Half-Double: Hammering From the Next Row Over

USENIX Security 2022

12th August 2022

Andreas Kogler
Graz University of Technology

Yoongu Kim
Google

Eric Shiu
Rivos

Jonas Juffinger
Graz University of Technology, Lamarr

Moritz Lipp
Amazon Web Services

Mattias Nissler
Google

Salman Qazi
Google

Nicolas Boichat
Google

Daniel Gruss
Graz University of Technology

Motivation

- **Rowhammer**
  - Default refresh window of 64 ms

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
- **Error Correting Code** (ECC)
  - Correct only one flip

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
- **Error Correting Code** (ECC)
  - Correct only one flip
- **Targeted Row Refresh** (TRR)
  - Refresh direct neighbours hammering rows
  - Exhaustion with multi-sided patterns [2, 1]

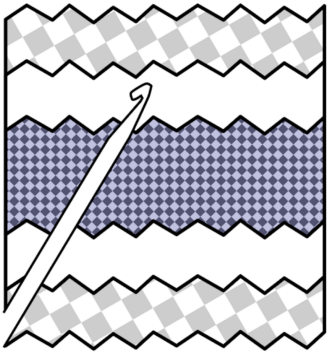
Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
- **Error Correting Code** (ECC)
  - Correct only one flip
- **Targeted Row Refresh** (TRR)
  - Refresh direct neighbours hammering rows
  - Exhaustion with multi-sided patterns [2, 1]
- Would perfect TRR fix Rowhammer attacks?

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

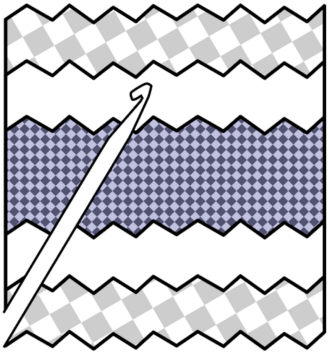
Observed Flips



- Short answer: No

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

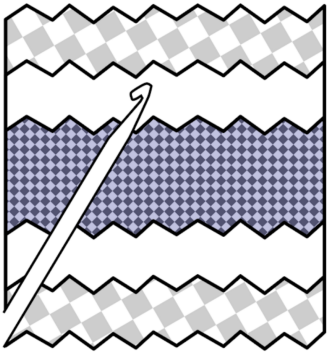
Observed Flips



- Short answer: No
- Hammering with three rows between the aggressors
  - Causes **flips** on LPDDR4x commodity devices
  - 5 out of 7 mobile devices affected
  - With active **TRR** and on-chip **ECC**

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Observed Flips



- Short answer: No
- Hammering with three rows between the aggressors
  - Causes **flips** on LPDDR4x commodity devices
  - 5 out of 7 mobile devices affected
  - With active **TRR** and on-chip **ECC**
- Is this *Distance-2* Rowhammer?
- What is the root cause?

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

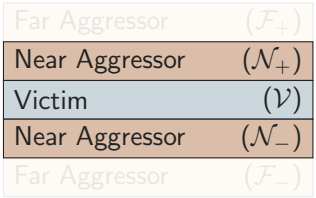
FPGA Experiments

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- FPGA setup
  - Control DIMM via FPGA
  - Full control over the refreshes
  - Deactivated TRR
  - No need for data retention

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

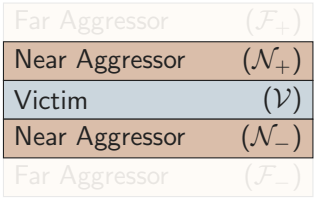
FPGA Experiments - Distance 1



- **Distance-1**
  - $(\mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty$
  - *Classic* double-sided Rowhammer

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

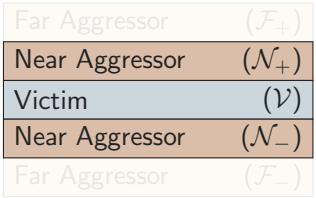
FPGA Experiments - Distance 1



- **Distance-1**
  - $(\mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty$
  - *Classic* double-sided Rowhammer
- **First** flip after:
  - 18 000 hammers in 1.2 *ms*

