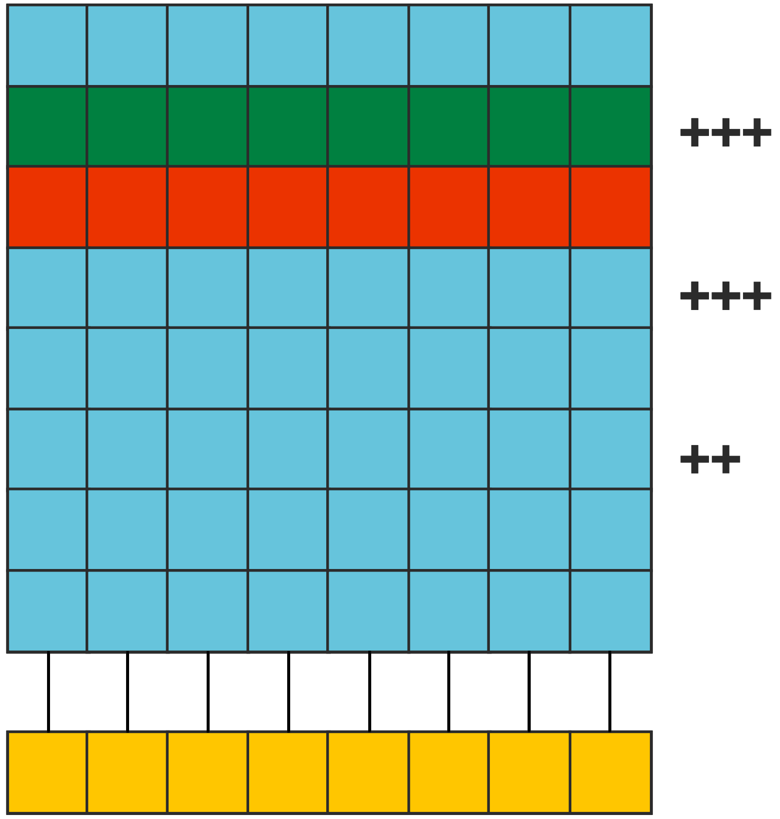


Can we maybe “harden” our DRAM?

Hardened DDR4 DRAM: TRR

- In DRAM mitigation deployed in DDR4
- Increment neighbour rows on **row activation**
- **Refresh** when counter reaches threshold

Rowhammer is fixed?



Problems with TRR

- Practical TRR has limitations
  - Randomly sample memory acess and refresh
  - Limited number of counters
- We do not need to stick to simple hammering patterns
- **Goal:** Trigger cases where TRR cannot refresh victim row

How to find such hammering patterns?

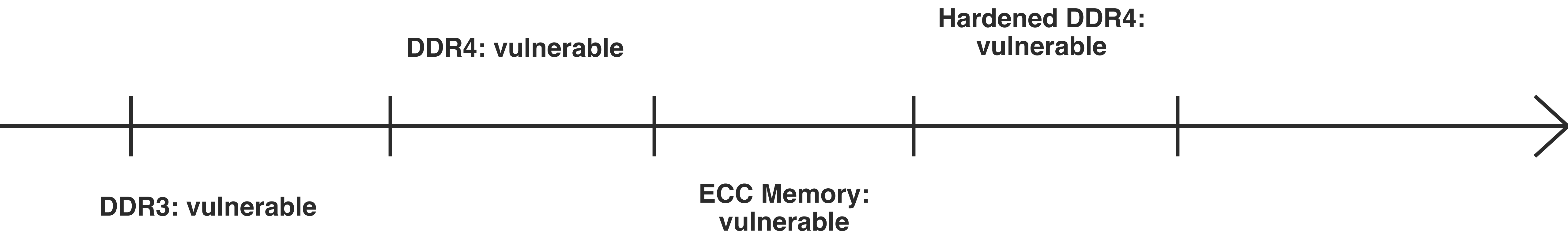
Fuzzing for good hammering patterns

- TRResspass
  - Specifically designed to bypass TRR
- Blacksmith
  - Generates new hammering patterns
  - Effective against TRR
  - Clever heuristics to generate good patterns

Frigo, Pietro, et al.
"TRRespass: Exploiting the many sides of target row refresh."
S&P2020.

