Comp 590-184: Hardware Security and Side-Channels

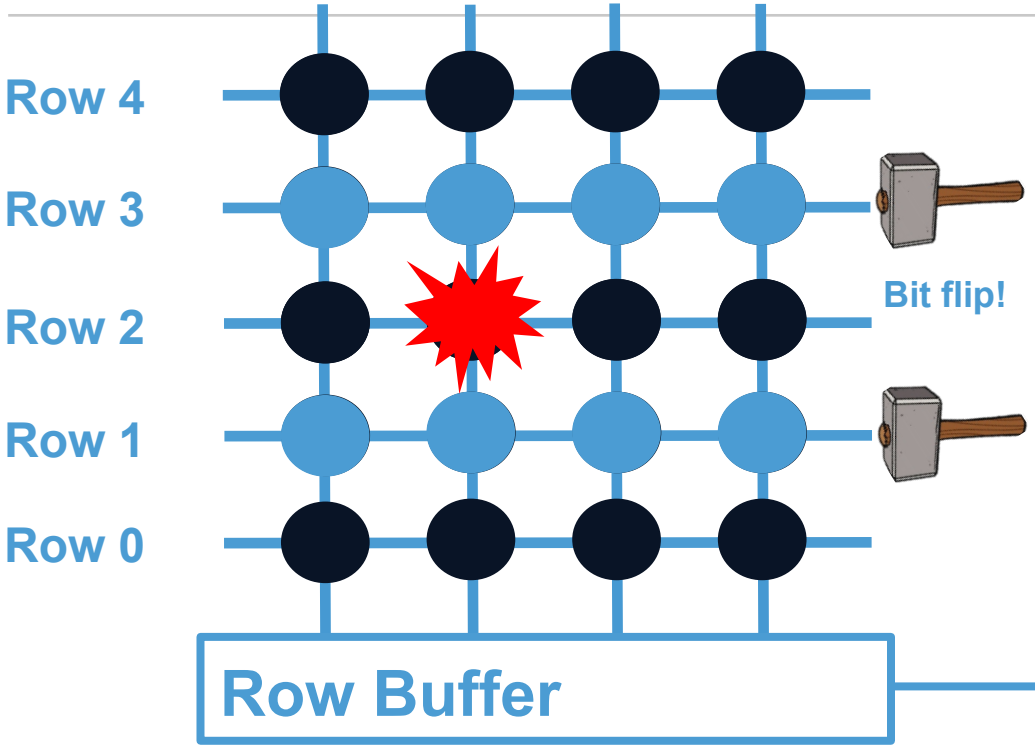
Lecture 19: Rowhammer Continued

March 31, 2026

Andrew Kwong

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Rowhammer



- Activating a row drains charge from nearby capacitors
- Repeated activation of rows causes bit flips in nearby rows!
- Attacker that controls values in rows 1 and 3 writes to victim's memory in row 2

Native Client (NaCl) Sandbox Escape

- NaCl is a sandbox for running native code (C/C++)
- Runs a “safe” subset of x86, statically verifying an executable
- Use bit flips to make an instruction sequence unsafe!

Example “Safe” Code:

```
andl $~31, %eax   // Truncate address to 32 bits
                  // and mask to be 32-byte-aligned.
addq %r15, %rax   // Add %r15, the sandbox base address.
jmp *%rax         // Indirect jump.
```

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn and Dullien)

Native Client (NaCl) Sandbox Escape

We can flip bits to allow for (unsafe) non 32-byte-aligned jumps!

Exploited “Safe” Code:

```
andl $~31, %ecx  // Truncate address to 32 bits
                 // and mask to be 32-byte-aligned.
addq %r15, %rax  // Add %r15, the sandbox base address.
jmp *%rax        // Indirect jump.
```

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn and Dullien)

- What other bad things can rowhammer do?
  - Talk with classmates

Kernel Privilege Escalation

What could happen if a user could gain direct write access to a page table?