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

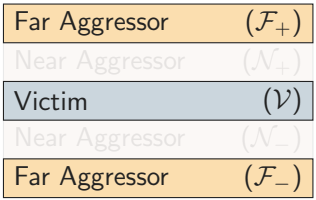
FPGA Experiments - Distance 1



- **Distance-1**
  - $(\mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty$
  - *Classic* double-sided Rowhammer
- **First** flip after:
  - 18 000 hammers in 1.2 *ms*
  - ✓ **Within** the refresh window
  - ✗ **Mitigated** by TRR

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

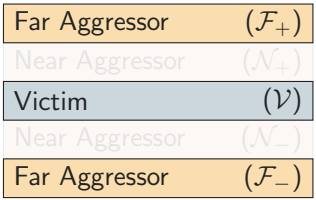
FPGA Experiments - Distance 2



- **Distance-2**
  - $(\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty$
  - Distance two double-sided Rowhammer

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

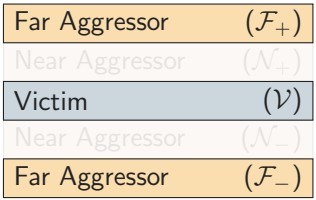
FPGA Experiments - Distance 2



- **Distance-2**
  - $(\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty$
  - Distance two double-sided Rowhammer
- **First** flip after:
  - 4 000 000 hammers in 270 *ms*

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Distance 2



- **Distance-2**
  - $(\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty$
  - Distance two double-sided Rowhammer
- **First** flip after:
  - 4 000 000 hammers in 270 *ms*
  - ✗ **Not** within the refresh windows

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\beta \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty$
  - **Many** distance-2 accesses with a **few** distance-1 accesses

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^{\beta} \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^{\infty}$
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296 960 hammers in 20 *ms*
  - Dilution $\beta = 57$ (5120 distance-1 accesses)

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\beta \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty$
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296 960 hammers in 20 *ms*
  - Dilution $\beta = 57$ (5120 distance-1 accesses)
  - ✓ **Within** the refresh window

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^{\beta} \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^{\infty}$
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296 960 hammers in 20 *ms*
  - Dilution $\beta = 57$ (5120 distance-1 accesses)
  - ✓ **Within** the refresh window
  - ✓ **Assisted** by TRR

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^{\beta} \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^{\infty}$
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296 960 hammers in 20 *ms*
  - Dilution $\beta = 57$ (5120 distance-1 accesses)
  - ✓ **Within** the refresh window
  - ✓ **Assisted** by TRR
- Attacker $\rightarrow \mathcal{F}$

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

FPGA Experiments - Half-Double

Far Aggressor	$(\mathcal{F}_+)$
Near Aggressor	$(\mathcal{N}_+)$
Victim	$(\mathcal{V})$
Near Aggressor	$(\mathcal{N}_-)$
Far Aggressor	$(\mathcal{F}_-)$

- **Half-Double**
  - $((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^{\beta} \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^{\infty}$
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296 960 hammers in 20 *ms*
  - Dilution $\beta = 57$ (5120 distance-1 accesses)
  - ✓ **Within** the refresh window
  - ✓ **Assisted** by TRR
- Attacker $\rightarrow \mathcal{F}$
- TRR $\rightarrow \mathcal{N}$

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Exploitable in the Wild?