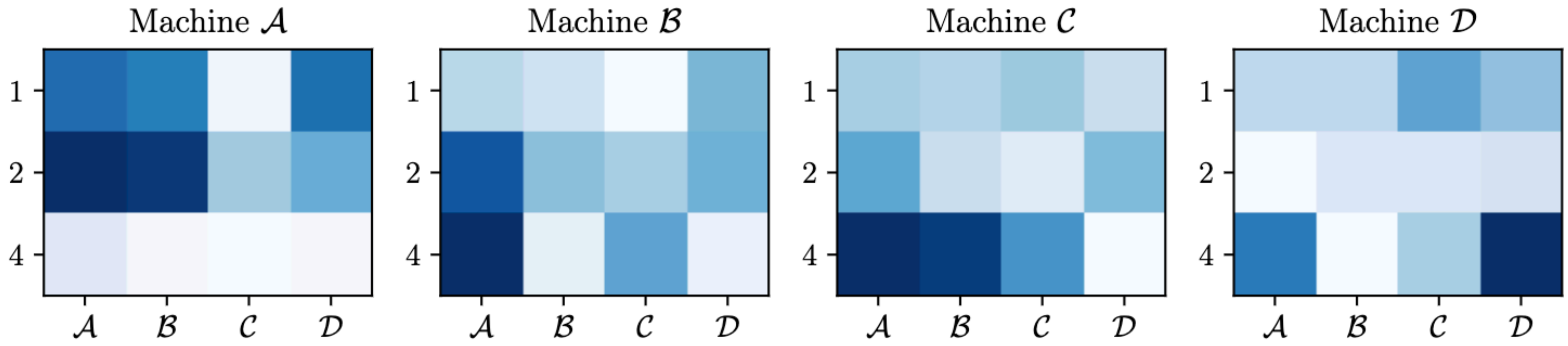
Jattke, Patrick, et al.
"Blacksmith: Scalable rowhammering in the frequency domain."
S&P 2022

Do New Memory Modules Save Us?



But I see different bitflips on my machine.

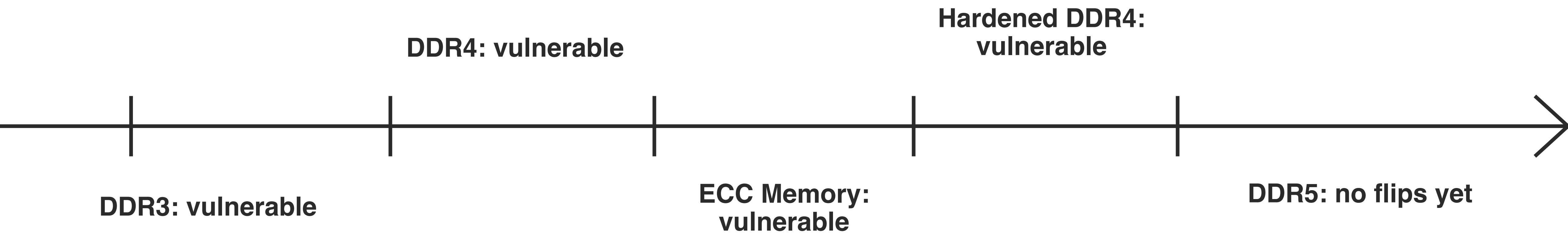
Do Bitflips Transfer Between Machines?



Result: Different machines → different bitflips

Gerlach, Lukas, et al.
"A Rowhammer Reproduction Study Using the Blacksmith Fuzzer."
ESORICS. 2023

Do New Memory Modules Save Us?



Future research will find out!

Exploitation

Pwn

- Page-table exploit
- Opcode flipping

Page-Table Exploitation with Rowhammer

- Page-Table Entries (PTEs) **control access rights**
- PTEs contain **pointer to controlled memory**
- PTEs are **stored in memory**

I. **Allocate** a lot of memory pages (and PTEs).

II. **Flip pointer** of PTE X

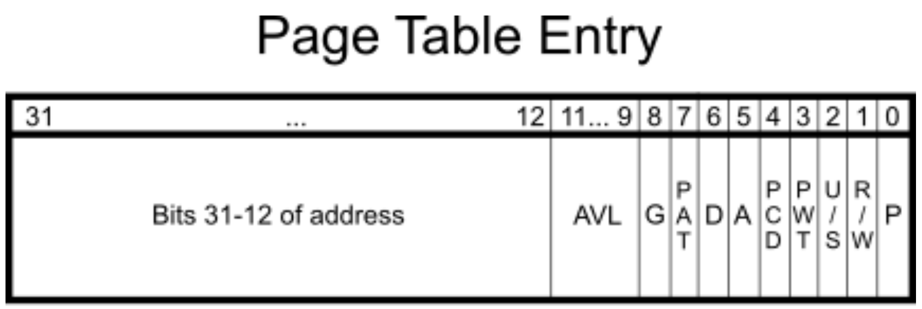
III. Hope that the pointer of **PTE X now points to one of your PTEs**

IV. If so: you got **read/write access to your own PTE**

Success: Mapping of PTE X can be used to modify PTE.

