# Marionette: A RowHammer Attack via Row Coupling

Seungmin Baek, Minbok Wi, Seonyong Park, Hwayong Nam, Michael Jaemin Kim, Nam Sung Kim, Jung Ho Ahn

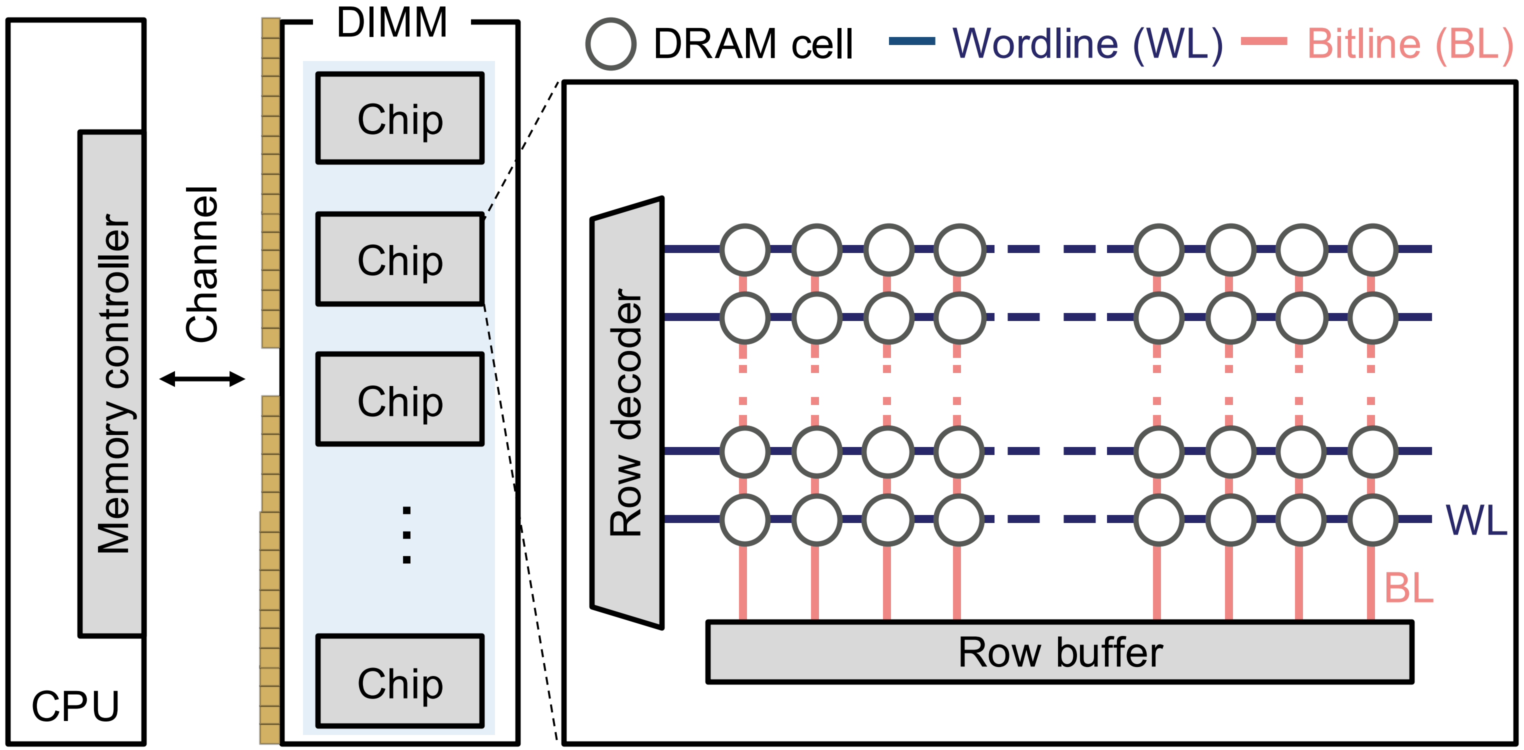
Seoul National University and University of Illinois Urbana Champaign

Presenter: **Seungmin Baek** (qortmdalss@snu.ac.kr)

# Coupled Row

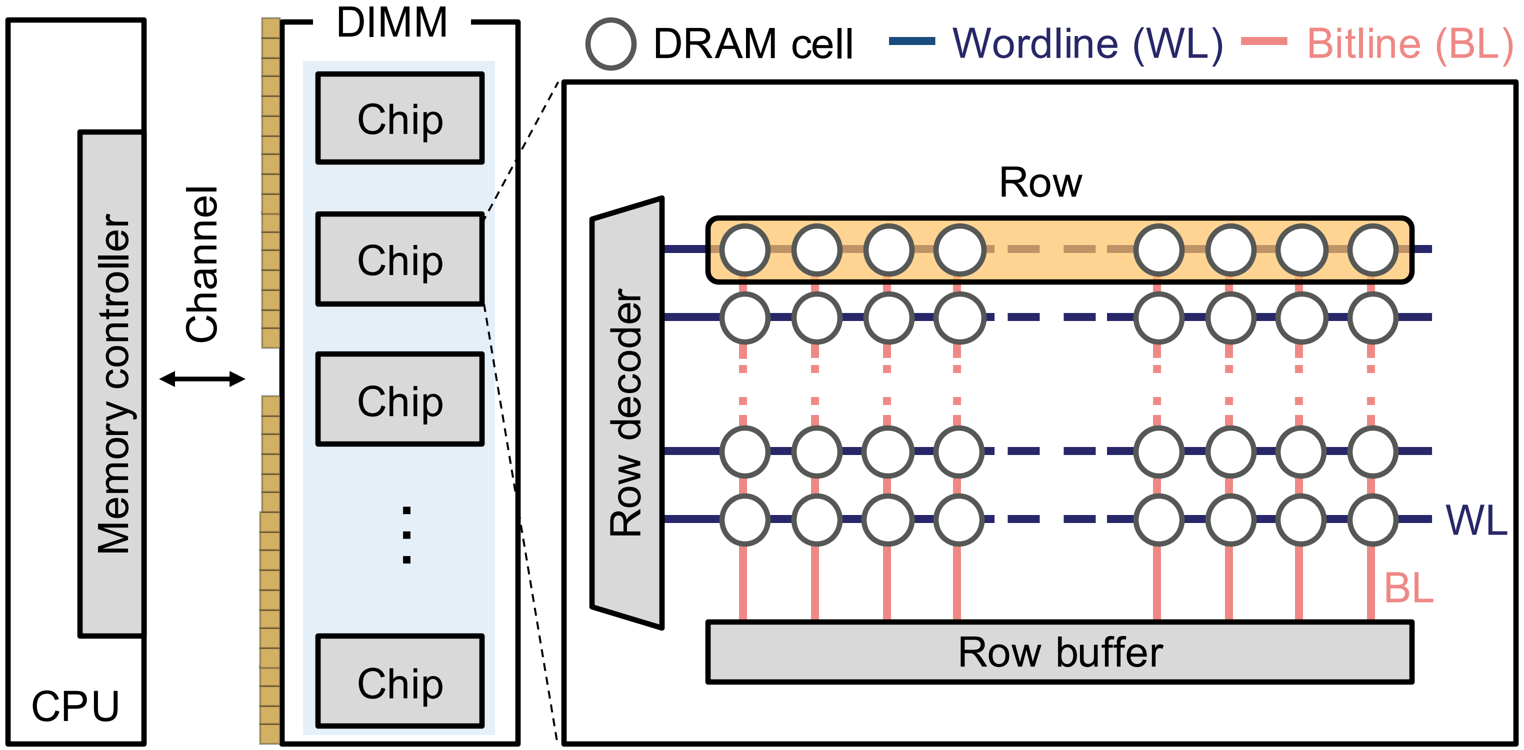
# General DRAM Organization

- A DRAM row consists of multiple DRAM cells connected to the same wordline.
- A row decoder selects and activates a specific WL to access the row.



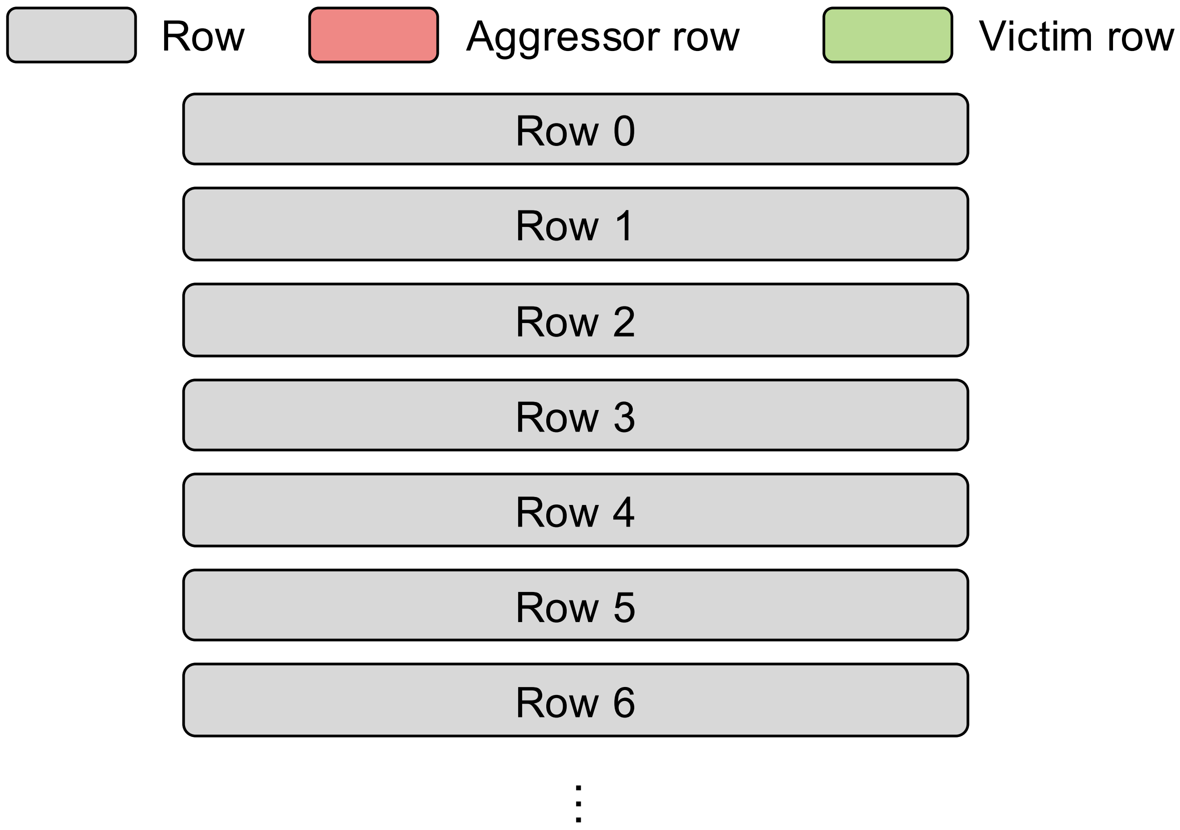
# General DRAM Organization

- A DRAM row consists of multiple DRAM cells connected to the same wordline.
- A row decoder selects and activates a specific WL to access the row.



# RowHammer

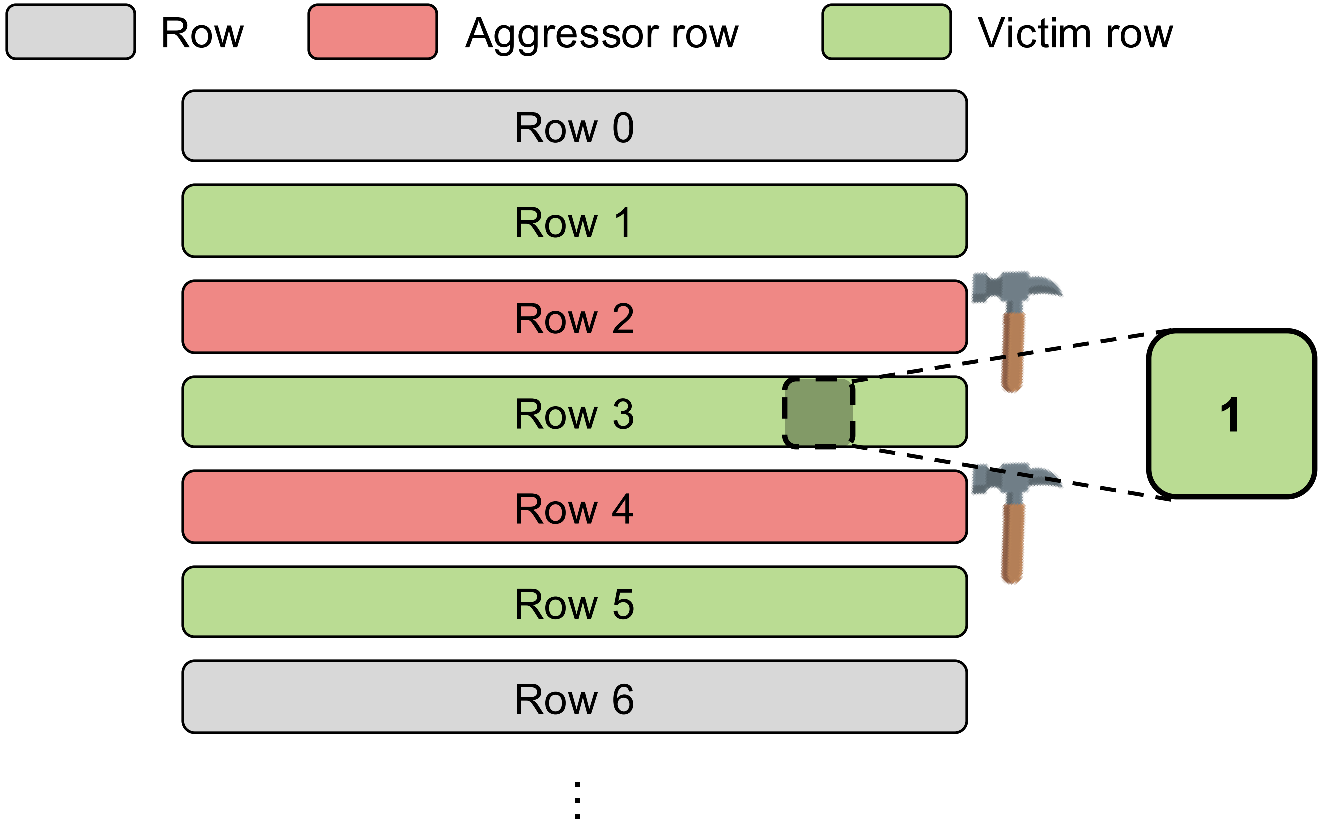
- Repeatedly accessing a specific (aggressor) row results in bitflips in its adjacent (victim) rows[1]



[1] Y. Kim et al., “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA, 2014.

