CSI:Rowhammer

Cryptographic Security and Integrity against Rowhammer

Jonas Juffinger, Lukas Lamster, Andreas Kogler, Moritz Lipp, Maria Eichlseder, Daniel Gruss

2023-05-23

IEEE Symposium on Security and Privacy 2023
The **Problem** with Rowhammer Countermeasures
The Problem with Rowhammer Countermeasures

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
The Problem with Rowhammer Countermeasures

- Focusing on characteristics
Focusing on characteristics
Which later turn out to be incomplete
The Problem with Rowhammer Countermeasures

- Focusing on characteristics
- Which later turn out to be incomplete
  - Bit Flips are infrequent - ECC, Refresh Rate
The Problem with Rowhammer Countermeasures

- Focusing on characteristics
- Which later turn out to be incomplete
  - Bit Flips are infrequent—ECC, Refresh Rate [Kim+14]
The Problem with Rowhammer Countermeasures

- Focusing on characteristics
- Which later turn out to be incomplete
  - Bit Flips are infrequent—ECC, Refresh Rate [Kim+14]
  - Detectable with Performance Counters - ANVIL
The Problem with Rowhammer Countermeasures

- Focusing on **characteristics**
- Which later turn out to be **incomplete**
  - Bit Flips are infrequent—ECC, Refresh Rate [Kim+14]
  - Detectable with Performance Counters—ANVIL [Gru+18]
The Problem with Rowhammer Countermeasures

- Focusing on **characteristics**
- Which later turn out to be **incomplete**
  - Bit Flips are infrequent—ECC, Refresh Rate [Kim+14]
  - Detectable with Performance Counters—ANVIL [Gru+18]
  - Hammer Distance is 1 - TRR, ZebRAM, B-CATT
The Problem with Rowhammer Countermeasures

- Focusing on **characteristics**
- Which later turn out to be **incomplete**
  - Bit Flips are infrequent—ECC, Refresh Rate [Kim+14]
  - Detectable with Performance Counters—ANVIL [Gru+18]
  - Hammer Distance is 1—TRR, ZebRAM, B-CATT [Kog+22]
CSI: Rowhammer
Generic approach to data integrity protection
• Generic approach to data integrity protection
- **Generic** approach to data integrity protection
- Detect all data integrity failures with a MAC
• **Generic** approach to data integrity protection
• Detect all data integrity failures with a MAC
• Best effort correction
• **Generic** approach to data integrity protection
• Detect **all** data integrity failures with a MAC
• Best effort correction
• **All** Rowhammer attacks are DoS in the **worst case**
MC
MAC Compute

CPU Core

MAC Compute
SecureMemory

Corruption Exception

Integrity Information

OS

Advanced Correction
e.g. Reload from Disk

Exception Handler

Correction as a Search
No Correct 1 Flip CPU Core

MAC Compute

Secure Memory

Corruption Exception

Integrity Information

OS Advanced Correction e.g. Reload from Disk

Correction as a Search

?
MC Compute

MAC Compute

Secure Memory

Corruption Exception

Integrity Information

OS

Advanced Correction

e.g. Reload from Disk

Exception Handler

Correction as a Search
MC
MAC Compute

Integrity Information

OS
Advanced Correction
e.g. Reload from Disk
Exception Handler
Correction as a Search
MC
MAC Compute

Integrity Information

No

Correct 1 Flip

OS

Advanced Correction
e.g. Reload from Disk
Exception Handler
Correction as a Search
MC

MAC Compute

Integrity Information

No

Correct 1 Flip

Corruption Exception

OS
MC = No Correct 1 Flip

MAC Compute

Integrity Information

Corruption Exception

Exception Handler

OS

Advanced Correction e.g. Reload from Disk

4 J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
MC Compute = No
Correct 1 Flip
Integrity Information

MC

CPU Core

Corruption Exception

OS

Exception Handler
Correction as a Search
MC\hspace{1cm} MAC Compute

CPU Core\hspace{1cm} MAC Compute

OS

Exception Handler

Correction as a Search

Corruption Exception

Correct 1 Flip

No

Integrity Information

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
MC Compute = No
Correct 1 Flip
CPU Core