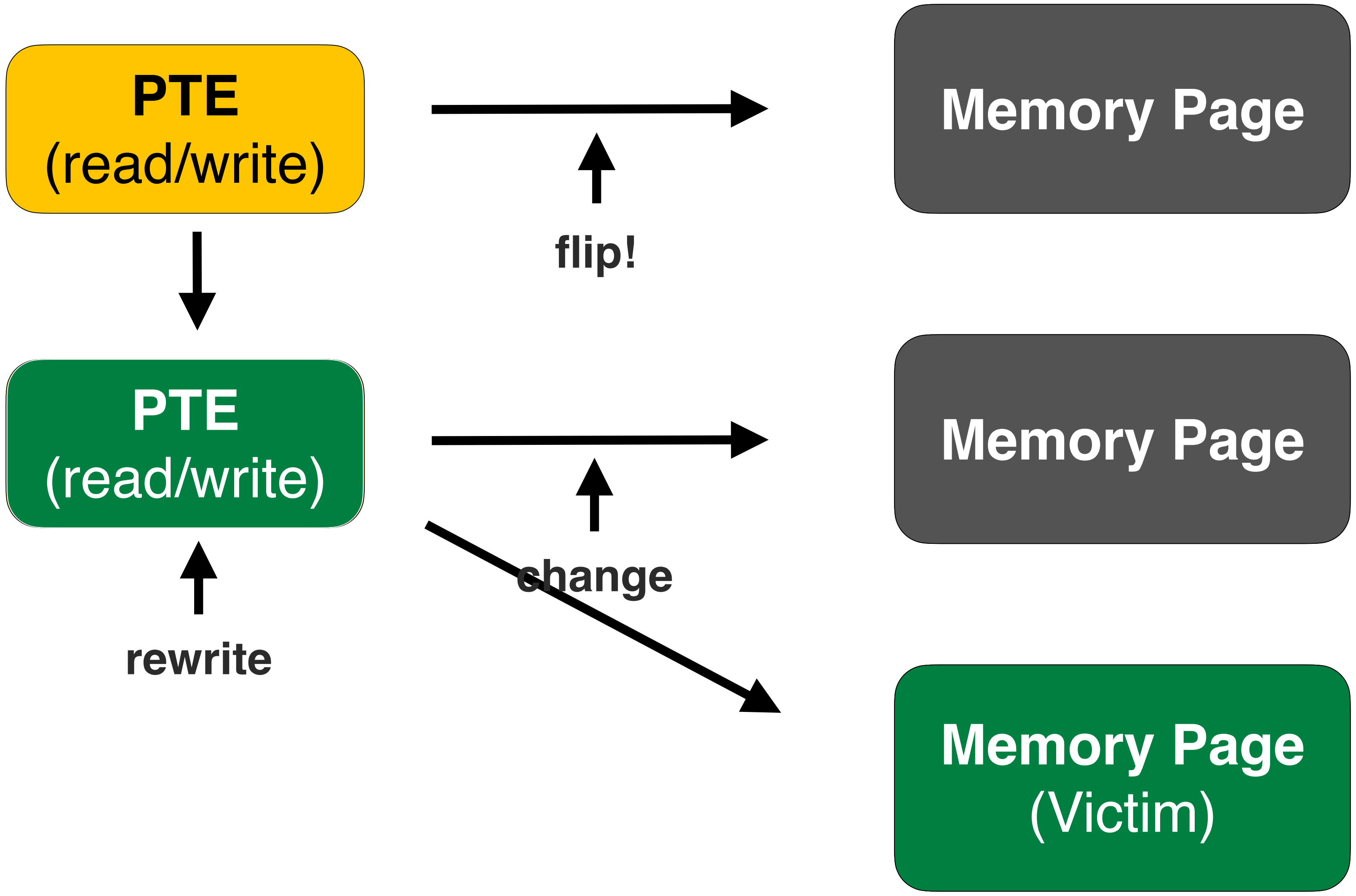
Allows **reading/writing arbitrary addresses**