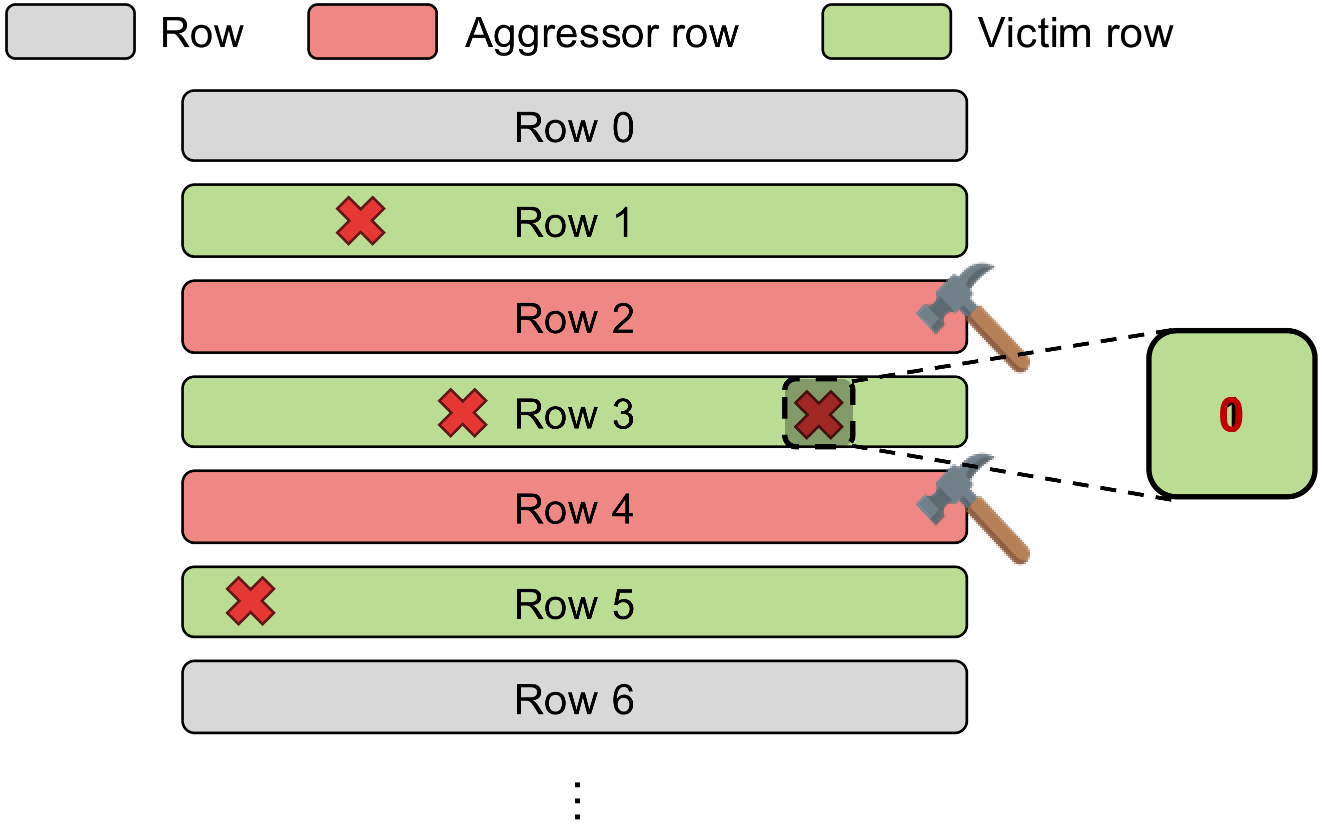
# RowHammer

- Repeatedly accessing a specific (aggressor) row results in bitflips in its adjacent (victim) rows



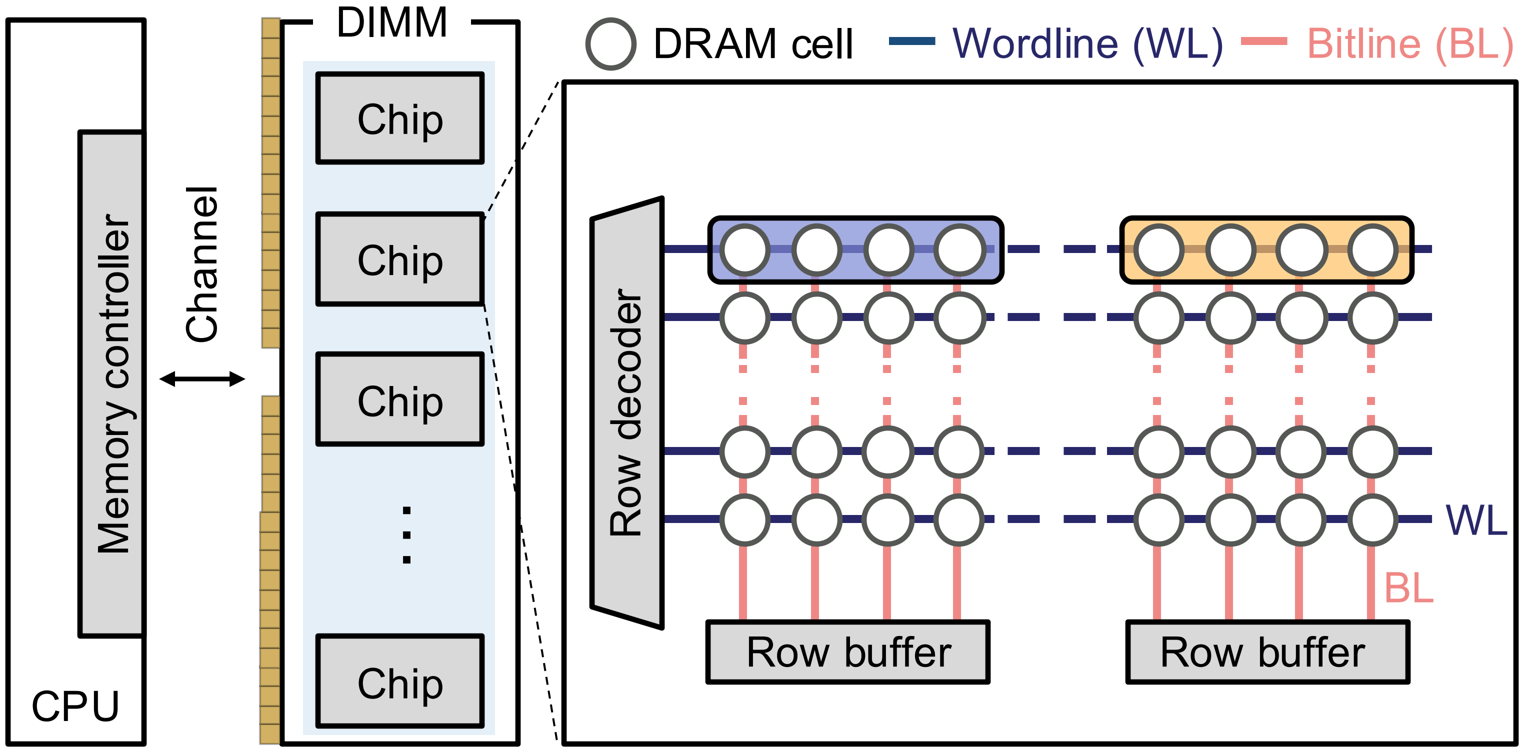
# RowHammer

- Repeatedly accessing a specific (aggressor) row results in bitflips in its adjacent (victim) rows



# Coupled Row

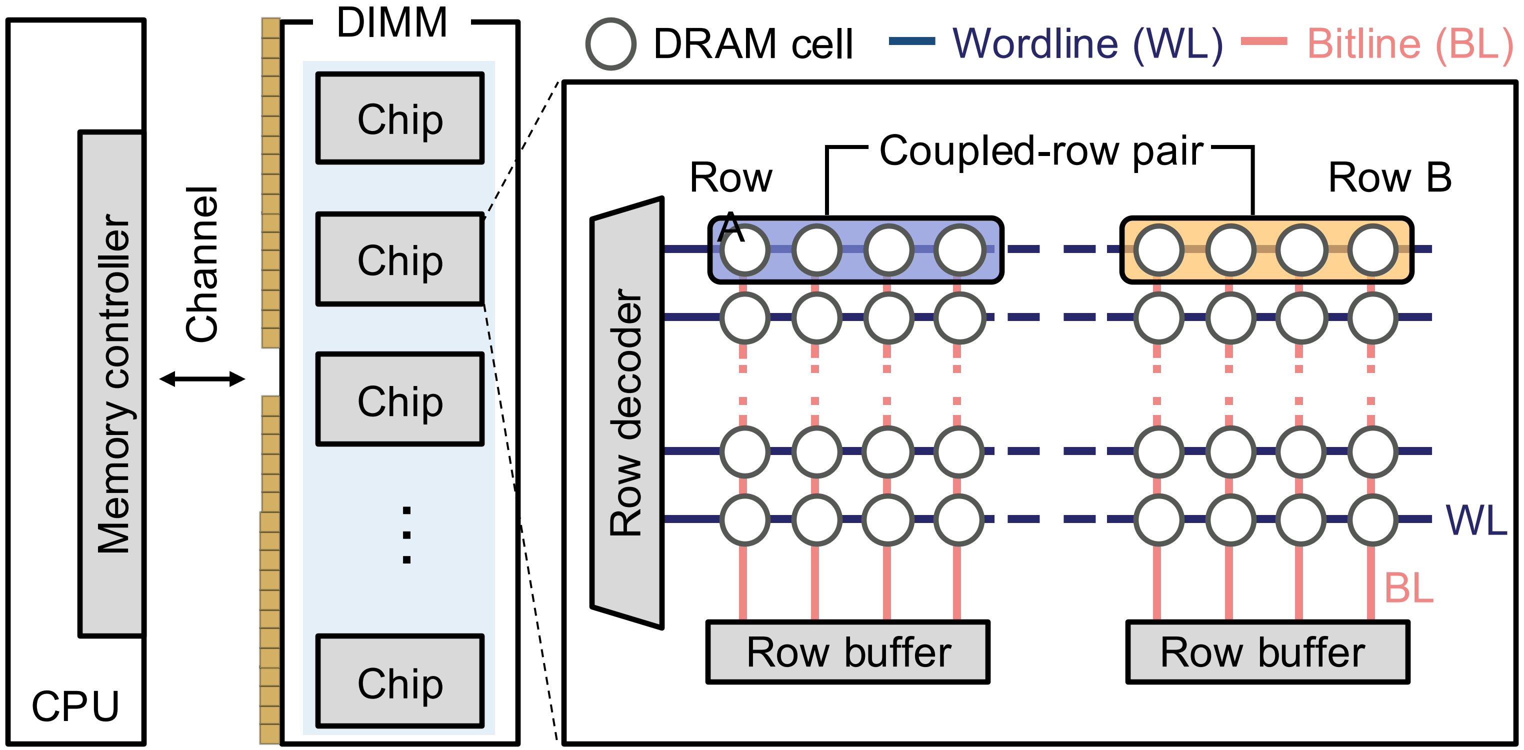
- In certain DRAM chips, **two different rows share the same WL** from the memory controller’s perspective[2-4].



[2] H. Nam et al., “X-ray: Discovering DRAM Internal Structure and Error Characteristics by Issuing Memory Commands," CAL, 2023.
[3] H. Nam et al., “DRAMScope: Uncovering DRAM Microarchitecture and Characteristics by Issuing Memory Commands," ISCA, 2024.
[4] J. S. Kim et al., “Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques," ISCA, 2020.

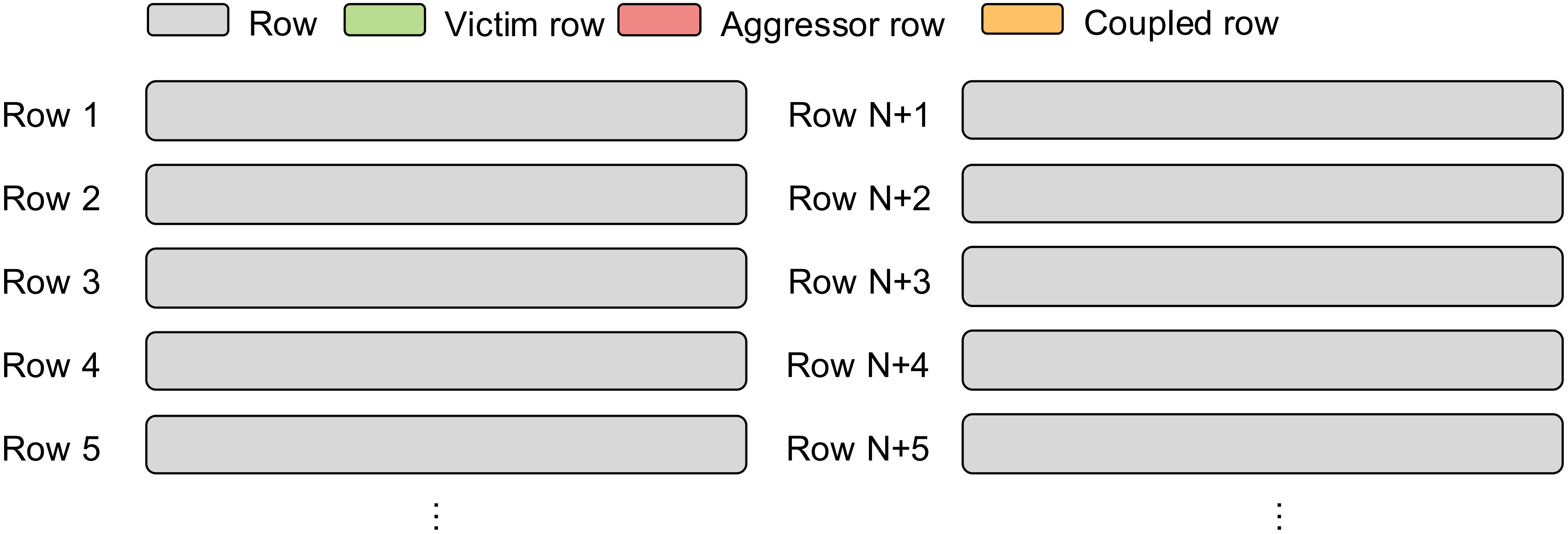
# Coupled Row

- **Row A** and **Row B** form a **coupled-row pair,** where each serves as the **coupled row** of the other.



# Coupled Row

- RowHammer bitflips occur not only in adjacent rows but also in their coupled rows.



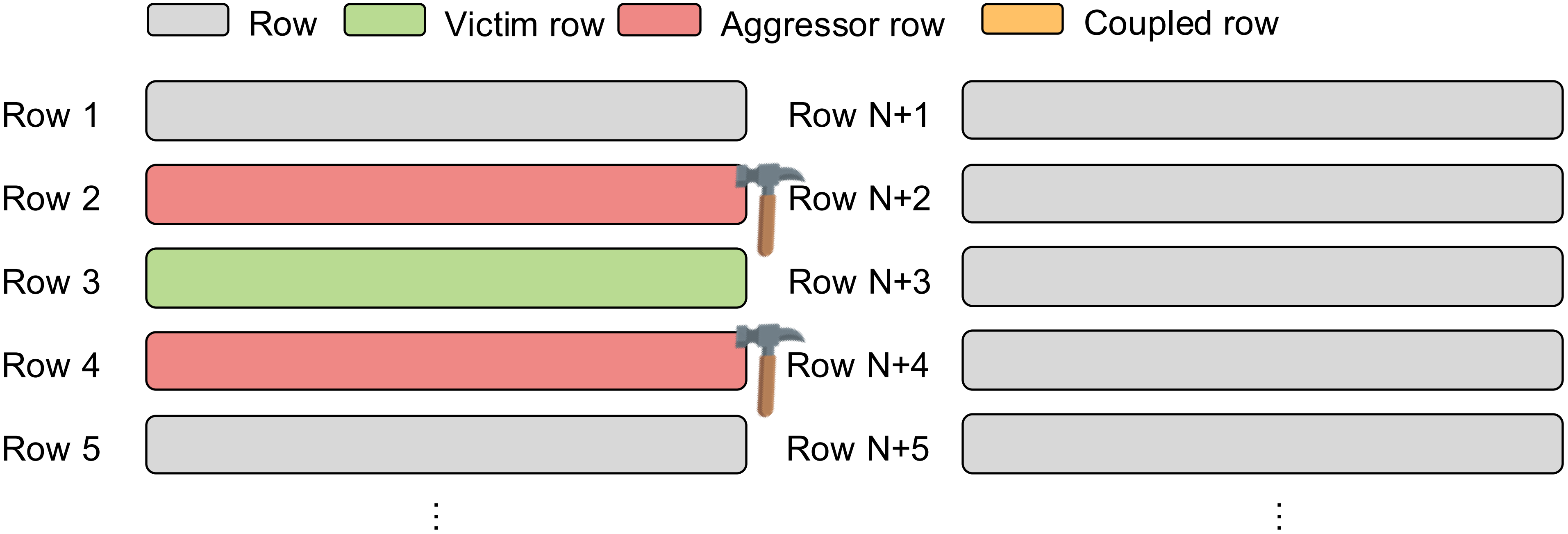
⬜ Row 🟩 Victim row 🟥 Aggressor row 🟧 Coupled row

Row 1
Row 2
Row 3
Row 4
Row 5
⋮

Row N+1
Row N+2
Row N+3
Row N+4
Row N+5
⋮

# Coupled Row

- RowHammer bitflips occur not only in adjacent rows but also in their coupled rows.