Integrity Information

MC
MAC Compute

No
Correct Flip

Corruption Exception

Advanced Correction
e.g. Reload from Disk

CPU Core
MAC Compute

Exception Handler
Correction as a Search

OS
Integrity Information

MC

MAC Compute

CPU Core

MAC Compute

Correct 1 Flip

No

Corruption Exception

OS

Advanced Correction

e.g. Reload from Disk

Exception Handler

Correction as a Search
MC
MAC Compute

No
Correct 1 Flip

CPU Core
MAC Compute

Corruption Exception

OS
Advanced Correction
e.g. Reload from Disk

Exception Handler
Correction as a Search
Corruption Exception

Advanced Correction
  e.g. Reload from Disk

Exception Handler
  Correction as a Search

OS

Integrity Information

MC

MAC Compute

CPU Core

Correct 1 Flip

No

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
When a memory corruption exception occurs, the system checks Integrity Information. If the Integrity Information indicates a correct memory value, the corrupted memory location is left uncorrected. If the Integrity Information indicates an incorrect value, the system proceeds to correct the flipped bit. The CPU Core recomputes the MAC for the memory location. If the recompute does not match the original MAC, the process repeats for another flip. If the Integrity Information is not available (No), the Advanced Correction mechanism is activated, possibly by reloading from disk. The Correction as a Search process is then initiated by the Exception Handler.
Integrity Information

MC
MAC Compute

CPU Core
MAC Compute

Correct 1 Flip

Corruption Exception

Advanced Correction
e.g. Reload from Disk

Exception Handler
Correction as a Search

OS
MC

MAC Compute

No

Correct 1 Flip

Integrity Information

CPU Core

MAC Compute

Corruption Exception

OS

Advanced Correction
e.g. Reload from Disk

Exception Handler

Correction as a Search
MC

MAC Compute

Containment

No

Correct 1 Flip

Corruption Exception

OS

Advanced Correction
e.g. Reload from Disk

Exception Handler

Correction as a Search

Integrity Information

CPU Core

MAC Compute

Secure Memory
**CSI: Rowhammer – MAC Design**

- **PMAC construction**
  - **QARMA**
  - Block cipher \([Ava17]\)

- **Physical address as tag**
  - **5.13 ns 256-bit**
  - **6.60 ns 512-bit**

---

**Diagram Details**

- **Phys Addr\(_1\)**
- **\(Q_K\)**
- **\(PA_2\)**
- **\(PA_3\)**
- **\(PA_4\)**
- **\(M_1\), \(M_2\), \(M_3\), \(M_4\)**
- **\(64\)**
- **\(56\)**
- **MAC**
**PMAC construction**
CSI: Rowhammer – MAC Design

- PMAC construction
- QARMA$_5$-64-$\sigma_0$ block cipher [Ava17]
- PMAC construction
- QARMA$_5$-64-$\sigma_0$ block cipher [Ava17]
- Physical address as tag
PMAC construction
QARMA\(_5\)-64-\(\sigma_0\) block cipher [Ava17]
Physical address as tag

5.13 ns 256-bit
• PMAC construction
• QARMA$_5$-64-$\sigma_0$ block cipher [Ava17]
• Physical address as tag

- 5.13 ns 256-bit
- 6.60 ns 512-bit
Data Correction
• MACs cannot correct bit flips
CSI:Rowhammer – Correction as a Search

- MACs cannot correct bit flips
- Brute force search with approximate equality
CSI: Rowhammer – Correction as a Search

- MACs cannot correct bit flips
- Brute force search with approximate equality

0010110100101101 $\xrightarrow{\text{MAC}}$ 01011010

MAC from DRAM $\rightarrow$ 01010010

Parity bits to shrink search space
- MACs cannot correct bit flips
- Brute force search with approximate equality