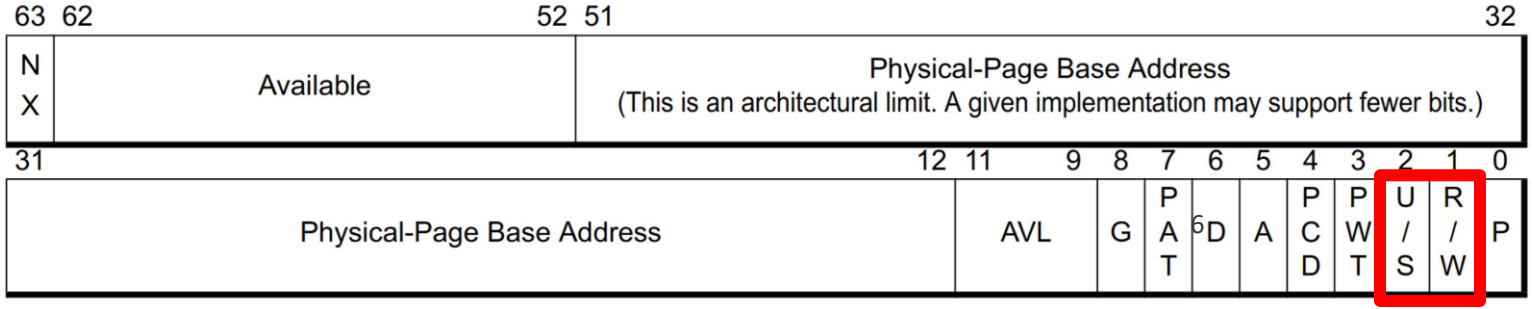
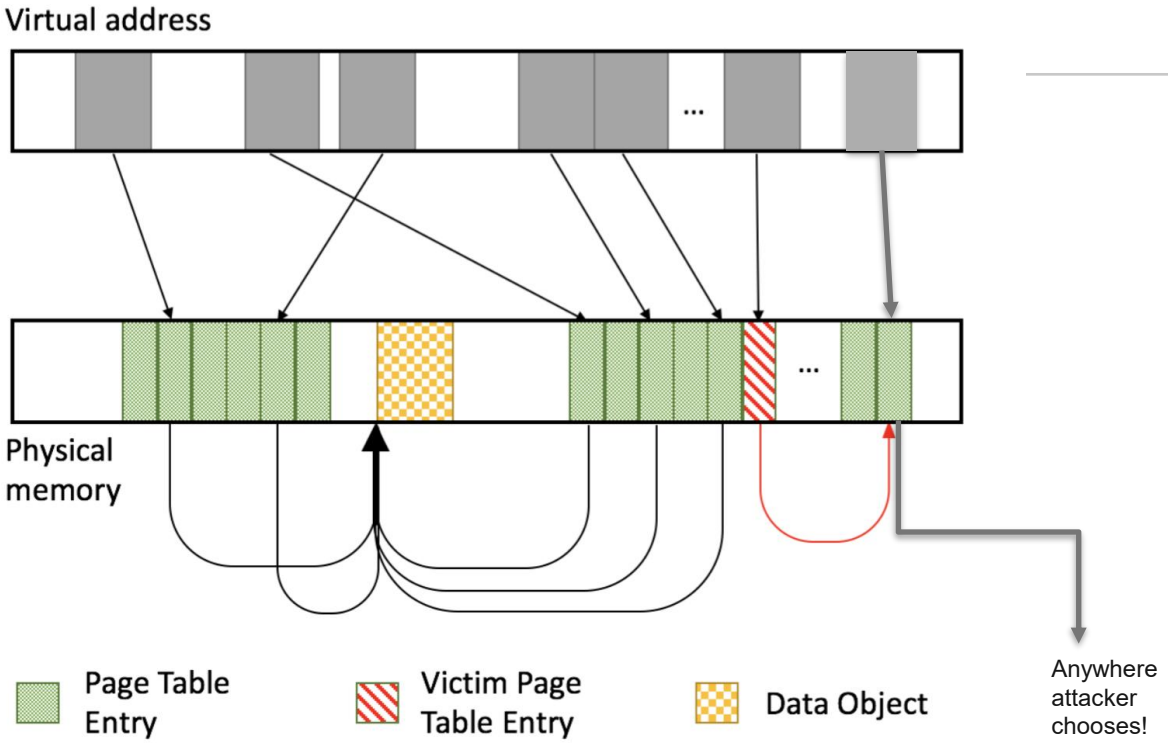
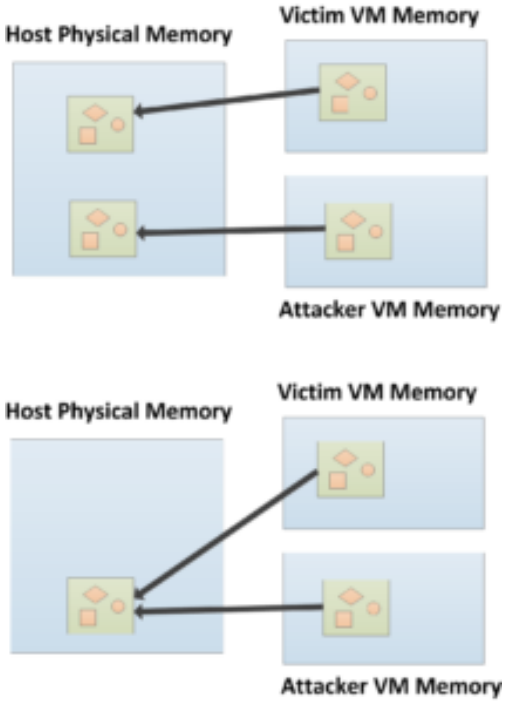


Figure 5-21. 4-Kbyte PTE—Long Mode


Virtual address
...
Physical memory
...
Page Table Entry
Victim Page Table Entry
Data Object
Anywhere attacker chooses!

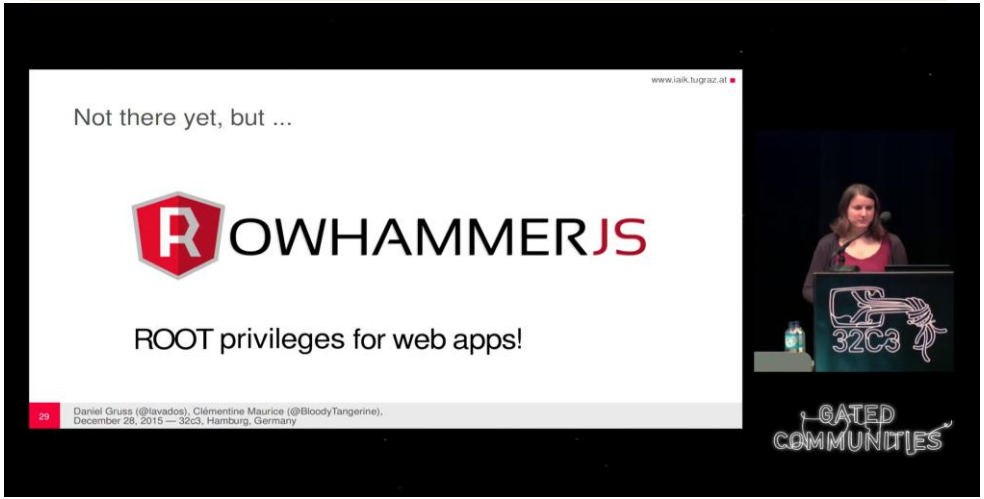
VM Takeover

- Cloud providers use de-duplication to conserve memory
  - Equivalent pages point to same physical memory
    - Copy-on-write (COW)
  - Hypervisor routinely scans memory for matching pages
- Attacker matches page against public RSA key
  - Deduplicates to flappable memory location
  - Flip bit in the key to weaken it