⬜ Row 🟩 Victim row 🟥 Aggressor row 🟧 Coupled row

Row 1

Row 2

Row 3

Row 4

Row 5

Row N+1

Row N+2

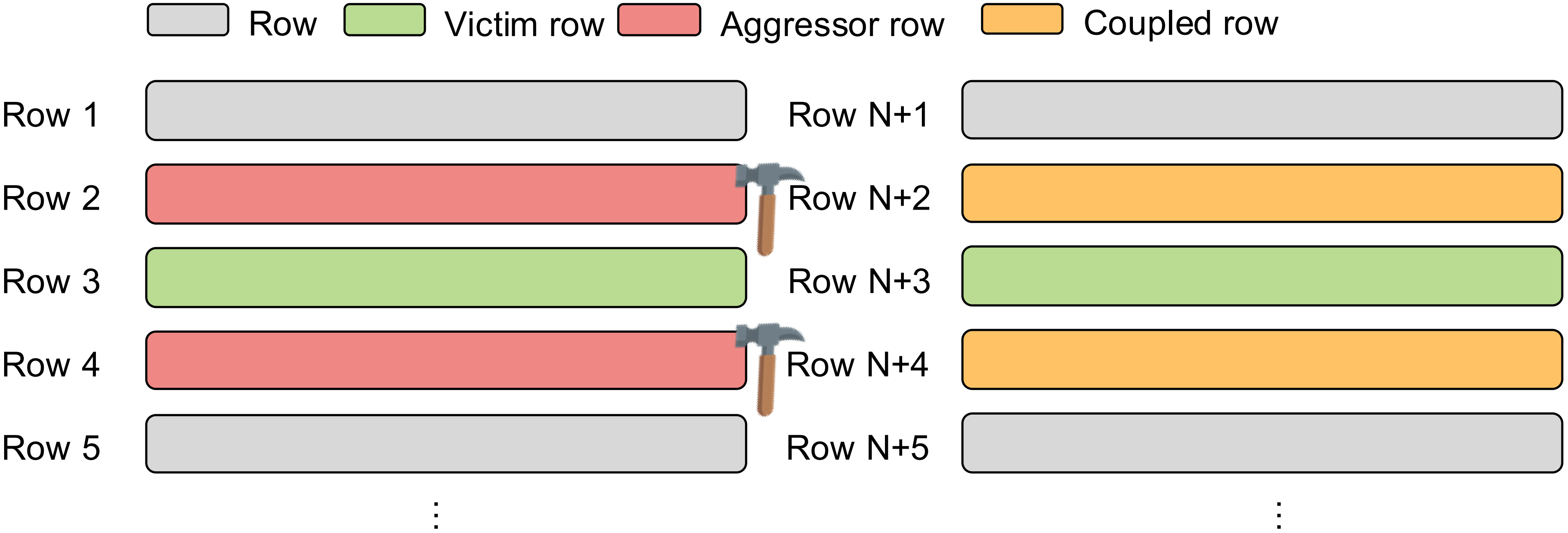
Row N+3

Row N+4

Row N+5

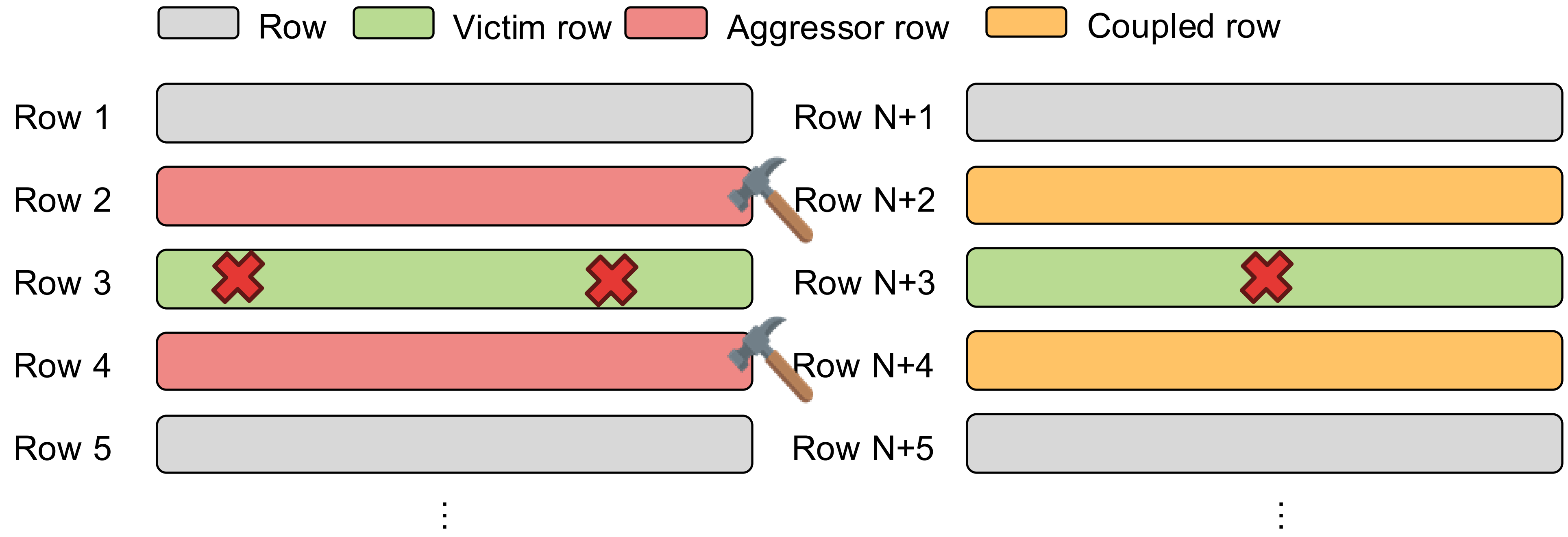
# Coupled Row

- RowHammer bitflips occur not only in adjacent rows but also in their coupled rows.



# Coupled Row

- RowHammer bitflips occur not only in adjacent rows but also in their coupled rows.



# Coupled Row

- Experimental setup
  - Intel Xeon CPU
    - System-a: Xeon E5-2620 v3 (Haswell) / ECC-disabled
    - System-b: Xeon E5-2680 v4 (Broadwell) **/** ECC-enabled
    - System-c: Xeon E5-2698 v4 (Broadwell) **/** ECC-enabled
    - System-d: Xeon Gold 6234 (Cascade Lake) **/** ECC-enabled

# Coupled Row

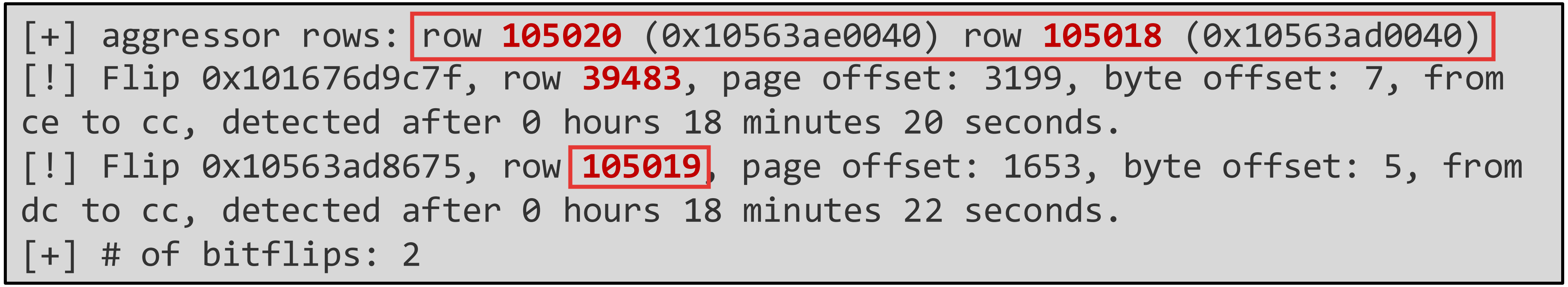
- System experiments
  - Blacksmith RowHammer fuzzer[5]

```
[+] aggressor rows: row 105020 (0x10563ae0040) row 105018 (0x10563ad0040)
[!] Flip 0x101676d9c7f, row 39483, page offset: 3199, byte offset: 7, from
ce to cc, detected after 0 hours 18 minutes 20 seconds.
[!] Flip 0x10563ad8675, row 105019, page offset: 1653, byte offset: 5, from
dc to cc, detected after 0 hours 18 minutes 22 seconds.
[+] # of bitflips: 2
```

[5] P. Jattke et al., “Blacksmith: Scalable rowhammering in the frequency domain," S&P, 2022.

# Coupled Row

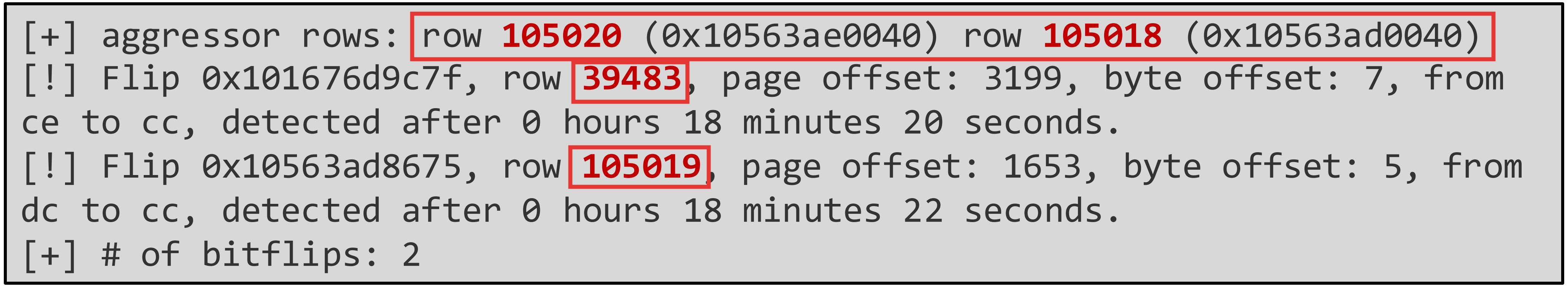
- System experiments
  - Blacksmith RowHammer fuzzer
  - Aggressor rows: **105020**, **105018**
  - Victim rows: **105019**



```
[+] aggressor rows: row 105020 (0x10563ae0040) row 105018 (0x10563ad0040)
[!] Flip 0x101676d9c7f, row 39483, page offset: 3199, byte offset: 7, from
ce to cc, detected after 0 hours 18 minutes 20 seconds.
[!] Flip 0x10563ad8675, row 105019, page offset: 1653, byte offset: 5, from
dc to cc, detected after 0 hours 18 minutes 22 seconds.
[+] # of bitflips: 2
```

# Coupled Row

- System experiments
  - Blacksmith RowHammer fuzzer
  - Aggressor rows: **105020**, **105018**
  - Victim rows: **105019**, **39483 (105019 - $2^{16}$)**



```
[+] aggressor rows: row 105020 (0x10563ae0040) row 105018 (0x10563ad0040)
[!] Flip 0x101676d9c7f, row 39483, page offset: 3199, byte offset: 7, from
ce to cc, detected after 0 hours 18 minutes 20 seconds.
[!] Flip 0x10563ad8675, row 105019, page offset: 1653, byte offset: 5, from
dc to cc, detected after 0 hours 18 minutes 22 seconds.
[+] # of bitflips: 2
```

# Coupled Row

- System experiments
  - – Linux kernel module
    - = Error Detection And Correction (EDAC)

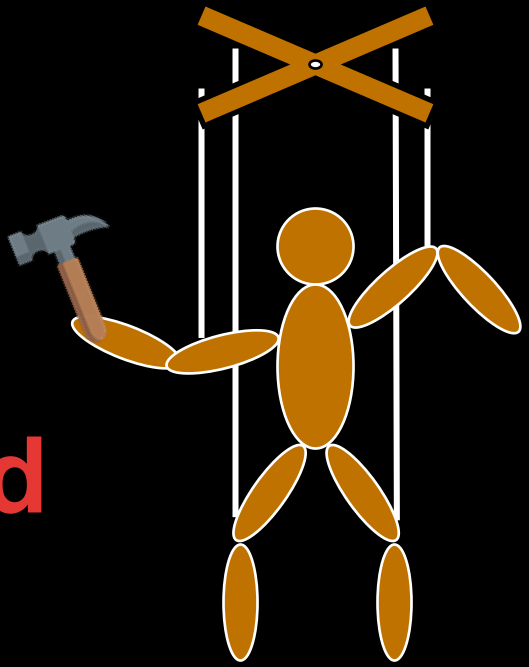
```
kernel: [35218.420544] EDAC MC0: 1 CE memory read error on
CPU_SrcID#0_Ha#0_Chan#0_DIMM#0 (channel:0 slot:0 page:0x67050d offset:0xf00
grain:32 syndrome:0x0 -  area:DRAM err_code:0001:0090 socket:0 ha:0
channel_mask:1 rank:0 row:0x18831 col:0x3f0 bank_addr:2 bank_group:2)
kernel: [35220.409008] EDAC MC0: 1 CE memory read error on
CPU_SrcID#0_Ha#0_Chan#0_DIMM#0 (channel:0 slot:0 page:0x27050d offset:0xb80
grain:32 syndrome:0x0 -  area:DRAM err_code:0001:0090 socket:0 ha:0
channel_mask:1 rank:0 row:0x8831 col:0x3b8 bank_addr:2 bank_group:2)
```

# Coupled Row

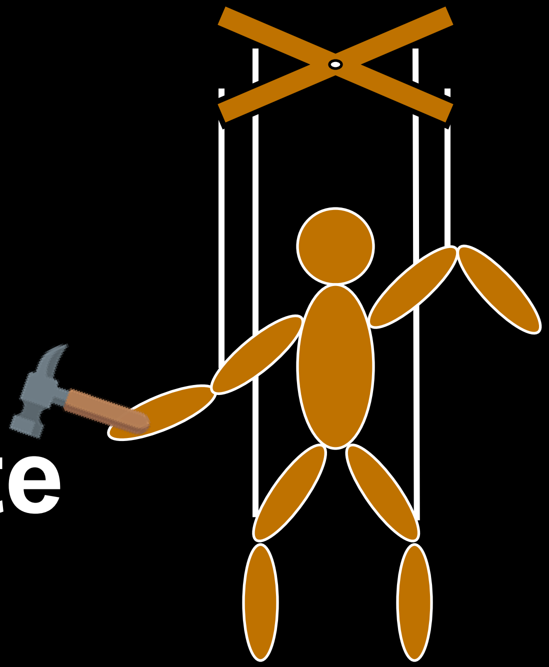
- System experiments
  - Linux kernel module
    - = Error Detection And Correction (EDAC)
  - Aggressor rows: **0x18830**, **0x18832**
  - Victim rows: **0x18831**, **0x8831 (0x18831 - 0x10000)**

```
kernel: [35218.420544] EDAC MC0: 1 CE memory read error on
CPU_SrcID#0_Ha#0_Chan#0_DIMM#0 (channel:0 slot:0 page:0x67050d offset:0xf00
grain:32 syndrome:0x0 -  area:DRAM err_code:0001:0090 socket:0 ha:0
channel_mask:1 rank:0 row:0x18831 col:0x3f0 bank_addr:2 bank_group:2)
kernel: [35220.409008] EDAC MC0: 1 CE memory read error on
CPU_SrcID#0_Ha#0_Chan#0_DIMM#0 (channel:0 slot:0 page:0x27050d offset:0xb80
grain:32 syndrome:0x0 -  area:DRAM err_code:0001:0090 socket:0 ha:0
channel_mask:1 rank:0 row:0x8831 col:0x3b8 bank_addr:2 bank_group:2)
```