to access an arbitrary address



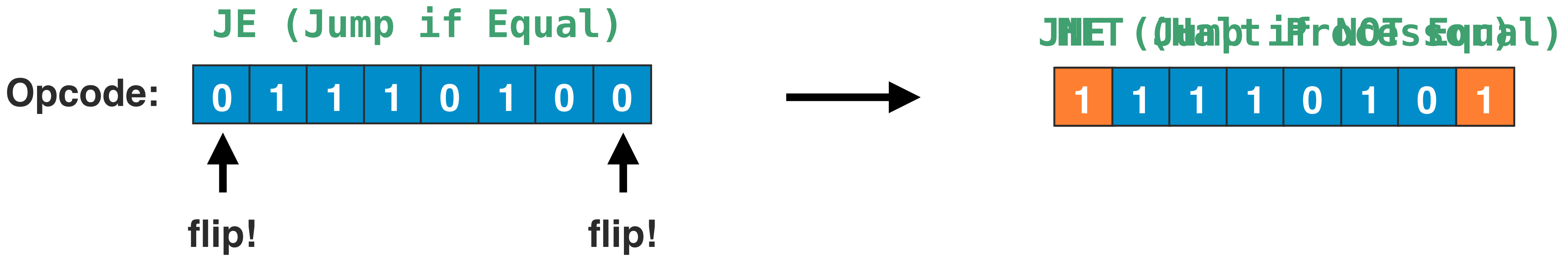
P: Present	**D:** Dirty
R/W: Read/Write	**G:** Global
U/S: User/Supervisor	**AVL:** Available
PWT: Write-Through	**PAT:** Page Attribute Table
PCD: Cache Disable	
A: Accessed	

Page-Table Exploitation with Rowhammer



But why stop at flipping pointers?

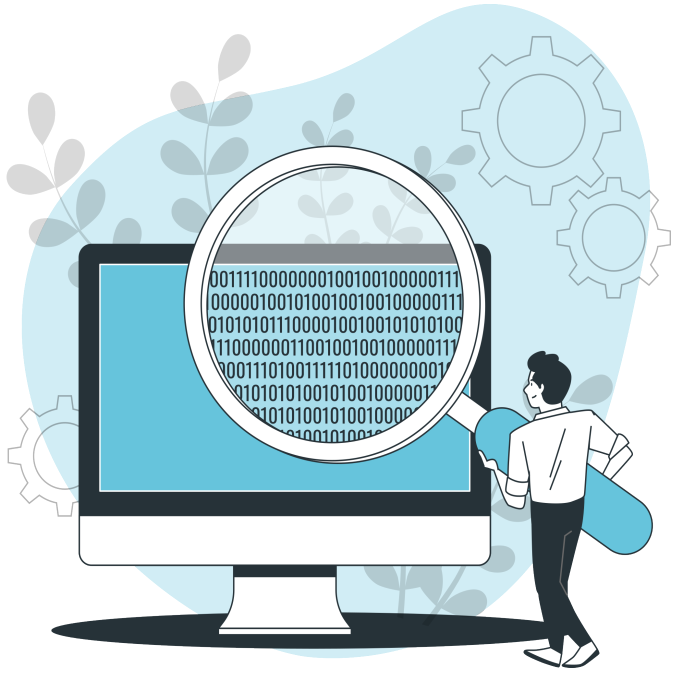
Opcode Flipping



Bitflips — Impact

- Exploit **sudo** binary to gain **root**
- Escape **browser sandbox** for **arbitrary code execution**
- Escape from **virtual machines**
- And **many more**…

Be creative!



Pwn

- Page-table exploit
- Opcode flipping

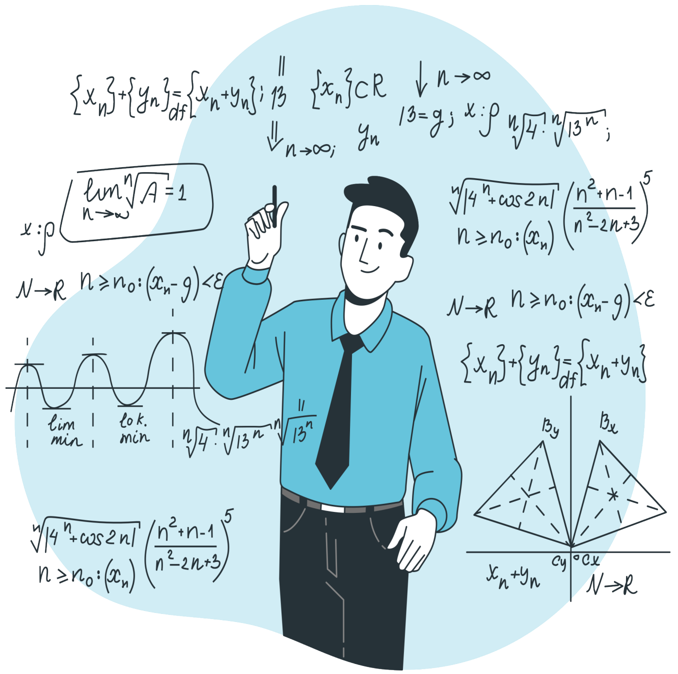
Crypto

- Bellcore attack with Rowhammer
- Attack on RSA modulus

Bellcore Attack

- Forge RSA-CRT Signatures
- Needs 2 signatures
  - Normal signature S
  - Faulty signature S'
- Now magically $gcd(S' - S, N) = q$