0010110100101101 \(\rightarrow\) 01011010

MAC from DRAM \(\rightarrow\) 01010010

✓
CSI: Rowhammer – Correction as a Search

- MACs cannot correct bit flips
- Brute force search with approximate equality

\[
\begin{align*}
0010110100101101 & \quad \xrightarrow{\text{MAC}} \quad 01011010 \\
\text{MAC from DRAM} & \quad \xrightarrow{} \quad 01010010 \checkmark
\end{align*}
\]
• MACs cannot correct bit flips
• Brute force search with approximate equality
  \[
  0010110100101101 \xrightarrow{\text{MAC}} 01011010
  \]
  \[
  \text{MAC from DRAM} \rightarrow 01010010 \checkmark
  \]
• Parity bits to shrink search space
CSI: Rowhammer – Correction Time

<table>
<thead>
<tr>
<th>Number of Flips</th>
<th>Correction Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 ns</td>
</tr>
<tr>
<td>2</td>
<td>100 µs</td>
</tr>
<tr>
<td>3</td>
<td>1 s</td>
</tr>
<tr>
<td>4</td>
<td>10 ms</td>
</tr>
<tr>
<td>5</td>
<td>100 µs</td>
</tr>
<tr>
<td>6</td>
<td>1.7 min</td>
</tr>
<tr>
<td>7</td>
<td>1 s</td>
</tr>
<tr>
<td>8</td>
<td>11.6 d</td>
</tr>
</tbody>
</table>

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
OS has some knowledge about the corrupted data
- Reload disk backed data instead of correcting
- Recompute data (page tables)
• OS has some **knowledge** about the corrupted data
OS has some knowledge about the corrupted data

- **Reload** disk backed data instead of correcting
• OS has some **knowledge** about the corrupted data
• **Reload** disk backed data instead of correcting
• **Recompute** data (page tables)
Evaluation
Implemented CSI:Rowhammer in gem5
Modified Linux kernel
Evaluated correct functionality
Evaluated performance overhead
CSI:Rowhammer – Evaluation

- Implemented CSI:Rowhammer in gem5
CSI:Rowhammer – Evaluation

- Implemented CSI:Rowhammer in gem5
- Modified Linux kernel
CSI:Rowhammer – Evaluation

- Implemented CSI:Rowhammer in gem5
- Modified Linux kernel
- Evaluated correct functionality
CSI:Rowhammer – Evaluation

- Implemented CSI:Rowhammer in gem5
- Modified Linux kernel
- Evaluated correct functionality
- Evaluated performance overhead
CSI:Rowhammer – Performance Overhead

- Overhead
- 5.13 ns Delay (DDR5)
- 6.60 ns Delay (DDR4)

- Overhead chart for various applications:
  - blackscholes
  - bodytrack
  - canneal
  - dedup
  - ferret
  - fluidanimate
  - freqmine
  - streamcluster
  - swaptions
  - barnes
  - cholesky
  - fft
  - lu cb
  - lu ncb
  - ocean cp
  - ocean ncp
  - radiosity
  - radix
  - volrend
  - water nsquared
  - water spatial
  - gm parsec
  - gm splash2x
  - geometric mean

- PARSEC
- SPLASH-2x
- GMEAN

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
### Approximate Equality

<table>
<thead>
<tr>
<th>Correction Tries</th>
<th>Flips log(^2)</th>
<th>MAC Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26.0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>31.5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>38.8</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>42.4</td>
<td>0</td>
</tr>
</tbody>
</table>

*J. Juffinger (@notimaginary_) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023*
• Approximate Equality
• Approximate Equality
• Rowhammer can induce bit flips in MAC
• Approximate Equality
• Rowhammer can induce bit flips in MAC
• Decreases MAC strength from initial 56 bit
CSI:Rowhammer – Security Evaluation