Rowhammer.js

- No clflush in browsers anymore
  - Because of rowhammer
- Evicted quickly enough for double-sided hammering



Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA'16)

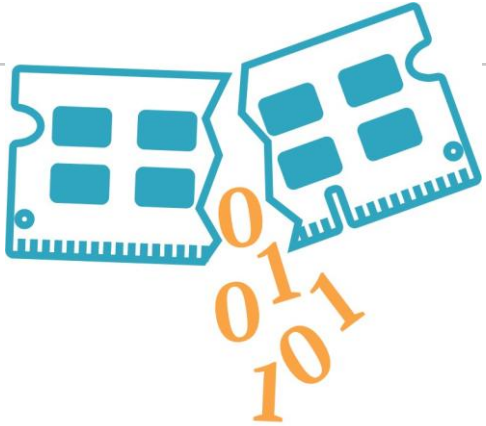
More Security Implications (II)

"Can gain control of a smart phone deterministically"

Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, CCS'16

https://fossbytes.com/drammer-rowhammer-attack-android-root-devices/

- IEEE S&P 2020



RAMBleed

RAMBleed: Reading Bits in Memory Without Accessing Them

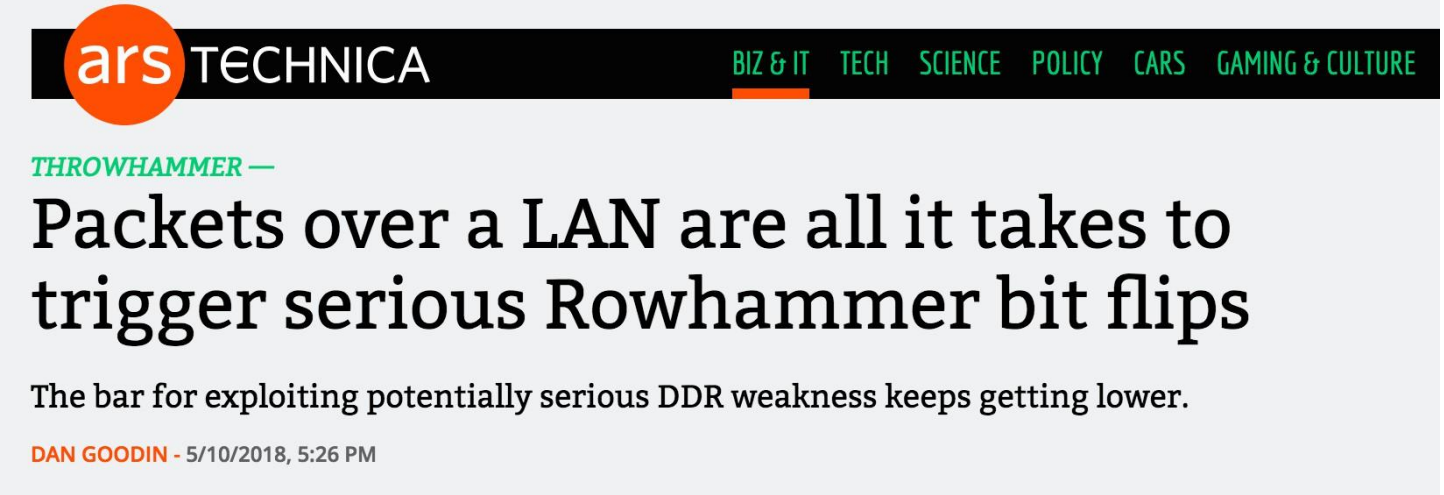
Andrew Kwong
University of Michigan
ankwong@umich.edu

Daniel Genkin
University of Michigan
genkin@umich.edu

Daniel Gruss
Graz University of Technology
daniel.gruss@iaik.tugraz.at

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

- Rowhammer over RDMA (I) USENIX ATC 2018



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

THROWHAMMER —

Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM

Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar
VU Amsterdam

Radhesh Krishnan
VU Amsterdam

Elias Athanasopoulos
University of Cyprus

Cristiano Giuffrida
VU Amsterdam

Herbert Bos
VU Amsterdam

Kaveh Razavi
VU Amsterdam

Rowhammer over RDMA (II)

The Hacker News™
Security in a serious way

Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests

Nethammer:
Inducing Rowhammer Faults through Network Requests

Moritz Lipp
Graz University of Technology

Misiker Tadesse Aga
University of Michigan

Michael Schwarz
Graz University of Technology

Daniel Gruss
Graz University of Technology

Clémentine Maurice
Univ Rennes, CNRS, IRISA

Lukas Raab
Graz University of Technology

Lukas Lamster
Graz University of Technology

- USENIX Security 2019

Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks

Sanghyun Hong, Pietro Frigo†, Yiğitcan Kaya, Cristiano Giuffrida†, Tudor Dumitraş

University of Maryland, College Park
†Vrije Universiteit Amsterdam

A Single Bit-flip Can Cause Terminal Brain Damage to DNNs

One specific bit-flip in a DNN's representation leads to accuracy drop over 90%

Our research found that a specific bit-flip in a DNN's bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...