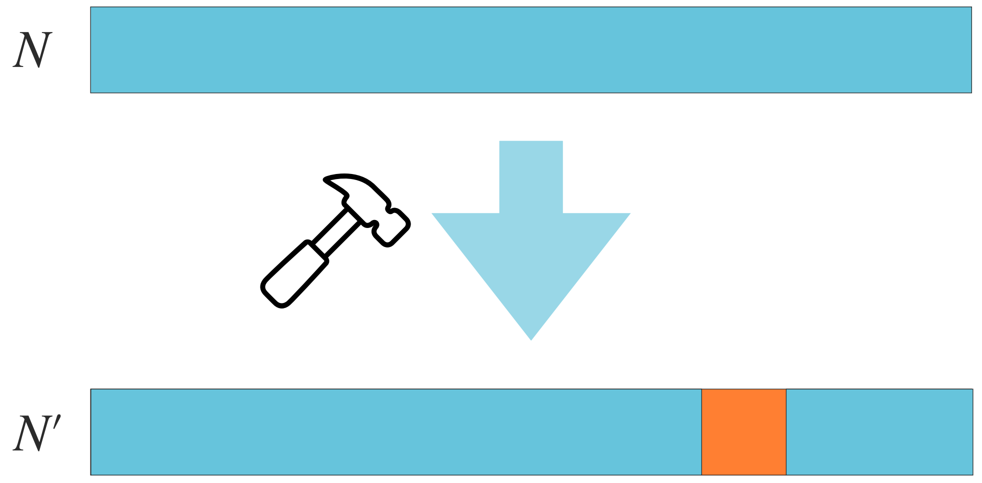
We can forge arbitrary signature with a single fault



RSA modulus attack

- Hammer the RSA modulus N before victim encrypts
- Victim encrypts with corrupted modulus N'

If N' has 1024-2048 bits, It can factorized efficiently with probability of 12–22%



Razavi, Kaveh, et al.
"Flip feng shui: Hammering a needle in the software stack."
USENIX Security 2016

Pwn

- Page-table exploit
- Opcode flipping

Crypto

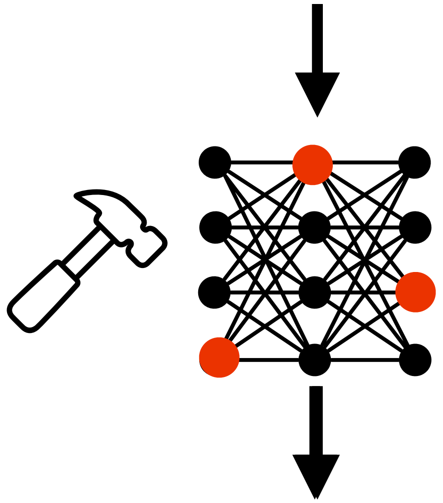
- Bellcore attack with Rowhammer
- Attack on RSA modulus

Misc

- Attack on Neural Networks
- Rapid Prototyping

Performance Degradation on Neural Networks

- Bitwise corruption
- Attacker can induce over 90% accuracy loss in Neural networks
- No knowledge of network architecture needed



Panda

Cat

Hong, Sanghyun, et al.
"Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks."
USENIX Security 2019