# Marionettd

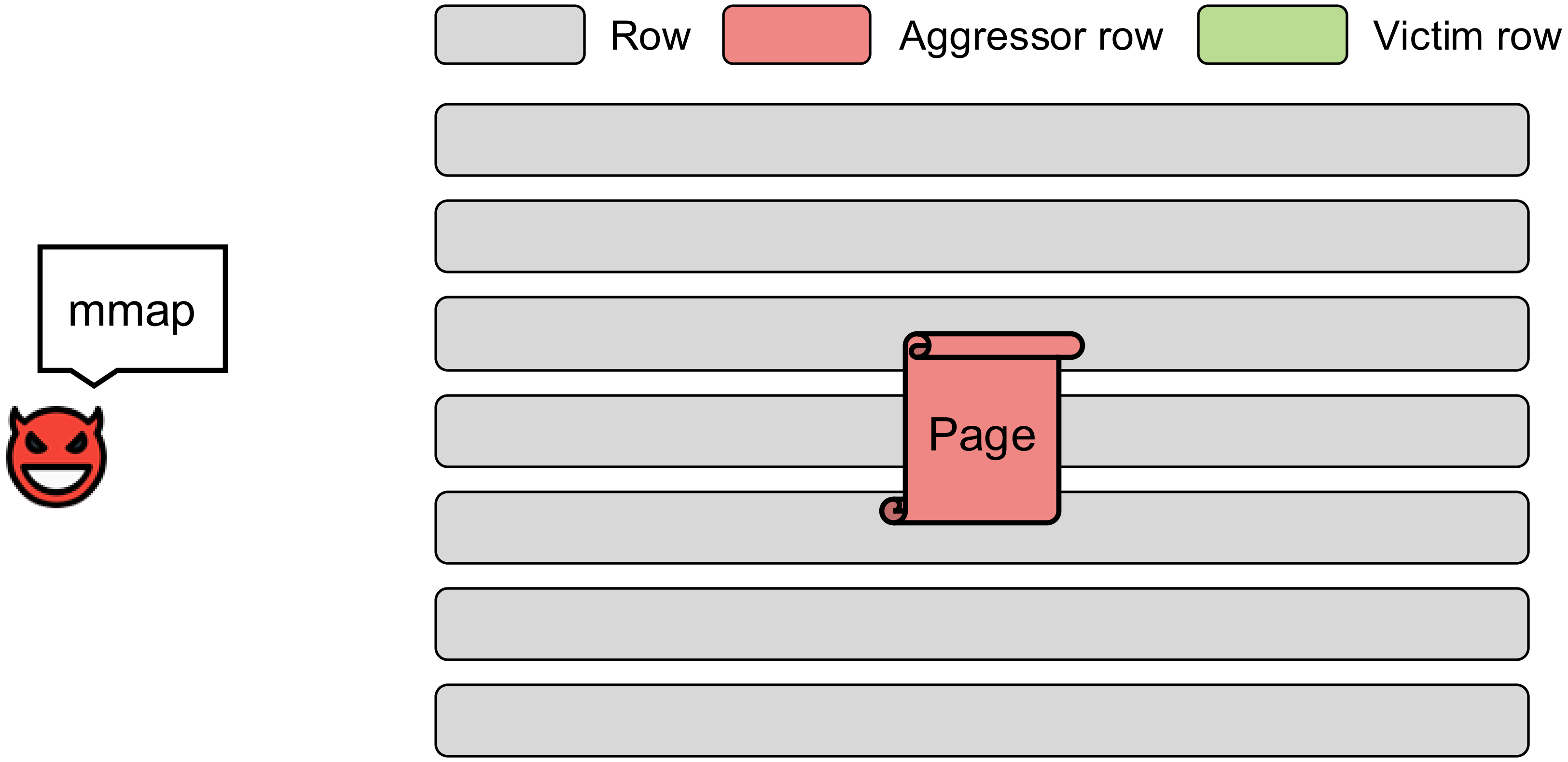


Marionette



# Marionette

- Conventional attack



Row

Aggressor row

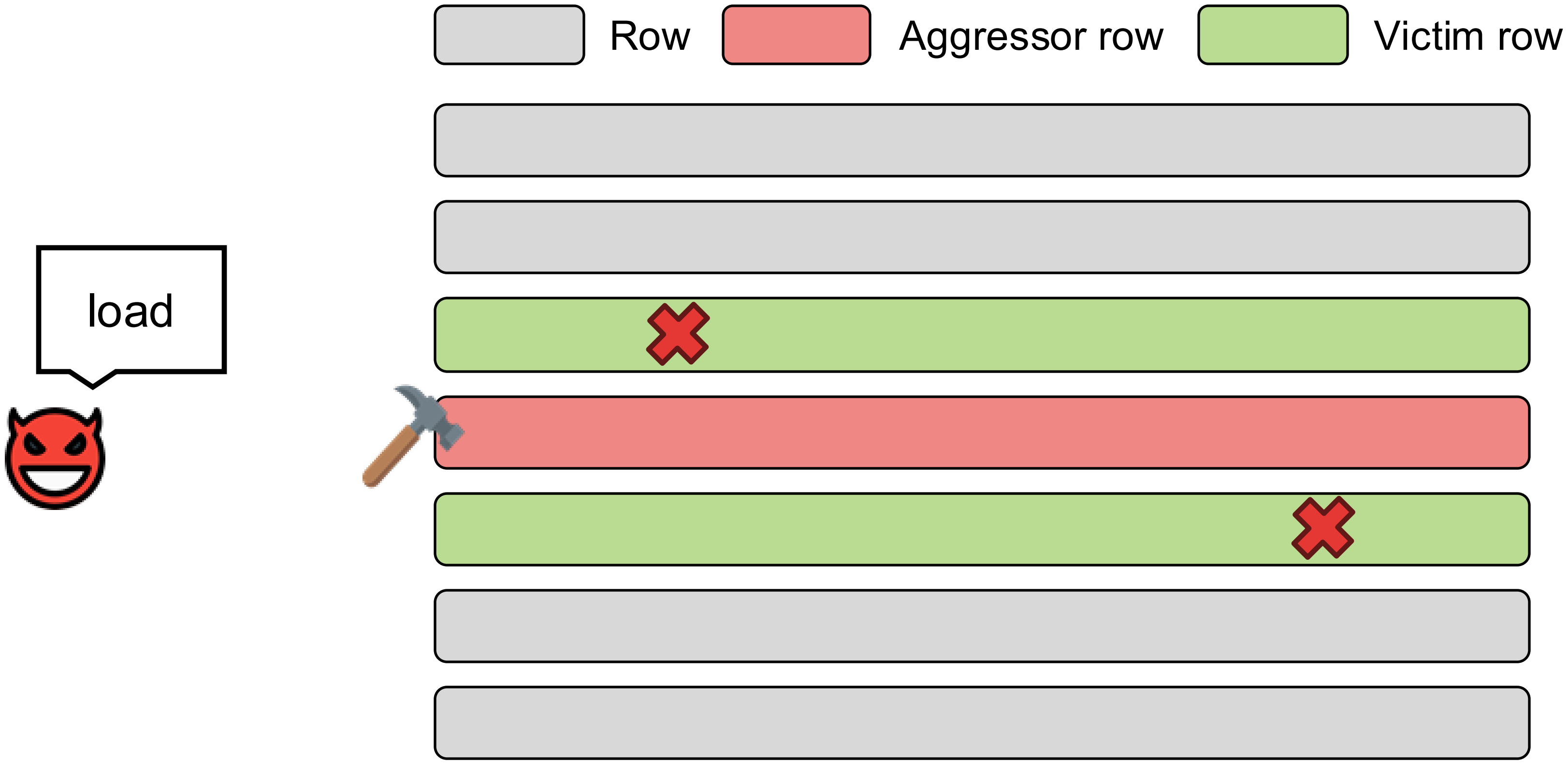
Victim row

mmap

Page

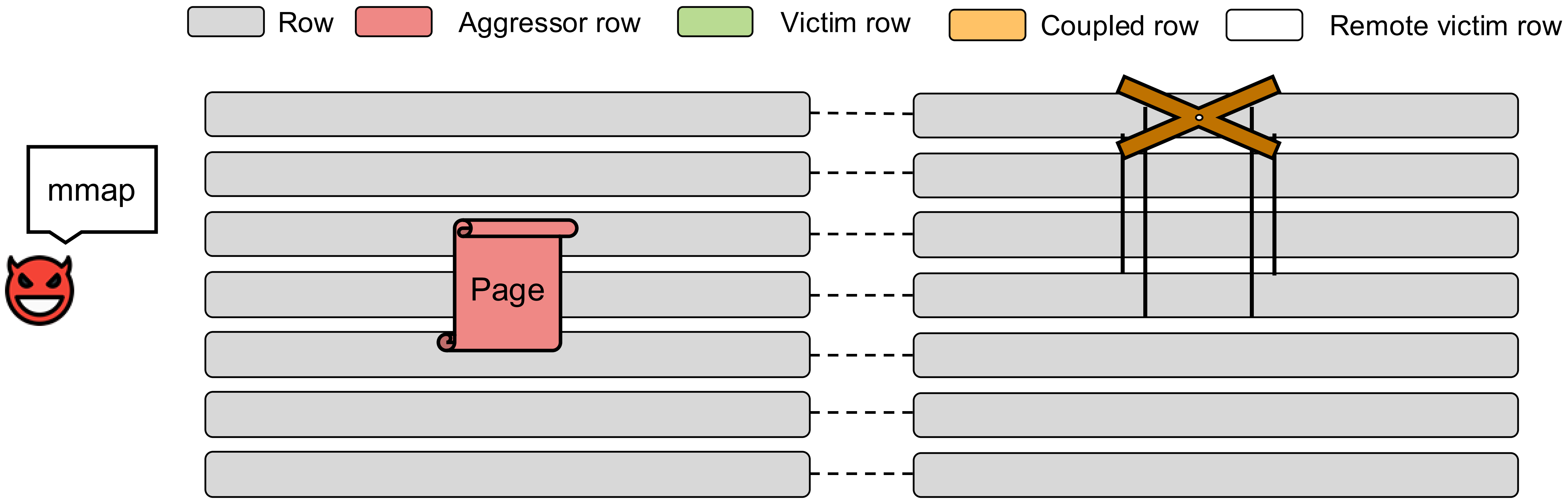
# Marionette

- Conventional attack



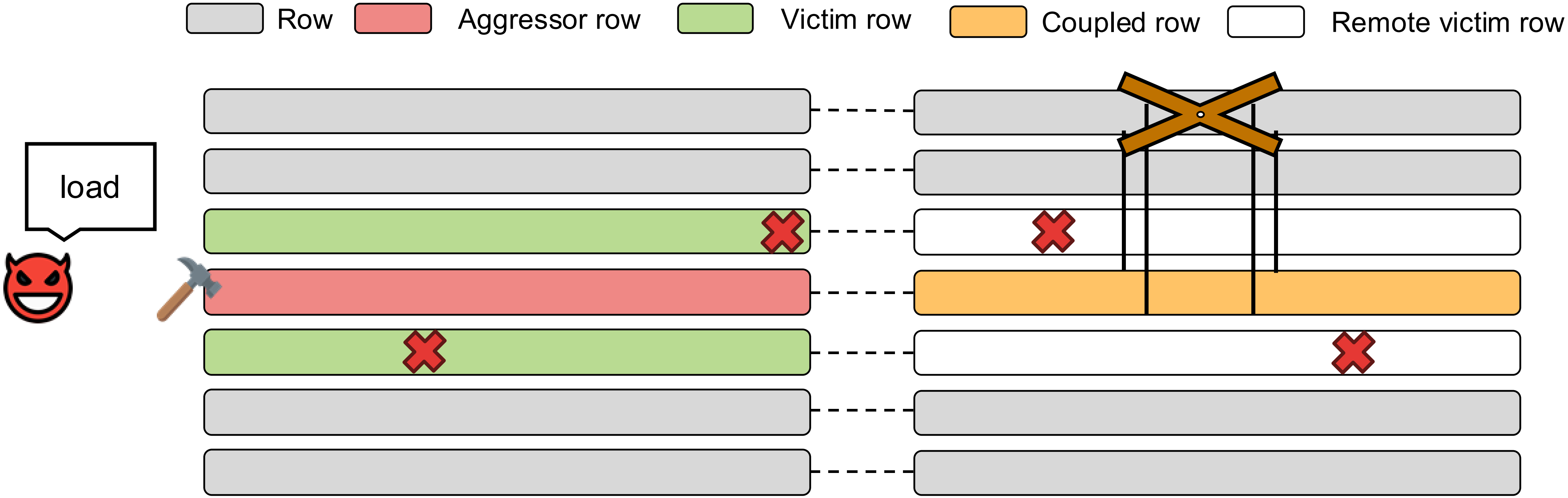
# Marionette

- Coupled-row attack



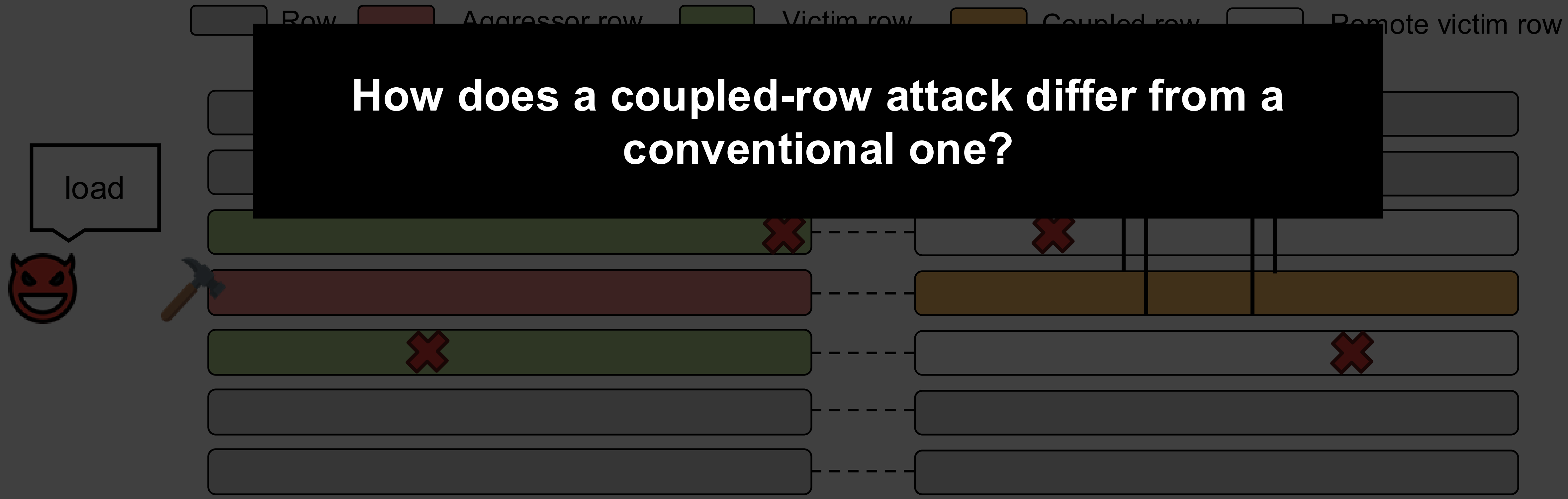
# Marionette

- Coupled-row attack



# Marionette

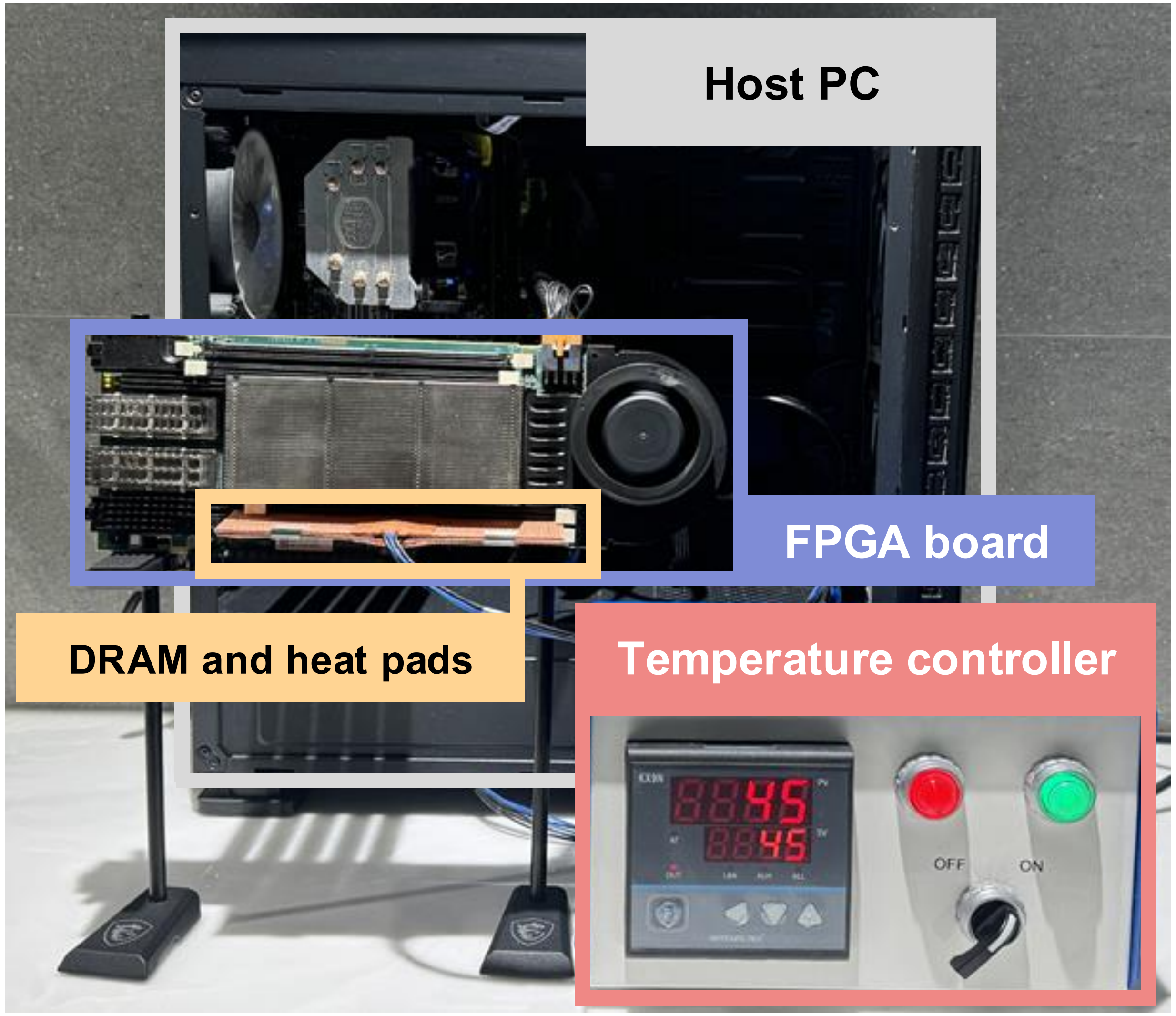
- Coupled-row attack



**How does a coupled-row attack differ from a conventional one?**

# Marionette Analysis

- Experimental setup
  - FPGA
    - = AMD Xilinx Alveo U200, U280
    - = DRAM-Bender[7]
  - Temperature
    - = Temperature controller
    - = 45°C[8]



[7] A. Olgun et al., “DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023.
[8] ANSI/ASHRAE Standard, “Equipment Thernal Guidelines for Data Processing Environments”, 2021.

# Marionette Analysis

- Experimental setup
  - DRAM
    - = DDR4 RDIMM (Registered DIMM), x4 DRAM chips

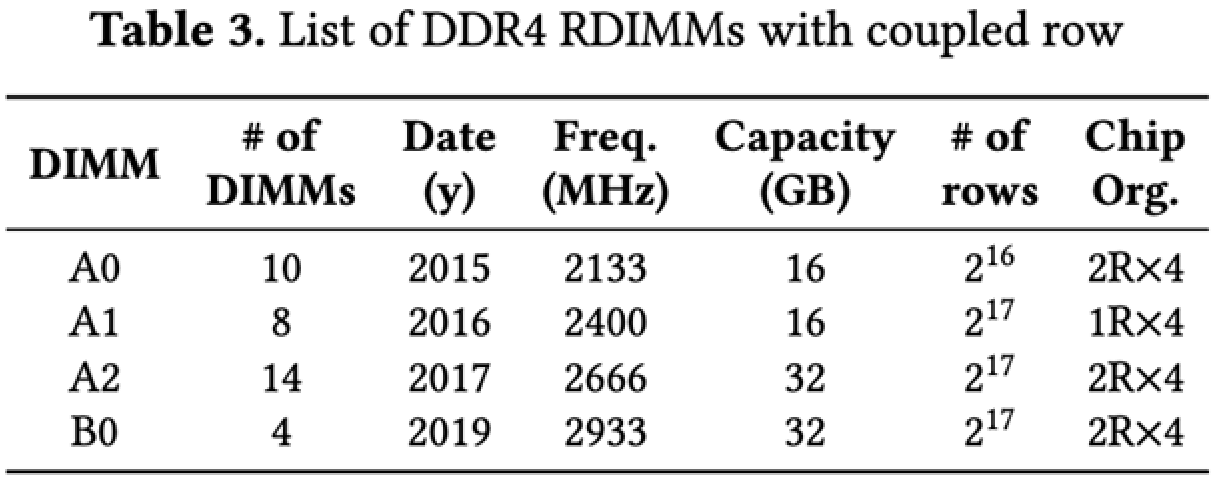
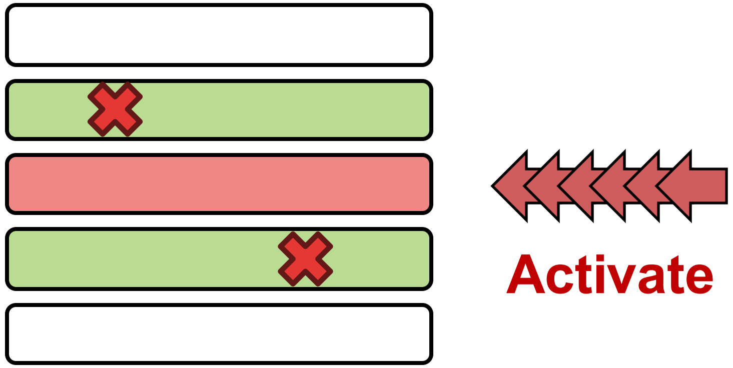