Read More

- USENIX Security 2020

DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

Fan Yao
University of Central Florida
fan.yao@ucf.edu

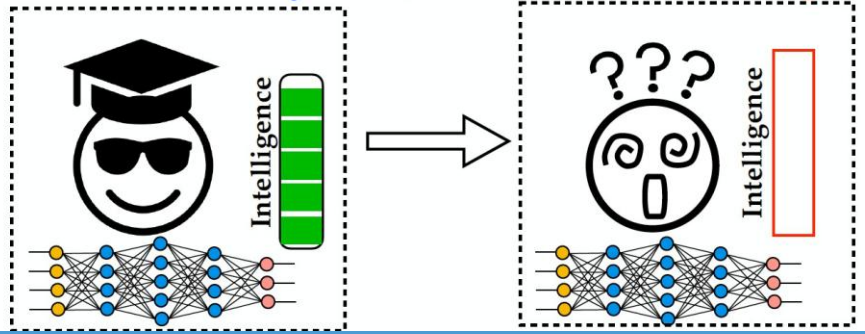
Adnan Siraj Rakin
Arizona State University
asrakin@asu.edu

Deliang Fan
dfan@asu.edu

Degrade the **inference accuracy** to the level of **Random Guess**

Example: ResNet-20 for CIFAR-10, **10** output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**



Rowhammer Mitigations?

- Bit flips everywhere!
  - Serious security concern
- What do we do now?

Apple's Security Patch for RowHammer

- https://support.apple.com/en-gb/HT204934

Available for: OS X Mountain Lion v10.8.5, OS X Mavericks v10.9.5

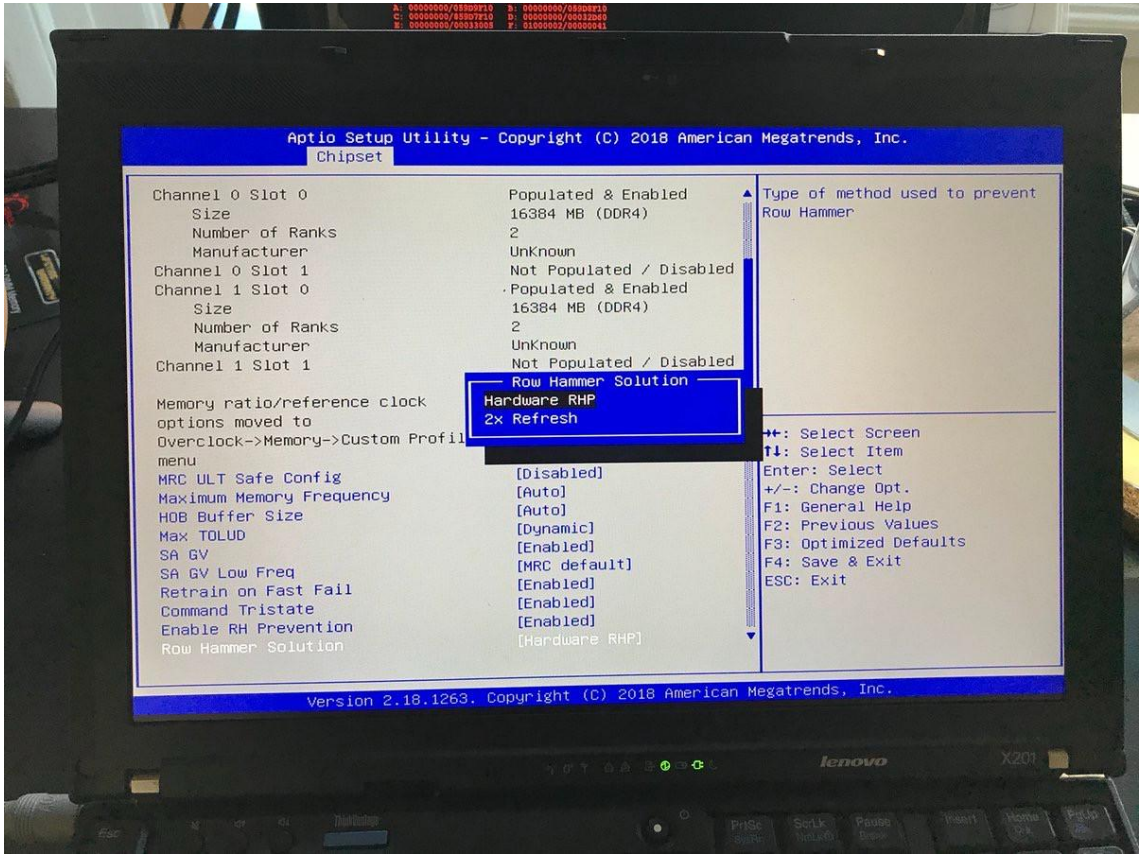
Impact: A malicious application may induce memory corruption to escalate privileges

Description: A disturbance error, also known as Rowhammer, exists with some DDR3 RAM that could have led to memory corruption. This issue was mitigated by increasing memory refresh rates.