End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]

End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- **C1**: Allocation of Contiguous Memory

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- **C1**: Allocation of Contiguous Memory
- **C2**: Alternative to Memory Templating

End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- **C1**: Allocation of Contiguous Memory
- **C2**: Alternative to Memory Templating
- **C3**: Memory Massaging

End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- **C1**: Allocation of Contiguous Memory
- **C2**: Alternative to Memory Templating
- **C3**: Memory Massaging
- **C4**: Bit-Flip Verification

- Mapping from virtual to physical addresses

$$\mathbf{X}_0 = b_8$$
$$\mathbf{X}_1 = b_{12} \oplus b_{16}$$
$$\mathbf{X}_2 = b_{13} \oplus b_{17}$$
$$\mathbf{X}_3 = b_{14} \oplus b_{18}$$

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

$$\mathbf{X}_0 = b_8$$
$$\mathbf{X}_1 = b_{12} \oplus b_{16}$$
$$\mathbf{X}_2 = b_{13} \oplus b_{17}$$
$$\mathbf{X}_3 = b_{14} \oplus b_{18}$$

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

$$\mathbf{X}_0 = b_8$$
$$\mathbf{X}_1 = b_{12} \oplus b_{16}$$
$$\mathbf{X}_2 = b_{13} \oplus b_{17}$$
$$\mathbf{X}_3 = b_{14} \oplus b_{18}$$

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks
- **Specific** *bank* access pattern if contiguous memory

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

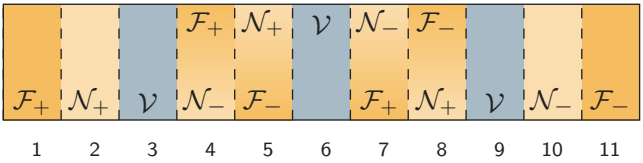
$$\mathbf{X}_0 = b_8$$
$$\mathbf{X}_1 = b_{12} \oplus b_{16}$$
$$\mathbf{X}_2 = b_{13} \oplus b_{17}$$
$$\mathbf{X}_3 = b_{14} \oplus b_{18}$$

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks
- **Specific** *bank* access pattern if contiguous memory

✓ Extract pattern with a timing side channel

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

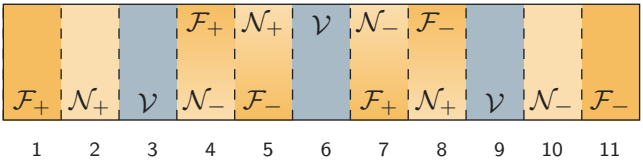
C2 & C3 - Memory Templating & Memory Massaging



- **Skip** templating

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

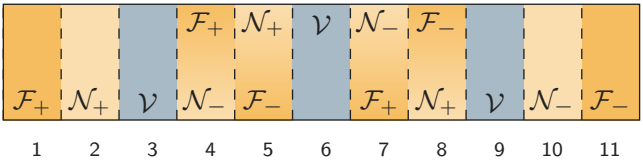
C2 & C3 - Memory Templating & Memory Massaging



- **Skip** templating
- Spray page tables

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

C2 & C3 - Memory Templating & Memory Massaging



- **Skip** templating
- Spray page tables
- Hammer with Half-Double

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

C4 - Bit-Flip Verification

- Corrupt page table entries can kill the attacker process

C4 - Bit-Flip Verification

- Corrupt page table entries can kill the attacker process

```
if ( /*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if ( is_cached(probe) ) {
    // ptr[0-4] valid
}
```

- **Verify** if address save to access
- **Spectre** gadget

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

C4 - Bit-Flip Verification

- Corrupt page table entries can kill the attacker process

```
if ( /*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if ( is_cached(probe) ) {
    // ptr[0-4] valid
}
```

- **Verify** if address save to access
- **Spectre** gadget
- Cached → accessible

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

C4 - Bit-Flip Verification

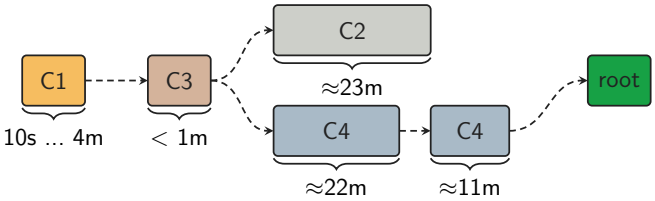
- Corrupt page table entries can kill the attacker process

```
if ( /*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if ( is_cached(probe) ) {
    // ptr[0-4] valid
}
```