**Table 3.** List of DDR4 RDIMMs with coupled row

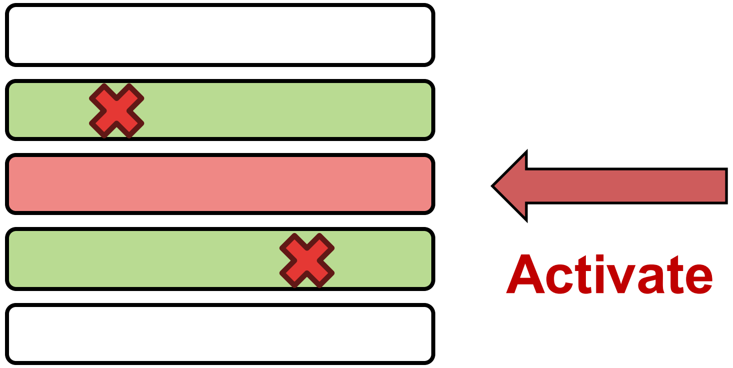
| DIMM | # of DIMMs | Date (y) | Freq. (MHz) | Capacity (GB) | # of rows | Chip Org. |
|---|---|---|---|---|---|---|
| A0 | 10 | 2015 | 2133 | 16 | $2^{16}$ | 2R×4 |
| A1 | 8 | 2016 | 2400 | 16 | $2^{17}$ | 1R×4 |
| A2 | 14 | 2017 | 2666 | 32 | $2^{17}$ | 2R×4 |
| B0 | 4 | 2019 | 2933 | 32 | $2^{17}$ | 2R×4 |

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering
  - RowHammer
  - RowPress[6]: keep an aggressor row activated for a long time.
- In-DRAM TRR against coupled rows



< RowHammer >



< RowPress >

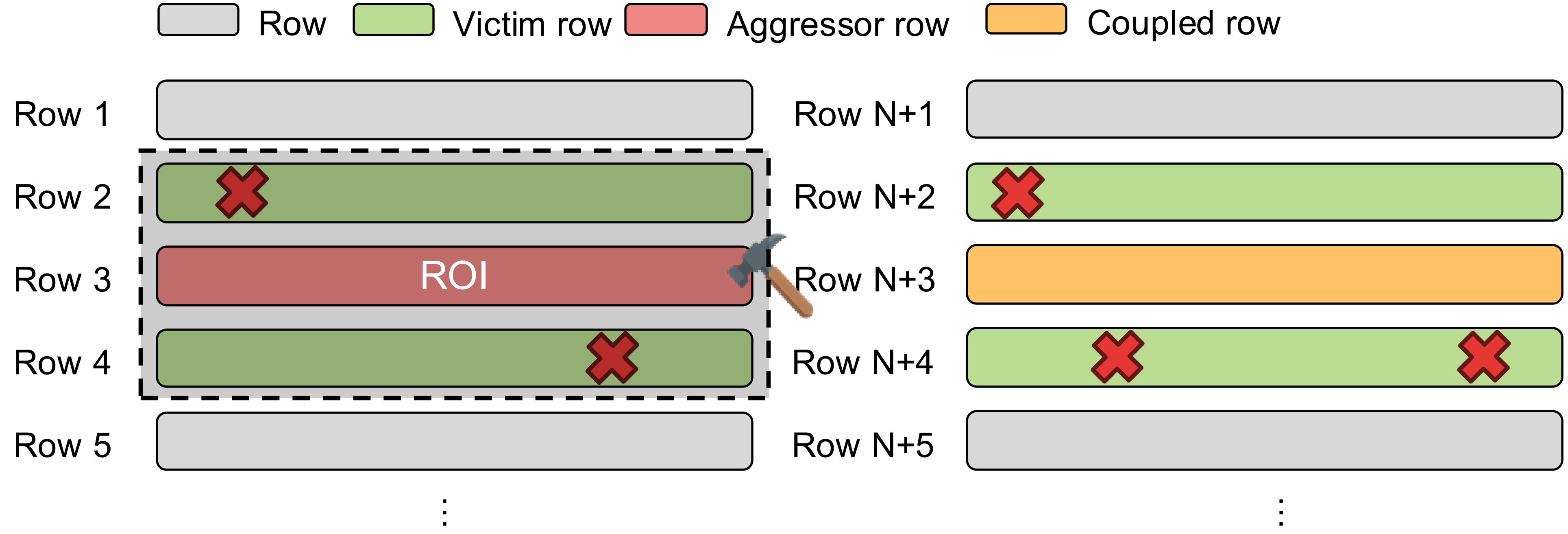
[6] H. Luo et al., “RowPress: Amplifying Read Disturbance in Modern DRAM Chips,” ISCA, 2023.

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering
  - Comparing bitflip locations and the number of bitflips
  - Attack pattern
    - Conventional hammering (baseline)
    - Pure coupled-row hammering
    - Interleaved coupled-row hammering

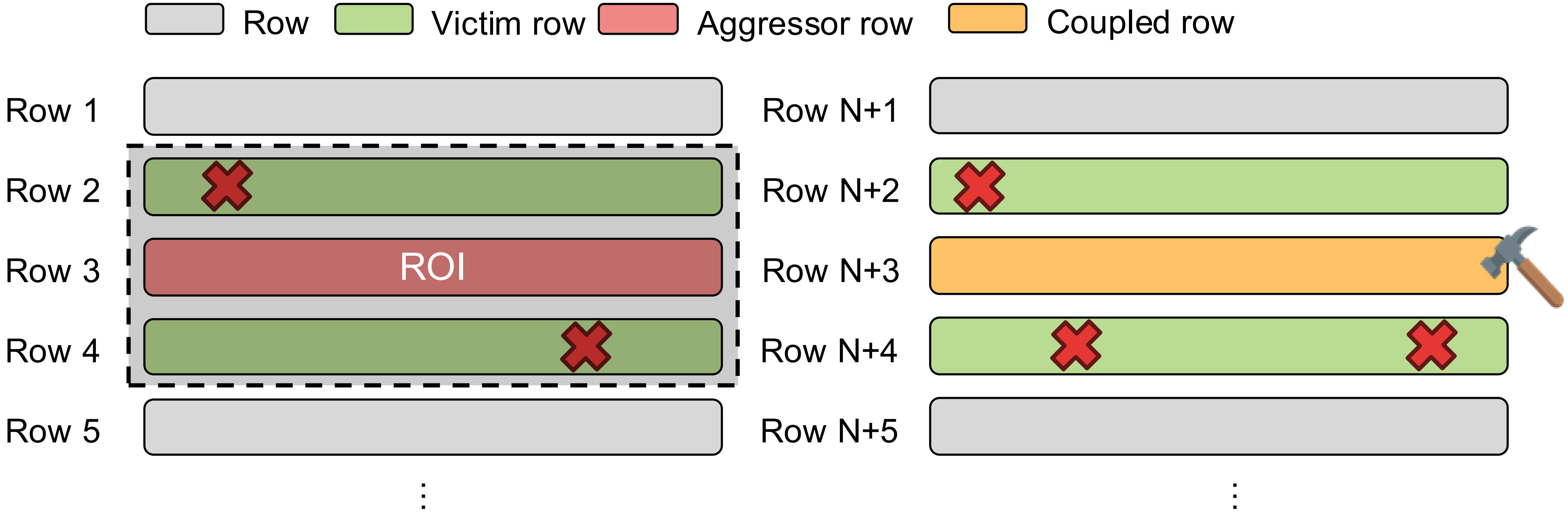
# Marionette Analysis

- Bitflip characteristics of coupled-row hammering
  - Attack pattern
    - = Conventional hammering
      - + E.g., {Row 3}$^N$