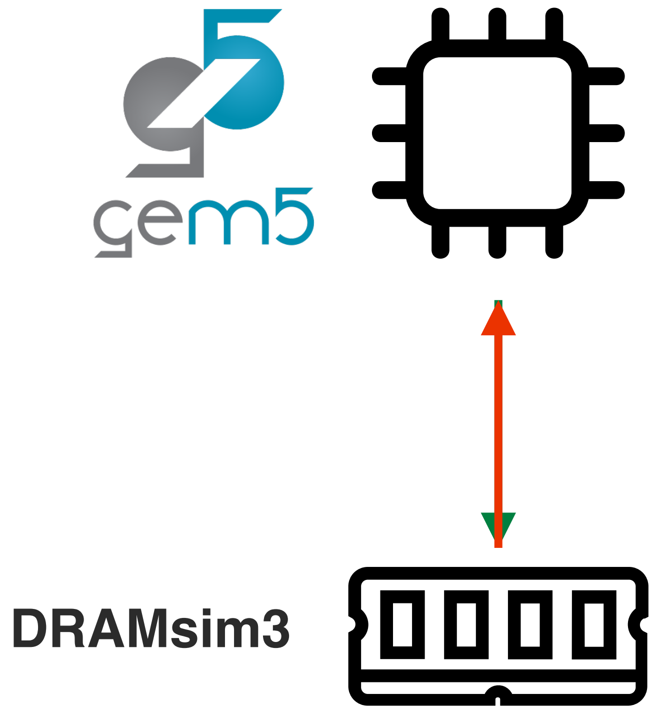
But how do I debug my Rowhammer exploit?

Prototyping Exploits

- Testing with real Rowhammer **not practical** for complex exploits
- Two options:
  - **Simple:** Inject bitflips via kernel module
  - **Accurate:** Simulate the whole system, with Rowhammer included

Hammulator: gem5 based Rowhammer Simulation

- Simulates **whole system** including timing
- How it works
  - **Simulate CPU** with gem5
  - **Simulate DRAM** with DRAMsim3
- **Memory requests** can **signal bitflips** if Rowhammer threshold exceeded



Thomas, Fabian, Lukas Gerlach, and Michael Schwarz.
"Hammulator: Simulate Now-Exploit Later."
3rd Workshop on DRAM Security. 2023.

Mitigation

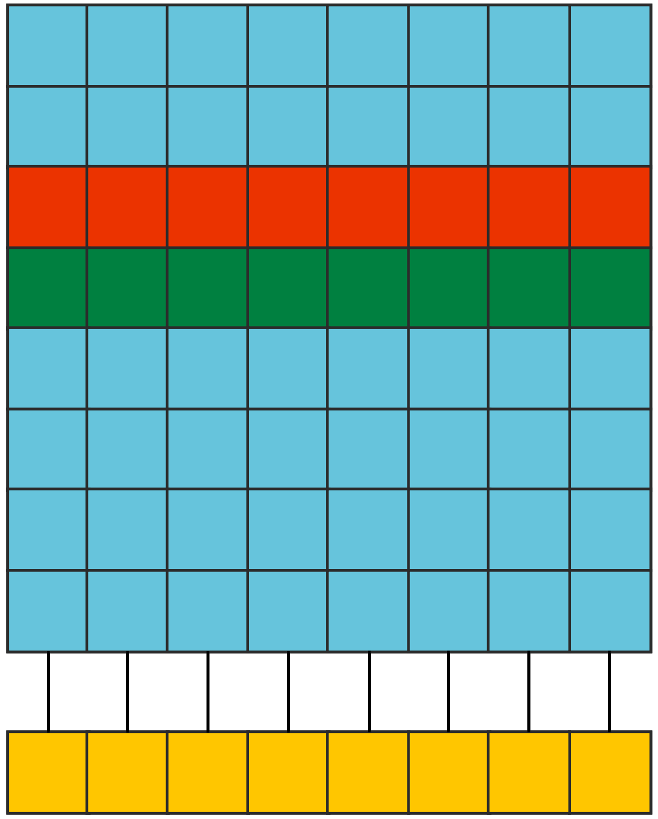
What Does Not Work

- Doubling refresh rate
  - Smaller refresh still enough for attacks
  - More power consumption, less performance
- Using ECC memory
  - Small number of bitflips can be corrected
  - Attacker can **overwhelm** ECC
- TRR
  - Special hammering patterns can **bypass** TRR

So what works?