- **Verify** if address save to access
- **Spectre** gadget
- Cached → accessible
- **Suppresses** corruption faults

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

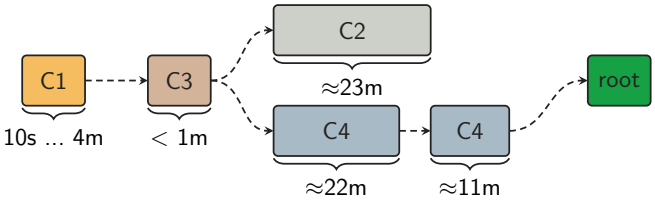
End-to-End Exploit - Timings



- **45** minutes (*Chromebook*$_2$)

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

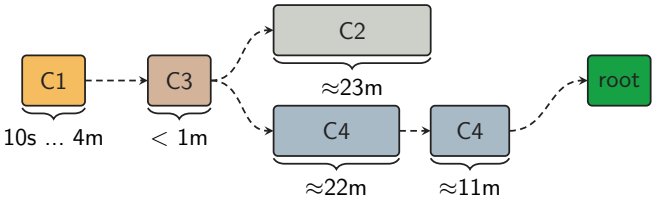
End-to-End Exploit - Timings



- **45** minutes (*Chromebook*$_2$)
- Full memory read & write primitive

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

End-to-End Exploit - Timings



- **45** minutes (*Chromebook*$_2$)
- Full memory read & write primitive
- Deployable inside an APP

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Final Remarks

- **Open Source** https://github.com/IAIK/halfdouble

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Final Remarks

- **Open Source** `https://github.com/IAIK/halfdouble`
- **Passed** artifact evaluation

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Final Remarks

- **Open Source** `https://github.com/IAIK/halfdouble`
- **Passed** artifact evaluation

- **More** details
  - *Dance*-experiments
  - Contiguous memory Z3 solver
  - Physical address bit recovery
  - ...

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Final Remarks

- **Open Source** `https://github.com/IAIK/halfdouble`
- **Passed** artifact evaluation

- **More** details
  - *Dance*-experiments
  - Contiguous m… …s solver
  - Physical a… …bit recovery
  - …

Read the Paper

References i

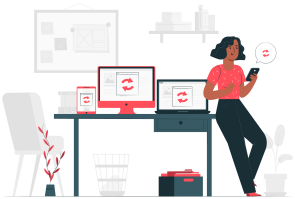

Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. TRRespass: Exploiting the Many Sides of Target Row Refresh. In: S&P. 2020.


Finn de Ridder, Pietro Frigo, Emanuele Vannacci, Herbert Bos, Cristiano Giuffrida, and Kaveh Razavi. SMASH: Synchronized Many-sided Rowhammer Attacks From JavaScript. In: USENIX Security Symposium. 2021.


Mark Seaborn and Thomas Dullien. Test DRAM for bit flips caused by the rowhammer problem. Retrieved on July 27, 2015. 2015. URL: https://github.com/google/rowhammer-test.

Additonal Slides

Affected Devices



- Tested **13** DIMMs & devices
- **2** DIMMs affected
  - FPGA analysis
  - Exact numbers
- **5** out of **7** mobile devices affected
  - Reversed addressing
  - Unprivileged flush
  - Uncachable memory ($10x$)

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)

Affected Devices - Flip Numbers

System	RAM	$N_{Hammers}$	$UC_{0\to1}$	$UC_{1\to0}$	$Flush_{0\to1}$	$Flush_{1\to0}$
$Chromebook_1$	LPDDR4x	23 274	27	40	2	5
$Chromebook_2$	LPDDR4x	23 586	235	2379	12	101
OnePlus 5T	LPDDR4x	25 687	2	30	1	24
Pixel 3	LPDDR4x	32 921	11	5	0	0
HTC U11	LPDDR4x	21 840	-	-	3	17

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)