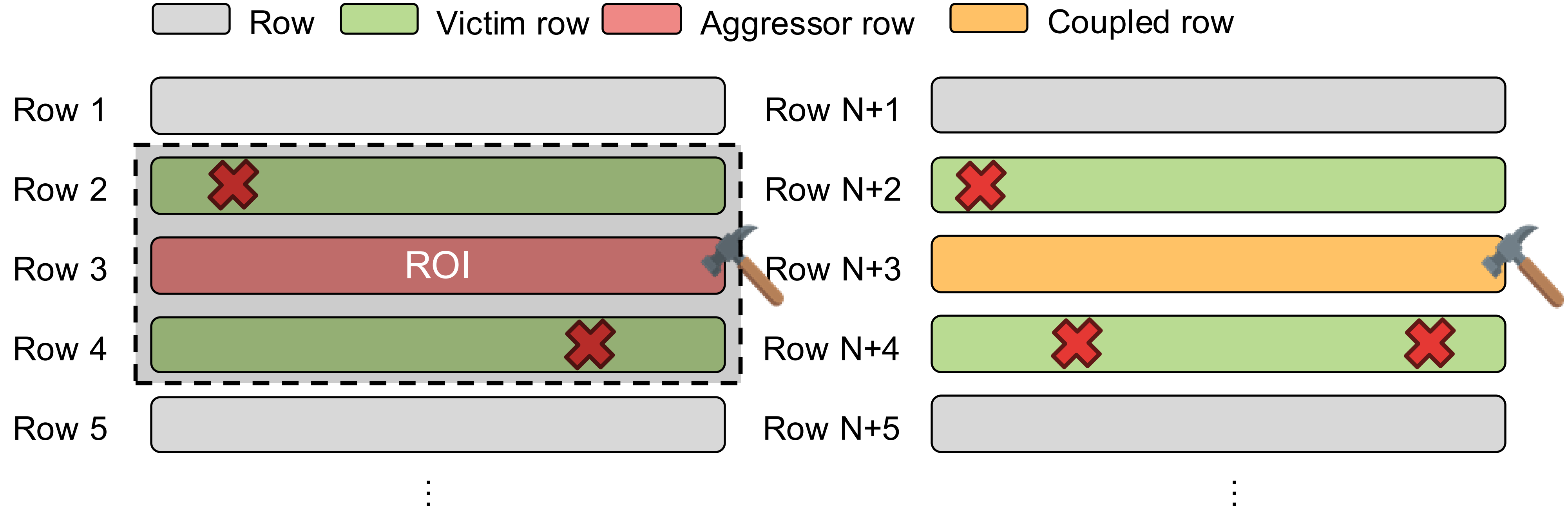
# Marionette Analysis

- Bitflip characteristics of coupled-row hammering
  - – Attack pattern
    - = Pure coupled-row hammering
      - + E.g., {Row N+3}$^N$



# Marionette Analysis

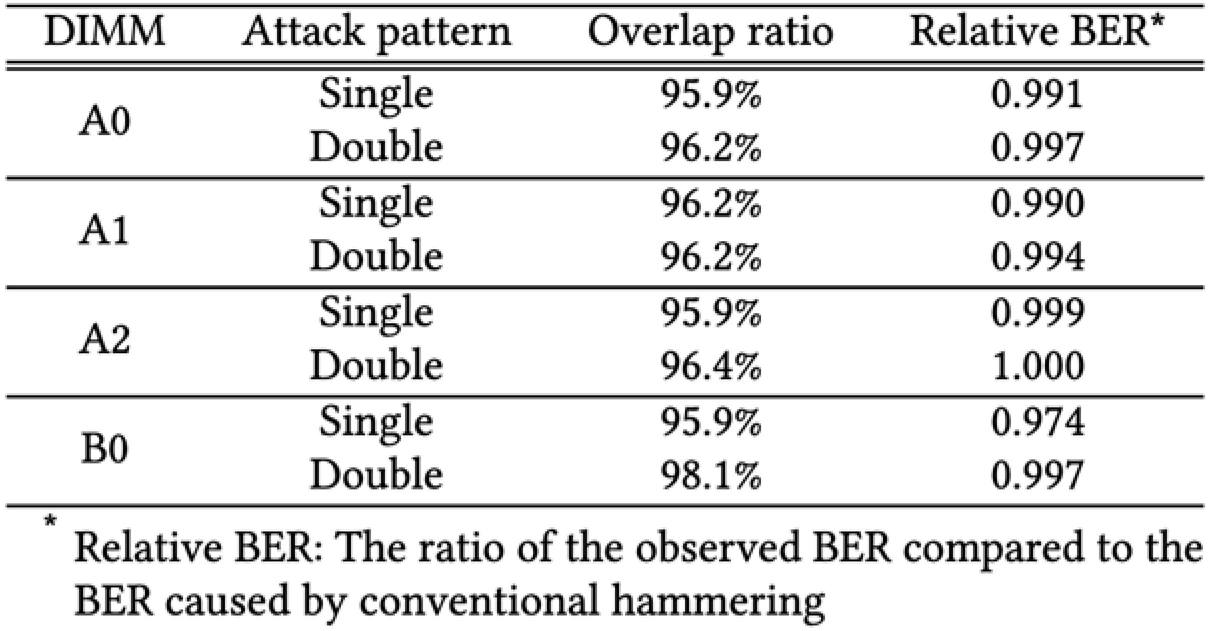
- Bitflip characteristics of coupled-row hammering
  - Attack pattern
    = Interleaved coupled-row hammering
      + E.g., {Row 3, Row N+3}$^{N/2}$



# Marionette Analysis

- Bitflip characteristics of coupled-row hammering (RowHammer)
  - Bit-error-rate (BER) = # of bitflips / # of bits in victim rows

**Pure coupled-row hammering** vs conventional hammering



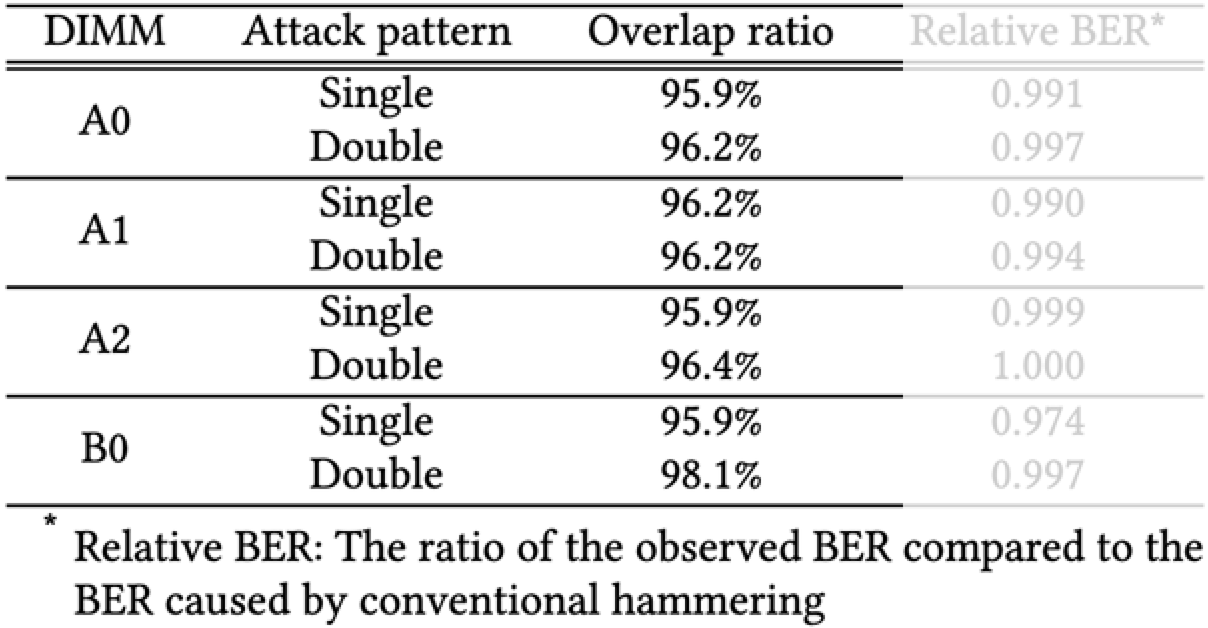
| DIMM | Attack pattern | Overlap ratio | Relative BER* |
|---|---|---|---|
| A0 | Single | 95.9% | 0.991 |
| | Double | 96.2% | 0.997 |
| A1 | Single | 96.2% | 0.990 |
| | Double | 96.2% | 0.994 |
| A2 | Single | 95.9% | 0.999 |
| | Double | 96.4% | 1.000 |
| B0 | Single | 95.9% | 0.974 |
| | Double | 98.1% | 0.997 |

* Relative BER: The ratio of the observed BER compared to the BER caused by conventional hammering

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering (RowHammer)
  - Bit-error-rate (BER) = # of bitflips / # of bits in victim rows

**Pure coupled-row hammering** vs conventional hammering



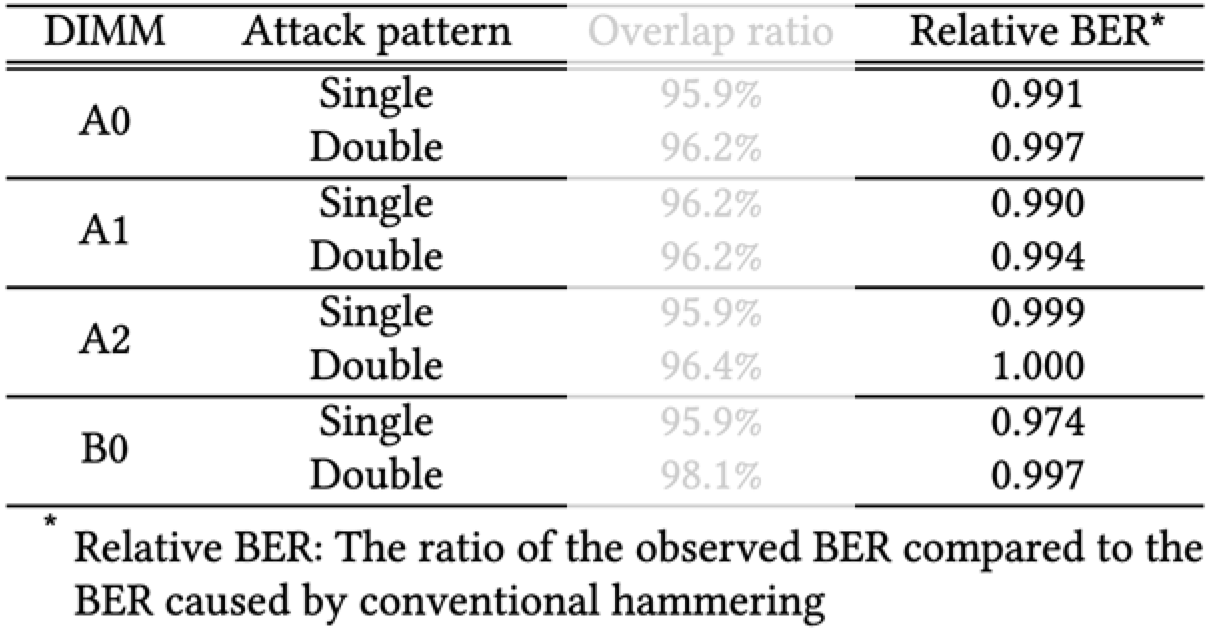
| DIMM | Attack pattern | Overlap ratio | Relative BER* |
|---|---|---|---|
| A0 | Single | 95.9% | 0.991 |
| | Double | 96.2% | 0.997 |
| A1 | Single | 96.2% | 0.990 |
| | Double | 96.2% | 0.994 |
| A2 | Single | 95.9% | 0.999 |
| | Double | 96.4% | 1.000 |
| B0 | Single | 95.9% | 0.974 |
| | Double | 98.1% | 0.997 |

* Relative BER: The ratio of the observed BER compared to the BER caused by conventional hammering

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering (RowHammer)
  - – Bit-error-rate (BER) = # of bitflips / # of bits in victim rows

**Pure coupled-row hammering** vs conventional hammering



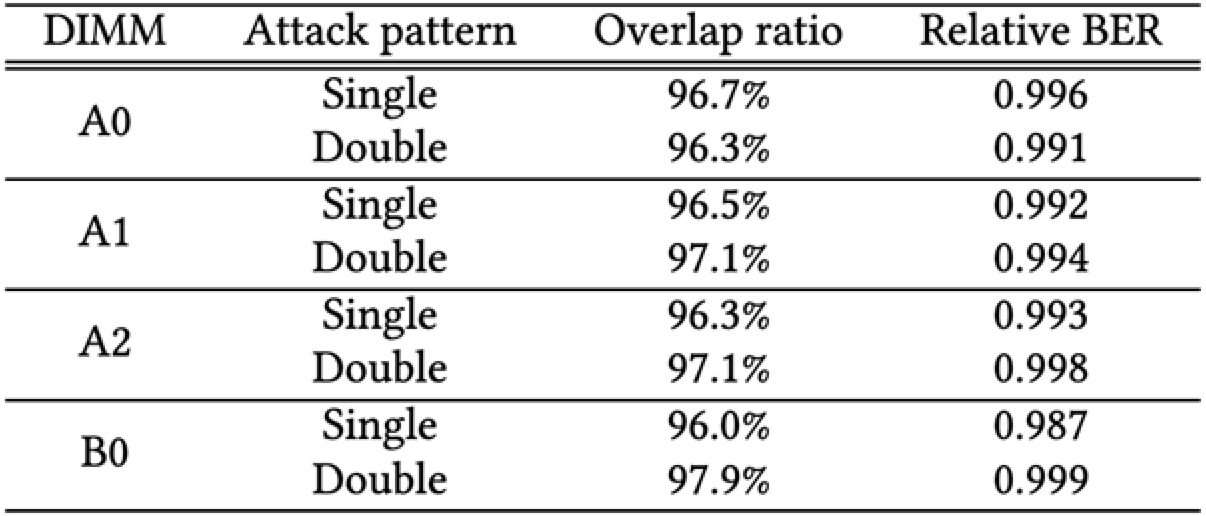
| DIMM | Attack pattern | Overlap ratio | Relative BER* |
|---|---|---|---|
| A0 | Single | 95.9% | 0.991 |
| | Double | 96.2% | 0.997 |
| A1 | Single | 96.2% | 0.990 |
| | Double | 96.2% | 0.994 |
| A2 | Single | 95.9% | 0.999 |
| | Double | 96.4% | 1.000 |
| B0 | Single | 95.9% | 0.974 |
| | Double | 98.1% | 0.997 |

* Relative BER: The ratio of the observed BER compared to the BER caused by conventional hammering

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering (RowHammer)

**Interleaved coupled-row hammering** vs conventional hammering

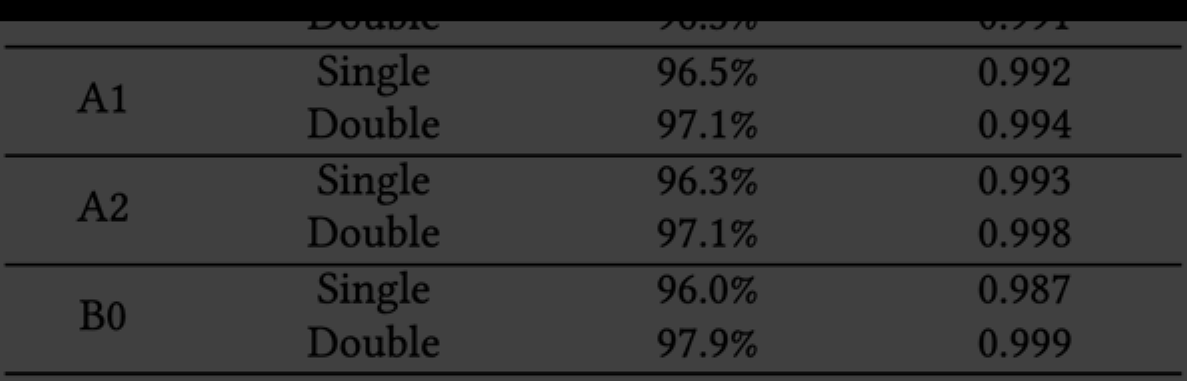


| DIMM | Attack pattern | Overlap ratio | Relative BER |
|---|---|---|---|
| A0 | Single | 96.7% | 0.996 |
| | Double | 96.3% | 0.991 |
| A1 | Single | 96.5% | 0.992 |
| | Double | 97.1% | 0.994 |
| A2 | Single | 96.3% | 0.993 |
| | Double | 97.1% | 0.998 |
| B0 | Single | 96.0% | 0.987 |
| | Double | 97.9% | 0.999 |

# Marionette Analysis

- Bitflip characteristics of coupled-row hammering (RowHammer)
  - **Interleaved coupled-row hammering** vs **conventional hammering**

**Coupled-row hammering has RowHammer/RowPress capabilities that are highly similar to those of conventional hammering.**



| | | | |
|---|---|---|---|
| | Double | 96.3% | 0.991 |
| A1 | Single | 96.5% | 0.992 |
| | Double | 97.1% | 0.994 |
| A2 | Single | 96.3% | 0.993 |
| | Double | 97.1% | 0.998 |
| B0 | Single | 96.0% | 0.987 |
| | Double | 97.9% | 0.999 |

# Marionette Analysis

- In-DRAM TRR against coupled rows

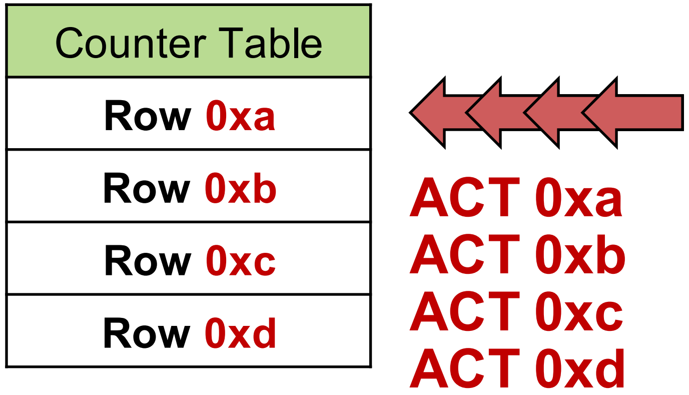
# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Uncovering TRR (U-TRR)[9]
    - = Tool for reverse engineering In-DRAM TRR behavior in an FPGA environment

[9] H. Hassan et al., “Uncovering In-DRAM RowHammer Protection Mechanism: A New Methodology, Custom RowHammer Patterns, and Implications," MICRO, 2021.