CVE-ID

CVE-2015-3693 : Mark Seaborn and Thomas Dullien of Google, working from original research by Yoongu Kim et al (2014)

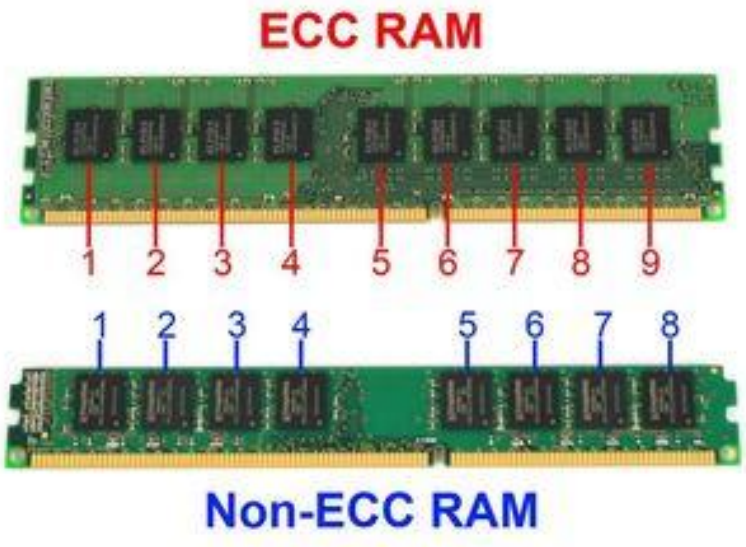
HP, Lenovo, and many other vendors released similar patches



https://twitter.com/isislovecruft/status/1021939922754723841

The obvious defense:

- ECC memory
  - Error-correcting code memory
  - **Basic Idea:** Store extra *redundant* bits to be used in case of a flip!
  - **Naive Implementation:** Store multiple copies and compare
    - DIMM stores extra bits in hamming codes that are used to detect and correct corruptions
    - Exact same memory technology
- Originally designed for reliability
  - Cosmic rays



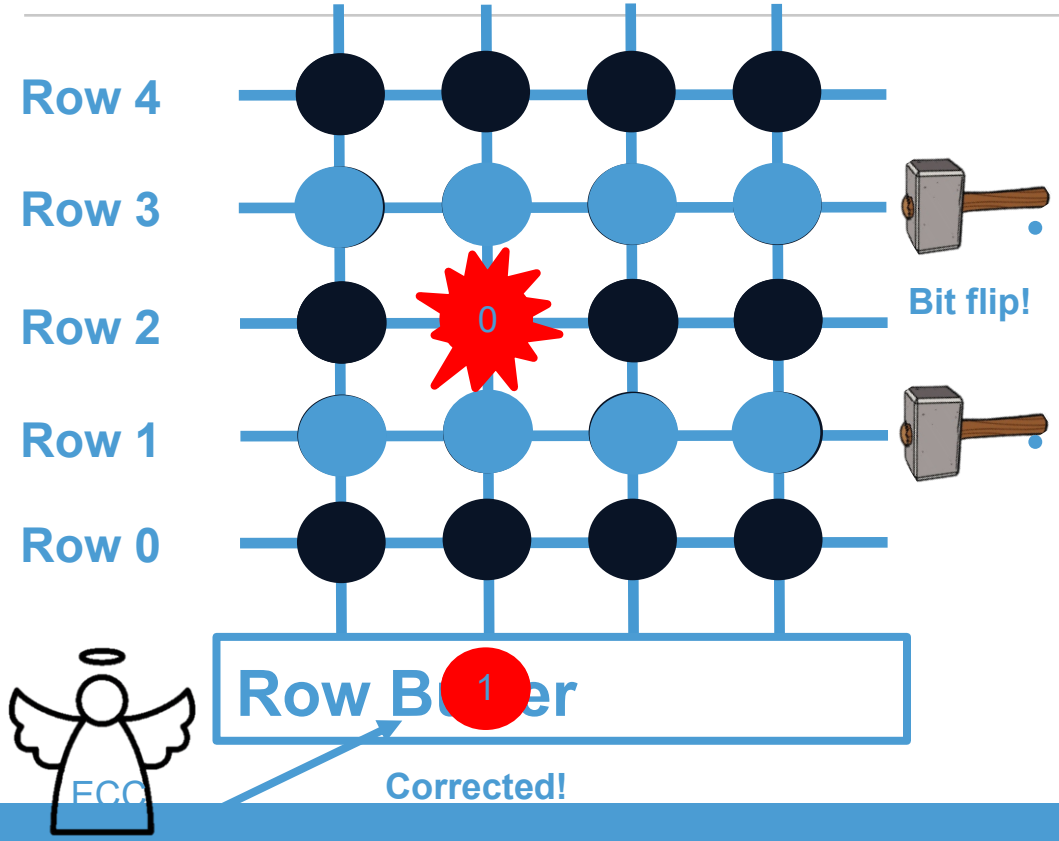
Error Correcting Codes (ECC)

Hamming codes allow for *single-error* correction, double error detection (aka **SECDED**)

How about more than 2-bit flips?