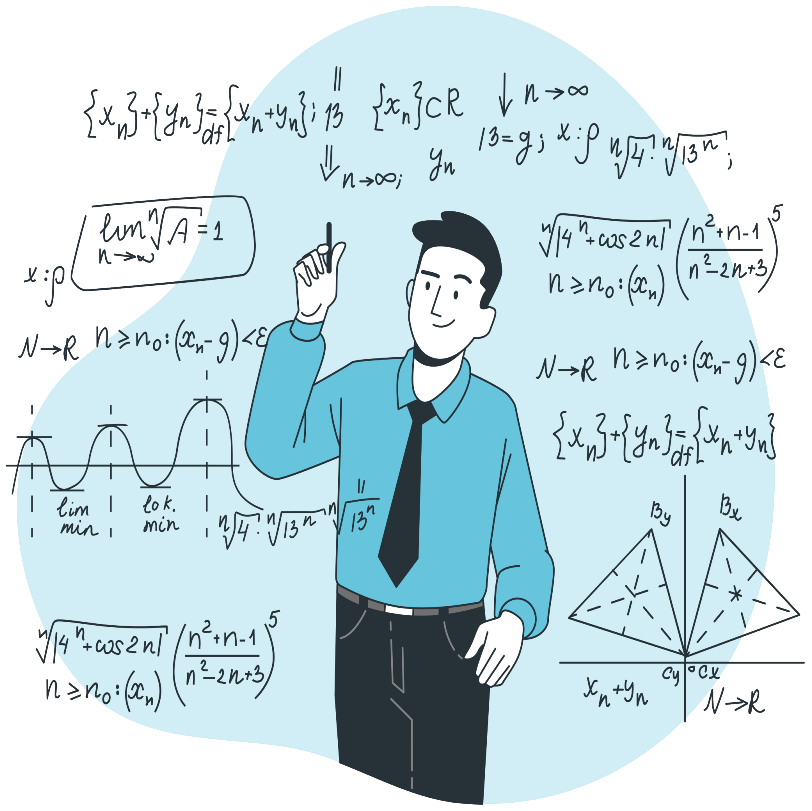
Probabilistic Mitigation: PARA

- Rowhammer is stochastic so is PARA
- Choose small probability $p \ll 1$
- On **row activation** do a **refresh** of adjacent row with probability p



+ Only memory controller changes
- No strong guarantees

Kim, Yoongu, et al.
"Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors."
ACM SIGARCH 2014



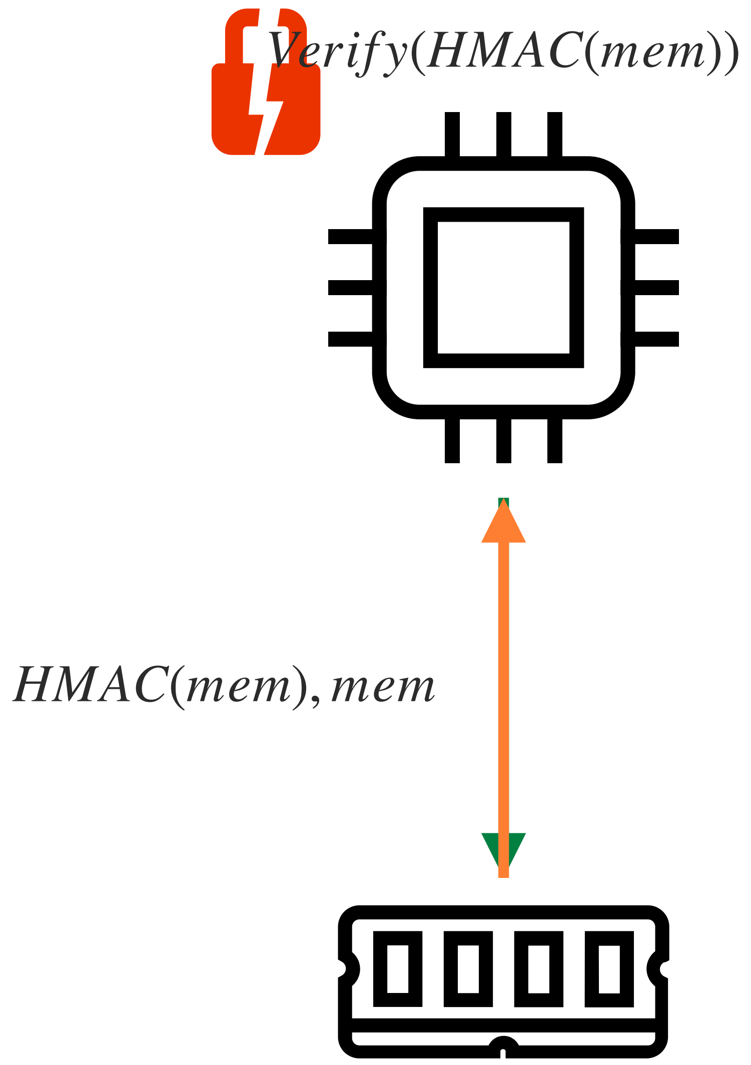
But we can have stronger guarantees!