- **Approximate Equality**
- Rowhammer can induce bit flips in MAC
- Decreases MAC strength from initial 56 bit

<table>
<thead>
<tr>
<th>Data Flips</th>
<th>$\log_2(# \text{ Correction Tries})$</th>
<th>Ignored Flips</th>
<th>MAC Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26.0</td>
<td>3</td>
<td>41.2</td>
</tr>
<tr>
<td>6</td>
<td>31.5</td>
<td>2</td>
<td>45.4</td>
</tr>
<tr>
<td>7</td>
<td>38.8</td>
<td>1</td>
<td>50.2</td>
</tr>
<tr>
<td>8</td>
<td>42.4</td>
<td>0</td>
<td>56.0</td>
</tr>
</tbody>
</table>
CSI: Rowhammer – Security Evaluation

- Approximate Equality

<table>
<thead>
<tr>
<th>Data Flips</th>
<th>$\log_2(# \text{Correction Tries})$</th>
<th>Ignored Flips</th>
<th>MAC Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26.0</td>
<td>3</td>
<td>41.2</td>
</tr>
<tr>
<td>6</td>
<td>31.5</td>
<td>2</td>
<td>45.4</td>
</tr>
<tr>
<td>7</td>
<td>38.8</td>
<td>1</td>
<td>50.2</td>
</tr>
<tr>
<td>8</td>
<td>42.4</td>
<td>0</td>
<td>56.0</td>
</tr>
</tbody>
</table>
• **Approximate Equality**

• Silent Data Corruption rate less than once per $10^9$ billion years.

<table>
<thead>
<tr>
<th>Data Flips</th>
<th>$\log_2(#$ Correction Tries)</th>
<th>Ignored Flips</th>
<th>MAC Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26.0</td>
<td>3</td>
<td>41.2</td>
</tr>
<tr>
<td>6</td>
<td>31.5</td>
<td>2</td>
<td>45.4</td>
</tr>
<tr>
<td>7</td>
<td>38.8</td>
<td>1</td>
<td>50.2</td>
</tr>
<tr>
<td>8</td>
<td>42.4</td>
<td>0</td>
<td>56.0</td>
</tr>
</tbody>
</table>
• **Approximate Equality**

* Silent Data Corruption rate less than once per $10^9$ billion years.
* Rowhammer second preimage after one year: $9.75 \cdot 10^{-5} \%$

<table>
<thead>
<tr>
<th>Data Flips</th>
<th>$\log_2(# \text{Correction Tries})$</th>
<th>Ignored Flips</th>
<th>MAC Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26.0</td>
<td>3</td>
<td>41.2</td>
</tr>
<tr>
<td>6</td>
<td>31.5</td>
<td>2</td>
<td>45.4</td>
</tr>
<tr>
<td>7</td>
<td>38.8</td>
<td>1</td>
<td>50.2</td>
</tr>
<tr>
<td>8</td>
<td>42.4</td>
<td>0</td>
<td>56.0</td>
</tr>
</tbody>
</table>
• Corruption exception nesting detection
Read the Paper:

- Corruption exception nesting detection
- Virtualization with or without guest support
Read the Paper:

- Corruption exception nesting detection
- Virtualization with or without guest support
- Many more interesting implementation details
Read the Paper:

- Corruption exception nesting detection
- Virtualization with or without guest support
- Many more interesting implementation details
- Detailed security evaluation

J. Juffinger (@notimaginary) — Graz University of Technology — IEEE Symposium on Security and Privacy 2023
CSI: Rowhammer

Cryptographic Security and Integrity against Rowhammer

Jonas Juffinger, Lukas Lamster, Andreas Kogler, Moritz Lipp, Maria Eichlseder, Daniel Gruss

2023-05-23

IEEE Symposium on Security and Privacy 2023

jonas.juffinger@iaik.tugraz.at
@notimaginary_
www.jonasjuffinger.com
PoC: github.com/CSIRowhammer/CSIRowhammerPoC