ECC Memory

- Error-Correcting Code Memory:
  - Corrects corrupted data words when when read back

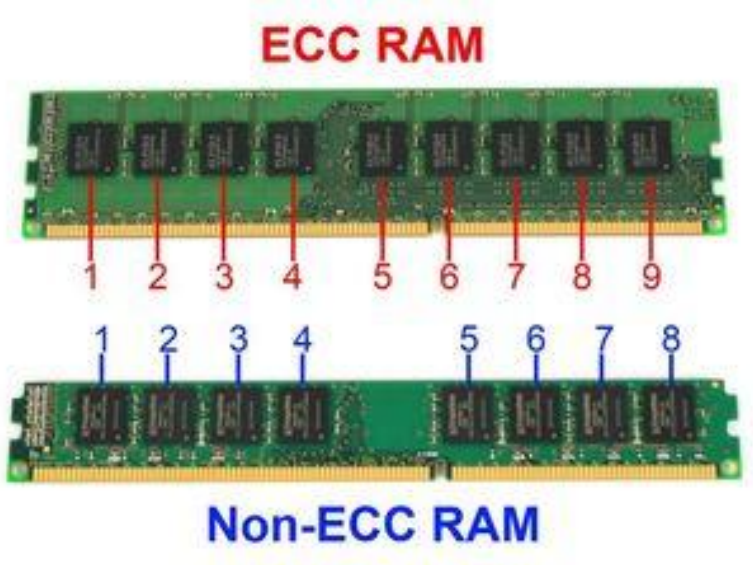


Hamming codes allow for *single-error* correction, double error detection (aka **SECDED**)

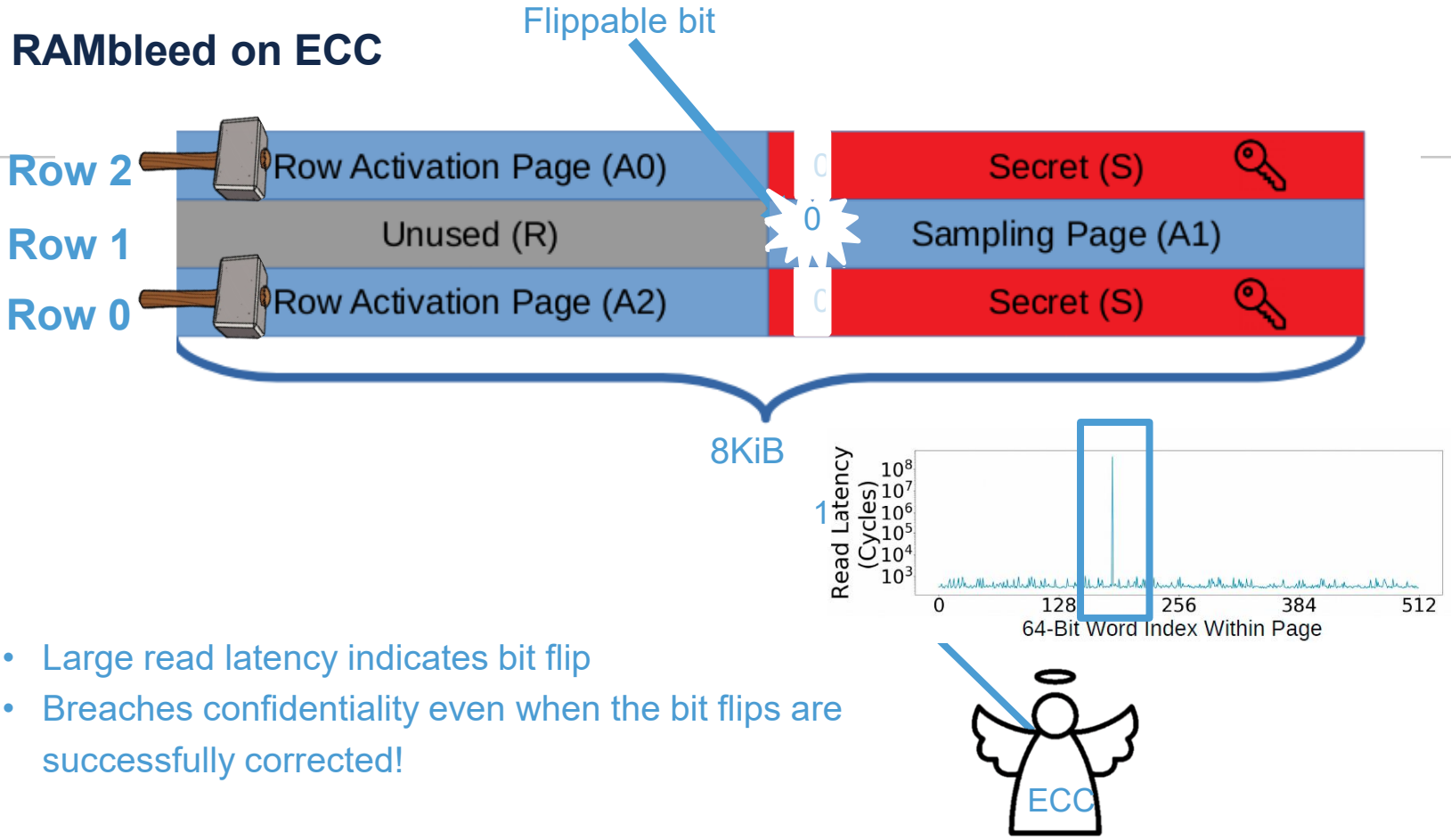
Can't handle more than 3 flips

ECC memory

- Pros
  - Extremely usable
    - Already deployed
- Cons
  - Cost
  - Performance overhead
    - Only 2-3%
- How much security does it actually provide?



RAMbleed on ECC



- Large read latency indicates bit flip
- Breaches confidentiality even when the bit flips are successfully corrected!

ECCploit (Oakland `2019)

- Flipping 3 (or more) bits in one word leads to silent data corruption!

TABLE VI: Percentages of rows with corruptions in an ECC DIMM.