Mitigate in the CPU: CSI Rowhammer

- Each **memory request** is **answered** with the memory content and a HMAC
- CPU has special hardware to quickly verify HMAC
- On errors the CPU **brute forces** the HMAC and correct the error

+Strong Guarantees
- CPU changes required

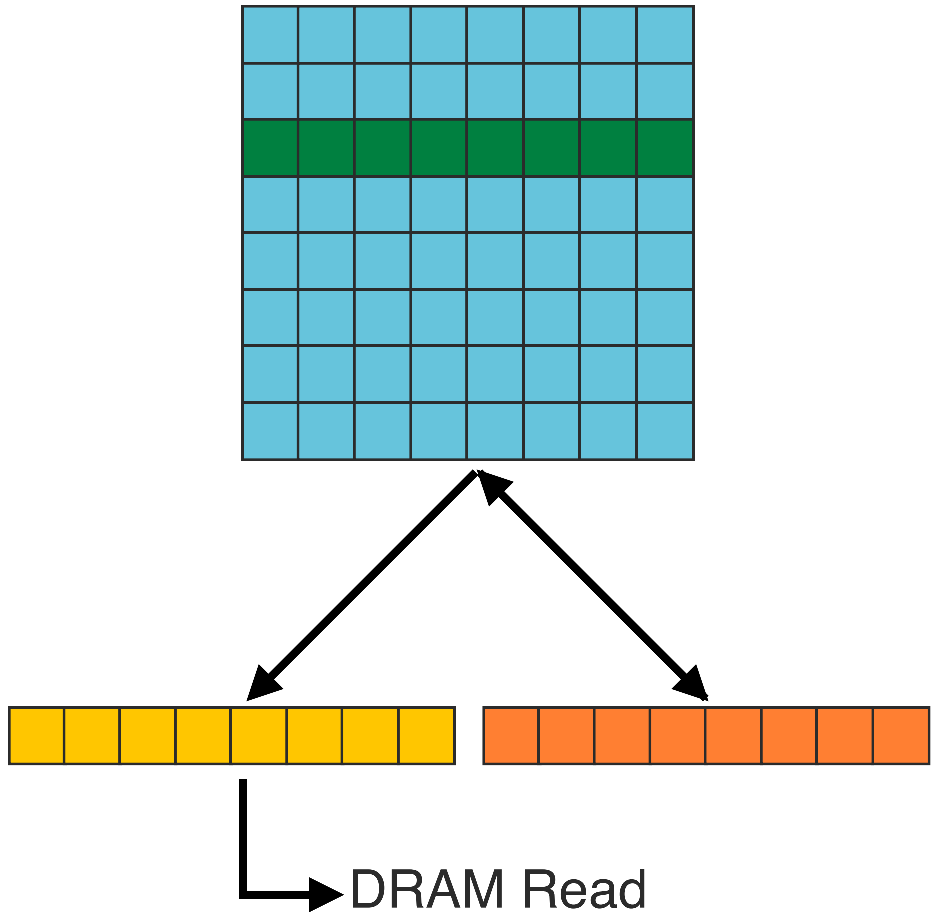
Juffinger, Jonas, et al.
“CSI: Rowhammer–Cryptographic security and integrity against rowhammer.”
S&P 2023



Changing the DRAM: Rega

- Mitigate in DRAM chip
- Parallel read/write and row refresh
- On Read request:
  - Write memory to **read buffer**
  - Refresh over **refresh buffer**

+Good guarantees, little overhead
- Requires new DRAM modules



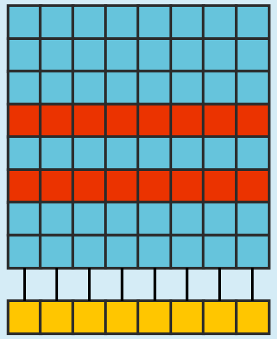
Marazzi, Michele, et al.
"REGA: Scalable Rowhammer Mitigation with Refresh-Generating Activations."
S&P 2023

Conclusion

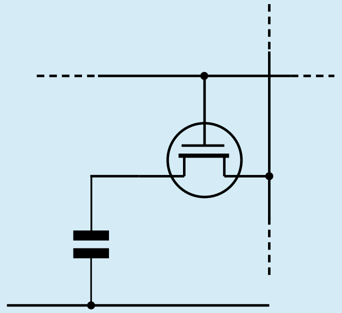
Links to Everything we Talked About

https://github.com/s8lvg/rowhammer-revisited-talk

What We Saw Today



Hammering Techniques



Physical Properties

Exploits

Mitigations

CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Rowhammer Revisited

From Exploration to Exploitation and Mitigation

Lukas Gerlach, Daniel Weber | m0leCon 2023 | 02.12.2023