TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, Kaveh Razavi

VU Qualcomm ETH Zürich
DRAM
The Rowhammer problem

We have reduced transistor without caring for reliability/security

Rowhammer: affects 87% of deployed DDR3 memory.

Kim et al., “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,” ISCA’14
So what can you do with it?
In the cloud

Razavi et al., “Flip Feng Shui: Hammering a Needle in the Software Stack,” SEC’16
## Rooting Android phones

<table>
<thead>
<tr>
<th>Device</th>
<th>#flips</th>
<th>1st exploitable flip after</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Nexus 5¹</td>
<td>1058</td>
<td>116s</td>
</tr>
<tr>
<td>LG Nexus 5⁴</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>LG Nexus 5⁵</td>
<td>747,013</td>
<td>1s</td>
</tr>
<tr>
<td>LG Nexus 4</td>
<td>1,328</td>
<td>7s</td>
</tr>
<tr>
<td>OnePlus One</td>
<td>3,981</td>
<td>942s</td>
</tr>
<tr>
<td>Motorola Moto G (2013)</td>
<td>429</td>
<td>441s</td>
</tr>
<tr>
<td>LG G4 (ARMv8 - 64-bit)</td>
<td>117,496</td>
<td>5s</td>
</tr>
</tbody>
</table>

22 seconds to root on 18 out of 27 tested phones.

Van der Veen et al., “Drammer: Deterministic Rowhammer Attacks on Mobile Phones,” CCS’16
And over the network...
What about DDR4?

Rowhammer: affects 87% of deployed DDR3 memory.

Drammer: bit flips on Pixel phones with LPDDR4 (2016)
ThirdIO’s Mark Lanteigne reports flips on DDR4 DIMMs (2016)
Gruss et al. report flips on DDR4 (SP’18, 2018)
Recent DDR4 systems

Micron's DDR4 devices automatically perform a type of TRR mode in the background and provide an MPR Page 3 MPR3[3:0] of 1000, indicating there is no restriction to the number of ACTIVATE commands to a given row in a refresh period provided DRAM timing specifications are not violated.

TRR: Keep track of intensely activated rows and refresh their neighbors.
So is Rowhammer a solved problem?

42 recent DIMMs from Samsung, Micron and Hynix (95%+ of market)

(a) Single-sided  (b) Double-sided  (c) One-location

Known hammering patterns.
Results

![Results Graph]

- Single (DDR3)
- Double (DDR4)
- One location

Bit Flips vs. Manufacturing Date

- 2015 - 2019
- Log scale for Bit Flips
So what is TRR and is it really effective?

After a few weeks of reading patents...
Maximum Activation Count

MAW: Maximum Activation Window
MAC: Maximum Activation Count

TRR-compliant DRAM modules advertise these values
Memory controller tunes its mitigation based on them
Detecting MC-based mitigations

Using memory access time $\rightarrow$ Detecting extra refresh commands

Using Rowhammer bit flips $\rightarrow$ Detecting existence of mitigation
With Timing

Xeon E5-2620 v2

MAC:  Untested  Unlimited
With Flips

pTRR: Stops most flips
No mitigation on clients
MC-based mitigations on different platforms

<table>
<thead>
<tr>
<th>CPU</th>
<th>Family</th>
<th>Year</th>
<th>DRAM generation</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Line</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xeon E5-2620 v4</td>
<td>Broadwell</td>
<td>2016</td>
<td>DDR4</td>
<td>REF ×2</td>
</tr>
<tr>
<td>Xeon E5-2620 v2</td>
<td>Ivy Bridge</td>
<td>2013</td>
<td>DDR3</td>
<td>p-TRR</td>
</tr>
<tr>
<td>Xeon E3-1270 v3</td>
<td>Haswell</td>
<td>2013</td>
<td>DDR3</td>
<td>—</td>
</tr>
<tr>
<td><strong>Consumer Line</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core i9-9900K</td>
<td>Coffee Lake R</td>
<td>2018</td>
<td>DDR4</td>
<td>—</td>
</tr>
<tr>
<td>Core i7-8700K</td>
<td>Coffee Lake</td>
<td>2017</td>
<td>DDR4</td>
<td>—</td>
</tr>
<tr>
<td>Core i7-7700K</td>
<td>Kaby Lake</td>
<td>2017</td>
<td>DDR4</td>
<td>—</td>
</tr>
<tr>
<td>Core i7-5775C</td>
<td>Broadwell</td>
<td>2015</td>
<td>DDR3</td>
<td>—</td>
</tr>
</tbody>
</table>
TRR timeline