$[\mathcal{P}_1]$	$[\mathcal{P}_2]$	$[\mathcal{P}_3]$	$[\mathcal{P}_4]$
0.12%	0.12%	0.06%	0.60%

TABLE VII: Percentages of rows with corruptions in the flip database of Tatar et al. [37] with 14 DIMMs.

ID	Bit flips	$[\mathcal{P}_1]$	$[\mathcal{P}_2]$	$[\mathcal{P}_3]$	$[\mathcal{P}_4]$
A_1	200468	18.38%	04.41%	00.79%	29.51%
A_2	21542	00.23%	00.03%	00.03%	02.81%
A_3	2926	00.00%	00.00%	00.00%	00.30%
A_4	256359	26.80%	08.52%	02.10%	37.52%
B_1	1504	00.00%	00.00%	00.00%	00.00%
C_1	16489	00.09%	00.00%	00.00%	01.32%
D_1	2131	00.00%	00.00%	00.00%	00.66%
E_1	202630	06.30%	00.76%	00.14%	17.16%
E_2	24587	00.06%	00.00%	00.00%	01.51%
F_1	413796	51.09%	26.02%	06.00%	53.03%
G_1	15990	00.06%	00.00%	00.00%	00.93%
H_1	16087	00.03%	00.00%	00.00%	00.77%
I_1	130187	00.82%	00.03%	00.00%	06.24%
J_1	7185	00.00%	00.00%	00.00%	00.70%
AVG	93705	7.42%	2.84%	0.65%	10.89%


Freeze the victim's DIMM

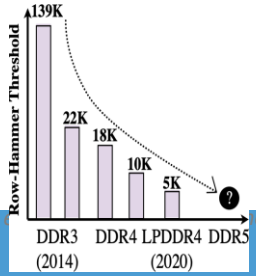
RowHammer Mitigations: A game of cat and mouse

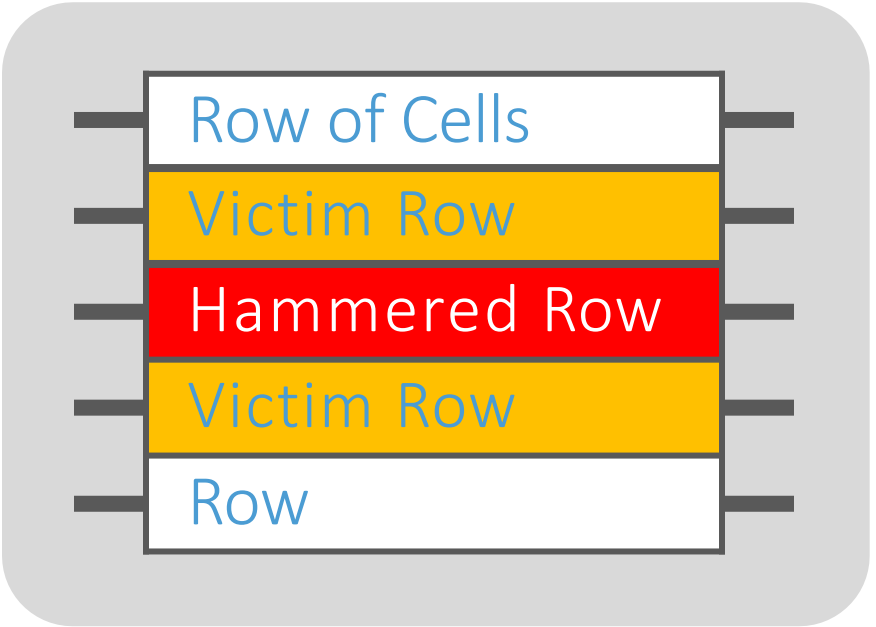
- ECC is insufficient, what's next?