# Marionette Analysis

- In-DRAM TRR against coupled rows

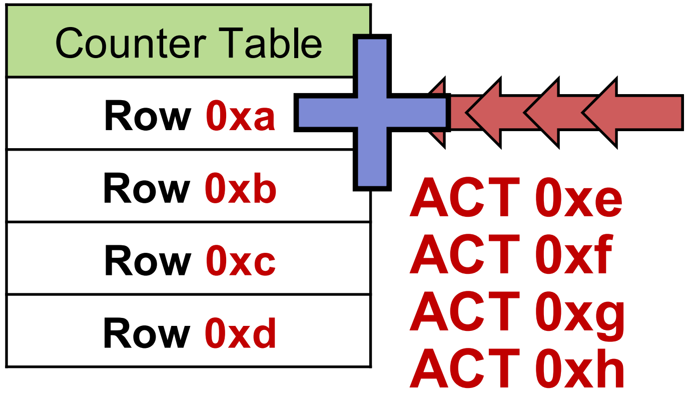


| Counter Table |
|---|
| Row 0xa |
| Row 0xb |
| Row 0xc |
| Row 0xd |

ACT 0xa
ACT 0xb
ACT 0xc
ACT 0xd

# Marionette Analysis

- In-DRAM TRR against coupled rows

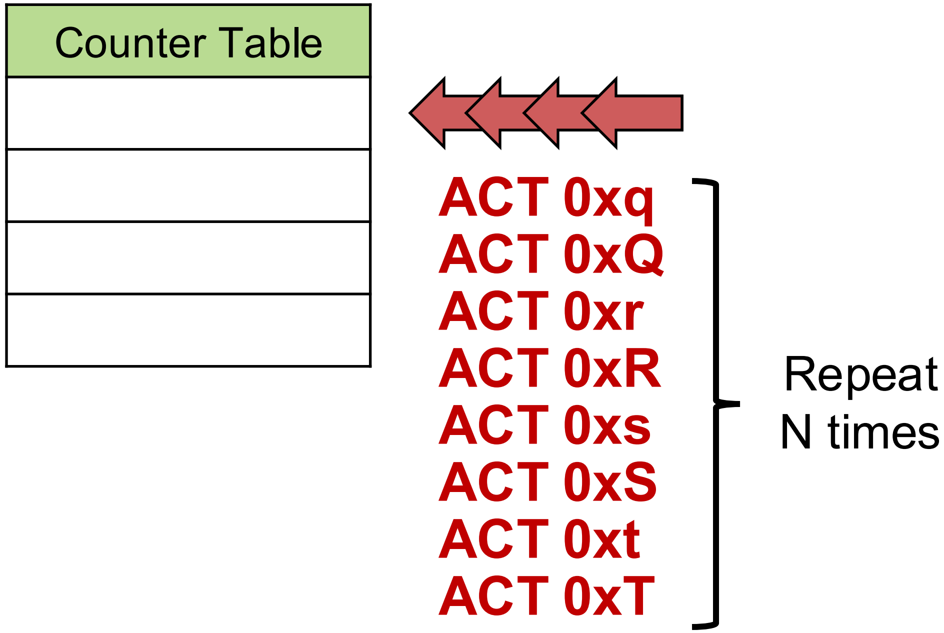


| Counter Table |
|---|
| Row 0xa |
| Row 0xb |
| Row 0xc |
| Row 0xd |

ACT 0xe
ACT 0xf
ACT 0xg
ACT 0xh

# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Hammering sequence: {0xq, 0xQ, 0xr, 0xR, 0xs, 0xS, 0xt, 0xT}$^N$
    - = Row **0xq**, **0xr**, **0xs**, and **0xt** are coupled row of **0xQ**, **0xR**, **0xS**, and **0xT**, respectively.



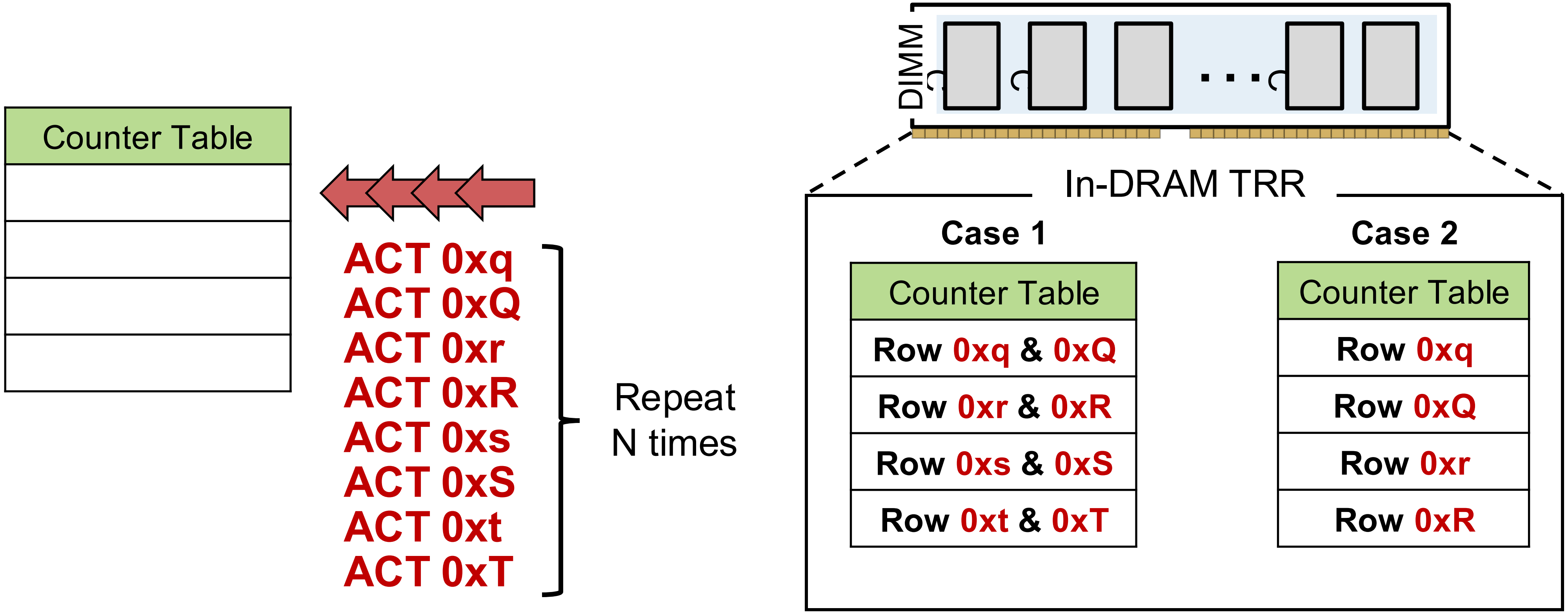
Counter Table

ACT 0xq
ACT 0xQ
ACT 0xr
ACT 0xR
ACT 0xs
ACT 0xS
ACT 0xt
ACT 0xT

Repeat
N times

# Marionette Analysis

- In-DRAM TRR against coupled rows



| Counter Table |
|---|
| |
| |
| |
| |

ACT 0xq
ACT 0xQ
ACT 0xr
ACT 0xR
ACT 0xs
ACT 0xS
ACT 0xt
ACT 0xT

Repeat
N times

In-DRAM TRR

**Case 1**

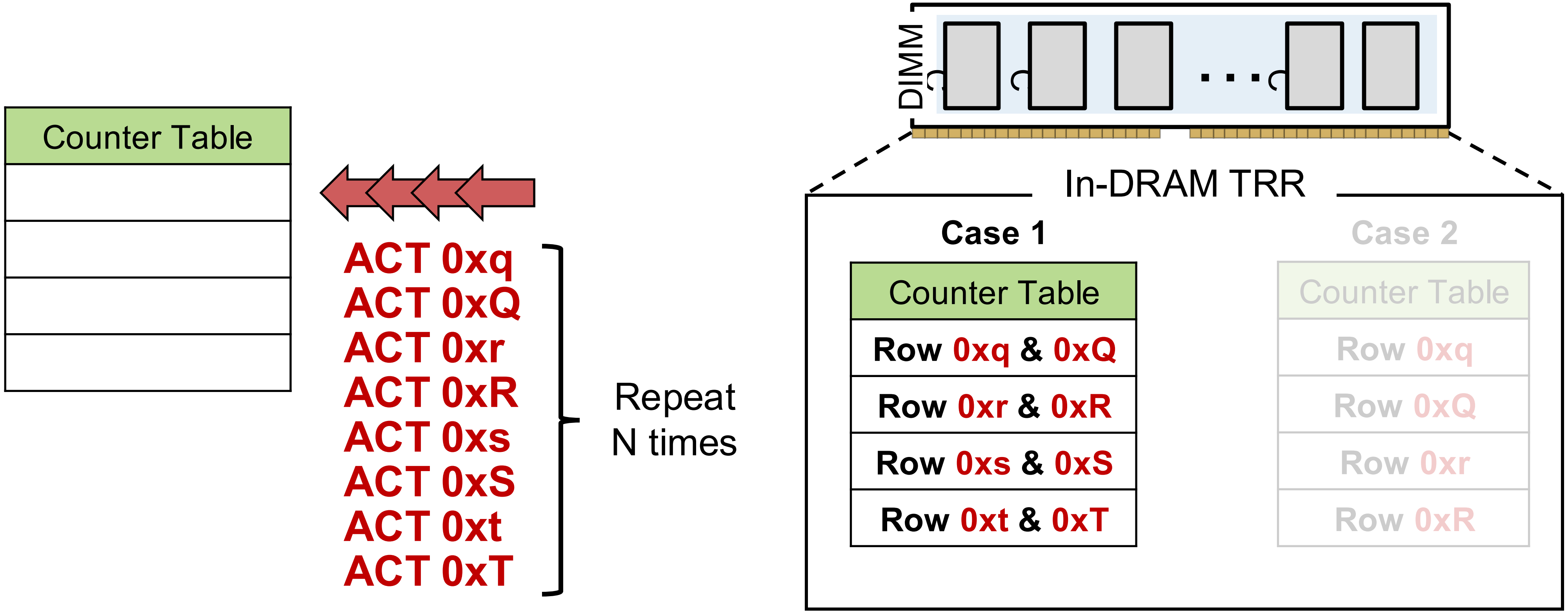
| Counter Table |
|---|
| **Row 0xq & 0xQ** |
| **Row 0xr & 0xR** |
| **Row 0xs & 0xS** |
| **Row 0xt & 0xT** |

**Case 2**

| Counter Table |
|---|
| **Row 0xq** |
| **Row 0xQ** |
| **Row 0xr** |
| **Row 0xR** |

# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Case 1: all rows are tracked.



| Counter Table |
|---|
| |
| |
| |
| |

ACT 0xq
ACT 0xQ
ACT 0xr
ACT 0xR
ACT 0xs
ACT 0xS
ACT 0xt
ACT 0xT

Repeat
N times

Case 1

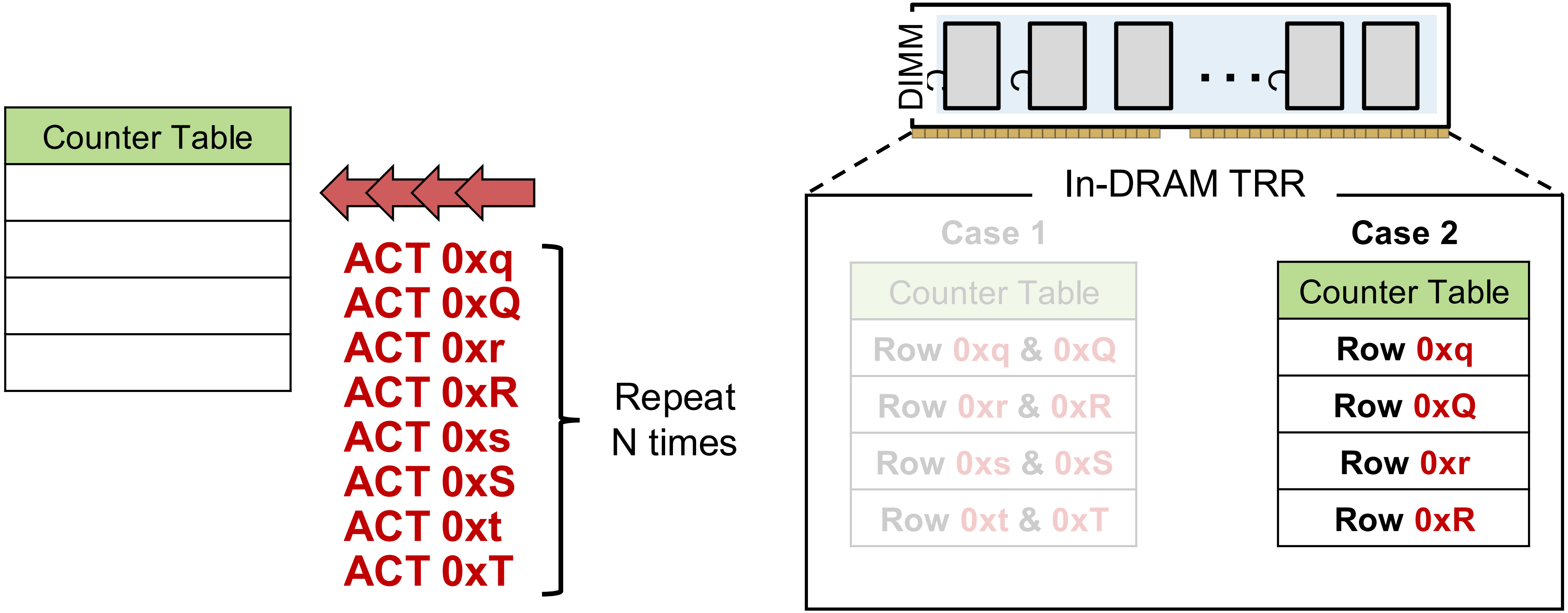
| Counter Table |
|---|
| Row 0xq & 0xQ |
| Row 0xr & 0xR |
| Row 0xs & 0xS |
| Row 0xt & 0xT |