All our DDR4 DIMMs after ‘16 have MAC set to unlimited
No bit flip with all known Rowhammer patterns
Understanding in-DRAM TRR

Using memory access time \xmark All accesses take the same

Using Rowhammer bit flips \xmark No bit flips

Any TRR solution:

1) Sampling mechanism → Happens at memory access
2) Inhibitor mechanism → Extra refreshes

When do extra internal refreshes happen?
SoftMC

- Open-source platform for DRAM studies
- Support for DDR4
- Precise control over DRAM commands
  - ACTIVATE, READ/WRITE, PRECHARGE, REFRESH
- Run DRAM out of spec
Extra refreshes

On a DIMM from manufacturer C:
No REFRESH command (generally 1 every 7.8us)

1) Write values in memory
2) Hammer for 64 ms (refresh cycle)
3) Check for flips
Results

JEDEC → No retention failure for 64ms
Rowhammer flips → Extra refreshes happen at REFRESH
DDR4 cells are leakier than DDR3 cells
Figuring out the sampler size

Pick N aggressor rows

1) Hammer each for 10K (N x 10K)
2) Send M REFRESH commands
3) Repeat for 10 times
# Corruptions

## Results

Flips are distributed uniformly over the victims
Results

1 TRR per REFRESH command
Results

Remaining flips likely due to aggressor being discarded
Results

Sampler size is 4 given that number of flips plateau after 4 REFs
Flips with native REFRESH rate
Other DIMMs?

Reverse engineered DIMMs from Manufacturer A: very different

Newer DIMMs from Manufacturer C? Different mitigation

Can we automate the analysis to try on different DIMMs?
Meet TRRespass

A Rowhammer fuzzer:

1) Cardinality (# aggressor rows)
2) Aggressor row location

- Executed from the CPU (DRAM mapping reverse engineering)
- Allocate a block of memory and try many random patterns
Successful patterns: many-sided Rowhammer