- Need Rowhammer specific mitigations
  - Not just retrofit existing solutions

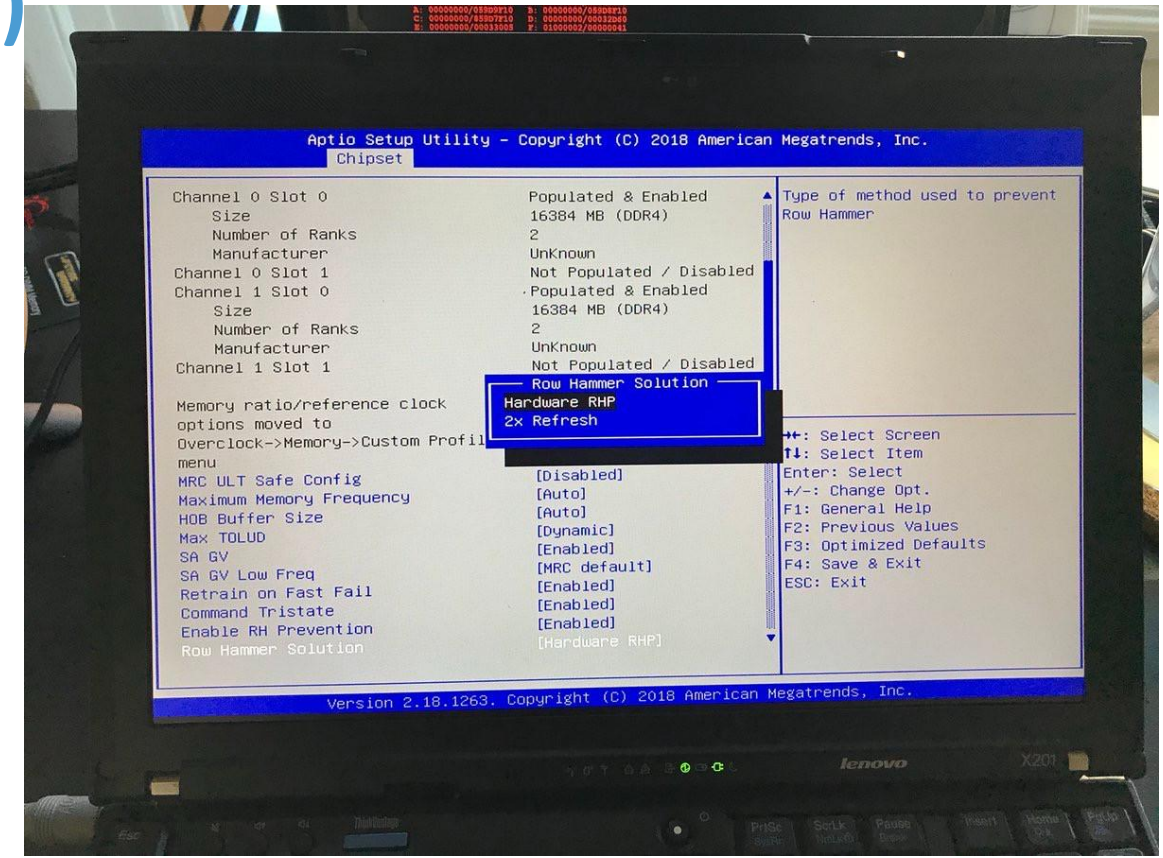
PARA: Probabilistic Row Activation

- Pick a probability "p"
- After closing a row, we activate (i.e., refresh) one of its neighbors with a low probability: $p = 0.005$
- Question: how to pick "p"? What is the consequence?



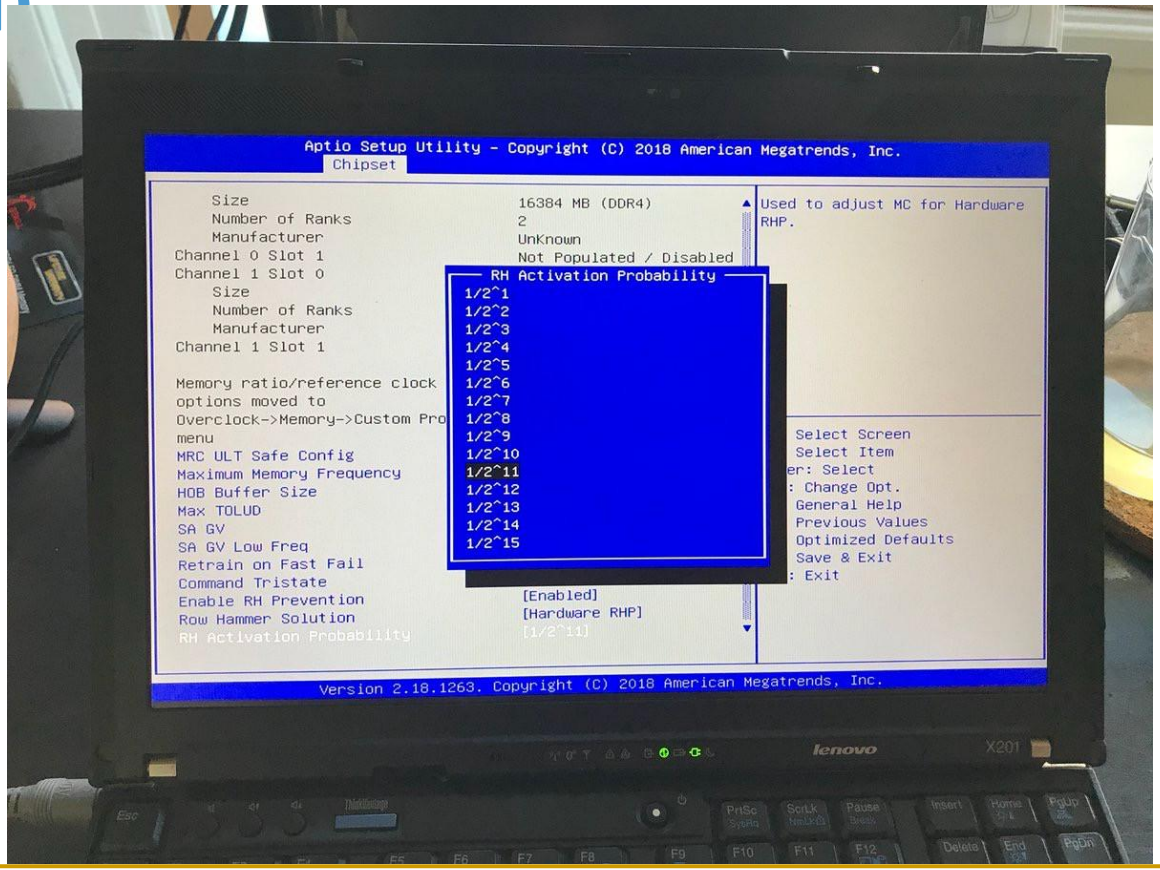


Probabilistic Activation in Real Life (I)



https://twitter.com/isislovecruft/status/1021939922754723841

Probabilistic Activation in Real Life (II)



https://twitter.com/isislovecruft/status/1021939922754723841

Counter-based Row Activation

- Maintain a counter to track the number of accesses per row
  - Increment the counter when accessing a row
  - When reaching the Maximum Activation Count (MAC), activate the neighboring rows
  - After activating, reset the counter

- Deployed in actual hardware (DDR4) in 2016 as Targeted Row Refresh (TRR)