4-sided

Assisted double-sided
## Results

<table>
<thead>
<tr>
<th>Module</th>
<th>Date (yy-wt)</th>
<th>Freq. (MHz)</th>
<th>Size (GB)</th>
<th>Organization</th>
<th>MAC</th>
<th>Found Patterns</th>
<th>Best Pattern</th>
<th>Corruptions</th>
<th>Double Refresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{0,1,2,3}$</td>
<td>16-37</td>
<td>2132</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$A_4$</td>
<td>16-51</td>
<td>2132</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>4</td>
<td>9-sided</td>
<td>7956 4008 3948</td>
</tr>
<tr>
<td>$A_5$</td>
<td>18-51</td>
<td>2400</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$A_{6,7}$</td>
<td>18-15</td>
<td>2666</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$A_8$</td>
<td>17-09</td>
<td>2400</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>33</td>
<td>19-sided</td>
<td>20808 10289 10519</td>
</tr>
<tr>
<td>$A_9$</td>
<td>17-31</td>
<td>2400</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>33</td>
<td>19-sided</td>
<td>24854 12580 12274</td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>19-02</td>
<td>2400</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>488</td>
<td>10-sided</td>
<td>11342 1809 11533</td>
</tr>
<tr>
<td>$A_{11}$</td>
<td>19-02</td>
<td>2400</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>523</td>
<td>10-sided</td>
<td>12830 1682 11148</td>
</tr>
<tr>
<td>$A_{12,13}$</td>
<td>18-50</td>
<td>2666</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$A_{14}$</td>
<td>19-08$^\dagger$</td>
<td>3200</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>120</td>
<td>14-sided</td>
<td>32723 16490 16233</td>
</tr>
<tr>
<td>$A_{15}$</td>
<td>17-08$^\dagger$</td>
<td>2132</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>2</td>
<td>9-sided</td>
<td>22397 12351 10046</td>
</tr>
<tr>
<td>$B_0$</td>
<td>18-11</td>
<td>2666</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>2</td>
<td>3-sided</td>
<td>17 10 7</td>
</tr>
<tr>
<td>$B_1$</td>
<td>18-11</td>
<td>2666</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>2</td>
<td>3-sided</td>
<td>22 16 6</td>
</tr>
<tr>
<td>$B_2$</td>
<td>18-49</td>
<td>3000</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>2</td>
<td>3-sided</td>
<td>5 2 3</td>
</tr>
<tr>
<td>$B_{3,4}$</td>
<td>19-08$^\dagger$</td>
<td>3000</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$B_{6,7}$</td>
<td>19-08$^\dagger$</td>
<td>2666</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$B_{8,9}$</td>
<td>19-08$^\dagger$</td>
<td>2400</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$B_{10,11}$</td>
<td>16-13$^\dagger$</td>
<td>2132</td>
<td>2</td>
<td>8</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>2</td>
<td>3-sided</td>
<td>12 12</td>
</tr>
<tr>
<td>$C_{0,1}$</td>
<td>18-46</td>
<td>2666</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{2,3}$</td>
<td>19-08$^\dagger$</td>
<td>2800</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{4,5}$</td>
<td>19-08$^\dagger$</td>
<td>3000</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{6,7}$</td>
<td>19-08$^\dagger$</td>
<td>3000</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{8}$</td>
<td>19-08$^\dagger$</td>
<td>3200</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_9$</td>
<td>18-47</td>
<td>2666</td>
<td>16</td>
<td>2</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{10,11}$</td>
<td>19-04</td>
<td>2933</td>
<td>8</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UL</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$C_{12}$</td>
<td>15-01$^\dagger$</td>
<td>2132</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UT</td>
<td>25</td>
<td>10-sided</td>
<td>190037 63904 126133</td>
</tr>
<tr>
<td>$C_{13}$</td>
<td>18-49</td>
<td>2132</td>
<td>4</td>
<td>1</td>
<td>16 $\times$ 8</td>
<td>UT</td>
<td>3</td>
<td>9-sided</td>
<td>694 239 455</td>
</tr>
</tbody>
</table>
TRRespass’ preliminary port to ARM

Limitations:

1) No DRAM mapping functions
2) No access to large pages
3) Detect N bank conflicts? Hammer.

<table>
<thead>
<tr>
<th>Mobile Phone</th>
<th>Year</th>
<th>SoC</th>
<th>Memory (GB)</th>
<th>Found Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Pixel</td>
<td>2016</td>
<td>MSM8996</td>
<td>4†</td>
<td></td>
</tr>
<tr>
<td>Google Pixel 2</td>
<td>2017</td>
<td>MSM8998</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Samsung G960F/DS</td>
<td>2018</td>
<td>Exynos 9810</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Huawei P20 DS</td>
<td>2018</td>
<td>Kirin 970</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Sony XZ3</td>
<td>2018</td>
<td>SDM845</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>HTC U12+</td>
<td>2018</td>
<td>SDM845</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>LG G7 ThinQ</td>
<td>2018</td>
<td>SDM845</td>
<td>4†</td>
<td></td>
</tr>
<tr>
<td>Google Pixel 3</td>
<td>2018</td>
<td>SDM845</td>
<td>4†</td>
<td></td>
</tr>
<tr>
<td>Google Pixel 4</td>
<td>2019</td>
<td>SM8150</td>
<td>6†</td>
<td></td>
</tr>
<tr>
<td>OnePlus 7</td>
<td>2019</td>
<td>SM8150</td>
<td>8†</td>
<td></td>
</tr>
<tr>
<td>Samsung G970F/DS</td>
<td>2019</td>
<td>Exynos 9820</td>
<td>6†</td>
<td></td>
</tr>
<tr>
<td>Huawei P30 DS</td>
<td>2019</td>
<td>Kirin 980</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Xiaomi Redmi Note 8 Pro</td>
<td>2019</td>
<td>Helio G90T</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

† LPDDR4 (not LPDDR4X)
Conclusion

Open PhD Positions @ ETH

Rowhammer is alive and kicking on latest systems

Maintaining data consistency in DRAM has become hard

Security through obscurity is trickier with DRAM

kaveh@ethz.ch @kavehrazavi