Case 2

| Counter Table |
|---|
| Row 0xq |
| Row 0xQ |
| Row 0xr |
| Row 0xR |

# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Case 1: all rows are tracked.
  - Case 2: row **0xs (0xS)** and row **0xt (0xT)** are not tracked.



| Counter Table |
|---|
| |
| |
| |
| |

**ACT 0xq**
**ACT 0xQ**
**ACT 0xr**
**ACT 0xR**
**ACT 0xs**
**ACT 0xS**
**ACT 0xt**
**ACT 0xT**

Repeat
N times

In-DRAM TRR

**Case 1**

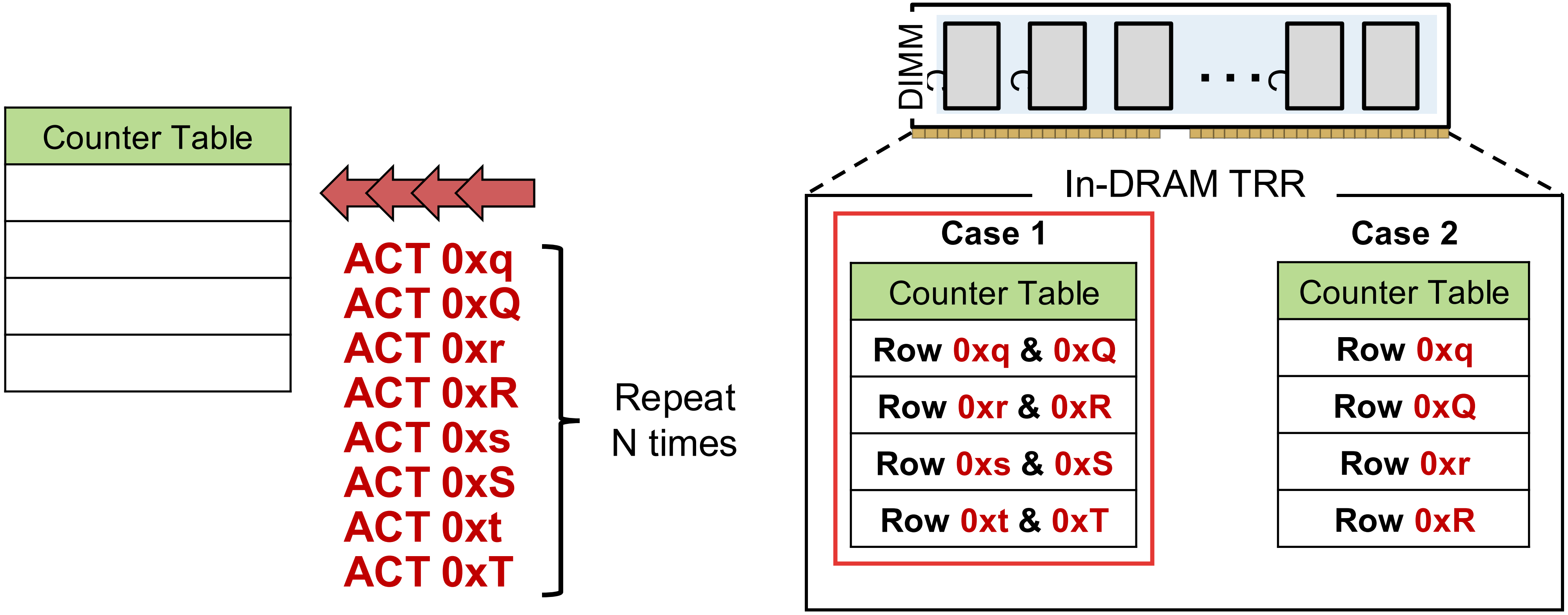
| Counter Table |
|---|
| **Row 0xq & 0xQ** |
| **Row 0xr & 0xR** |
| **Row 0xs & 0xS** |
| **Row 0xt & 0xT** |

**Case 2**

| Counter Table |
|---|
| **Row 0xq** |
| **Row 0xQ** |
| **Row 0xr** |
| **Row 0xR** |

# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Case 1: all rows are tracked.
  - Case 2: row **0xs (0xS)** and row **0xt (0xT)** are not tracked.



| Counter Table |
|---|
| |
| |
| |
| |

**ACT 0xq**
**ACT 0xQ**
**ACT 0xr**
**ACT 0xR**
**ACT 0xs**
**ACT 0xS**
**ACT 0xt**
**ACT 0xT**

Repeat N times

In-DRAM TRR

**Case 1**

| Counter Table |
|---|
| **Row 0xq & 0xQ** |
| **Row 0xr & 0xR** |
| **Row 0xs & 0xS** |
| **Row 0xt & 0xT** |

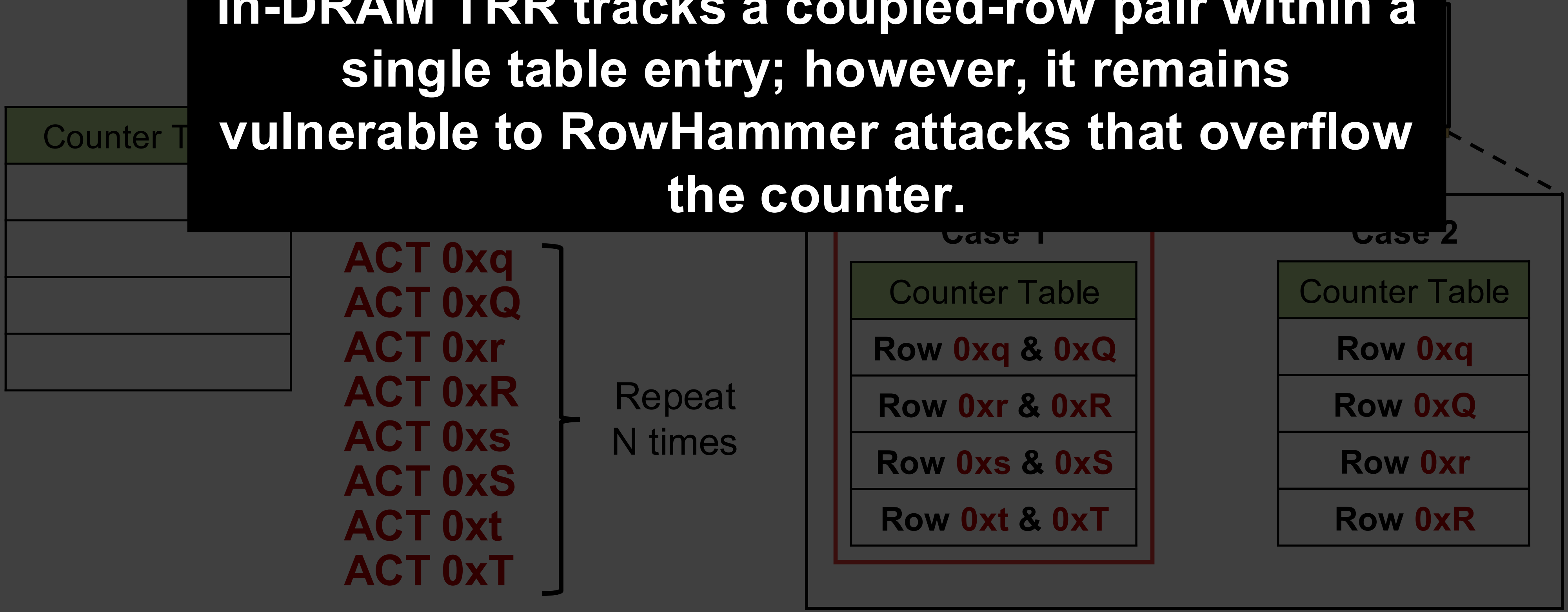
**Case 2**

| Counter Table |
|---|
| **Row 0xq** |
| **Row 0xQ** |
| **Row 0xr** |
| **Row 0xR** |

# Marionette Analysis

- In-DRAM TRR against coupled rows
  - Case 1: all rows are tracked.
  - Case 2: row **0xs (0xS)** and row **0xt (0xT)** are not tracked.

**In-DRAM TRR tracks a coupled-row pair within a single table entry; however, it remains vulnerable to RowHammer attacks that overflow the counter.**



| Counter T |
|---|
| |
| |
| |
| |

ACT 0xq
ACT 0xQ
ACT 0xr
ACT 0xR
ACT 0xs
ACT 0xS
ACT 0xt
ACT 0xT

Repeat N times

Case 1

| Counter Table |
|---|
| Row 0xq & 0xQ |
| Row 0xr & 0xR |
| Row 0xs & 0xS |
| Row 0xt & 0xT |

Case 2

| Counter Table |
|---|
| Row 0xq |
| Row 0xQ |
| Row 0xr |
| Row 0xR |

# Capabilities of Marionette

# Capabilities of Marionette

- Environmental setup
  - – System-a: E5-2620 v3 (Haswell), **ECC-disabled**, single DIMM per socket
  - – DIMM A0, A1, A2, B0 (DDR4 RDIMM)

# Capabilities of Marionette

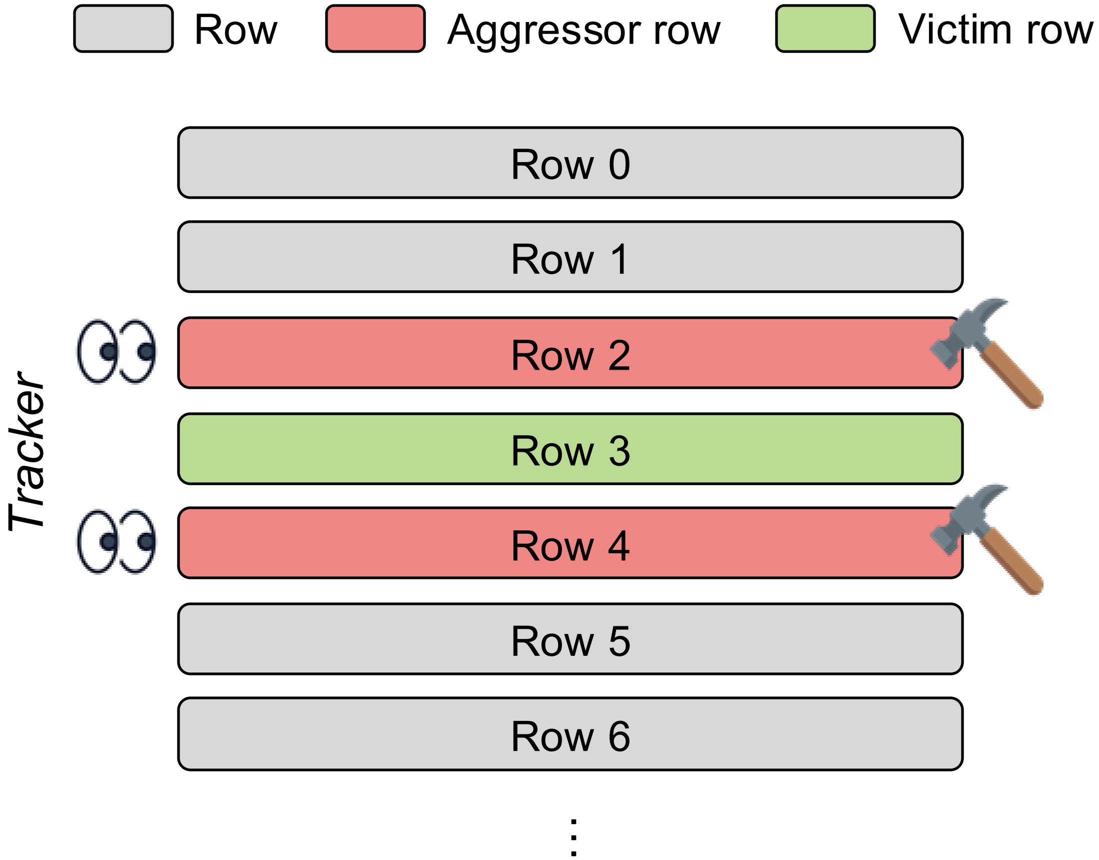
- Software-based mitigations
  - Readily deployable

# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based
  - Isolation-based

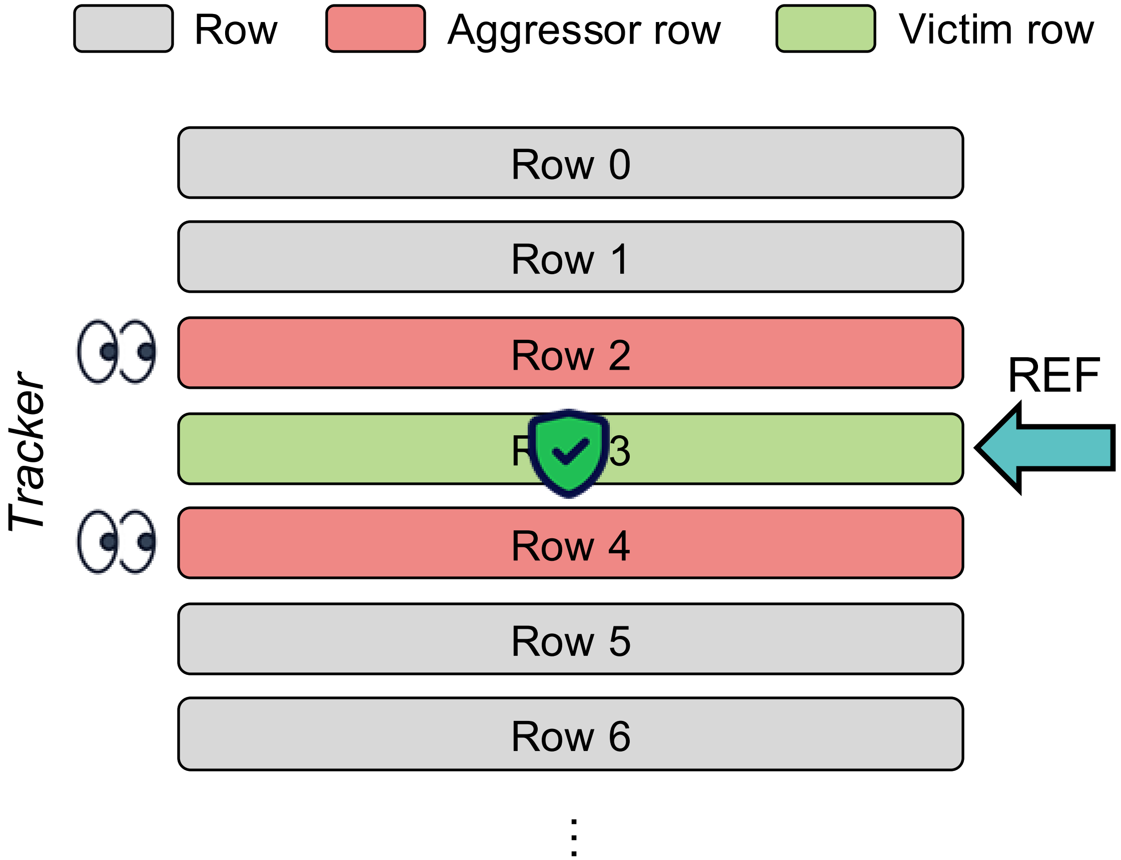
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based



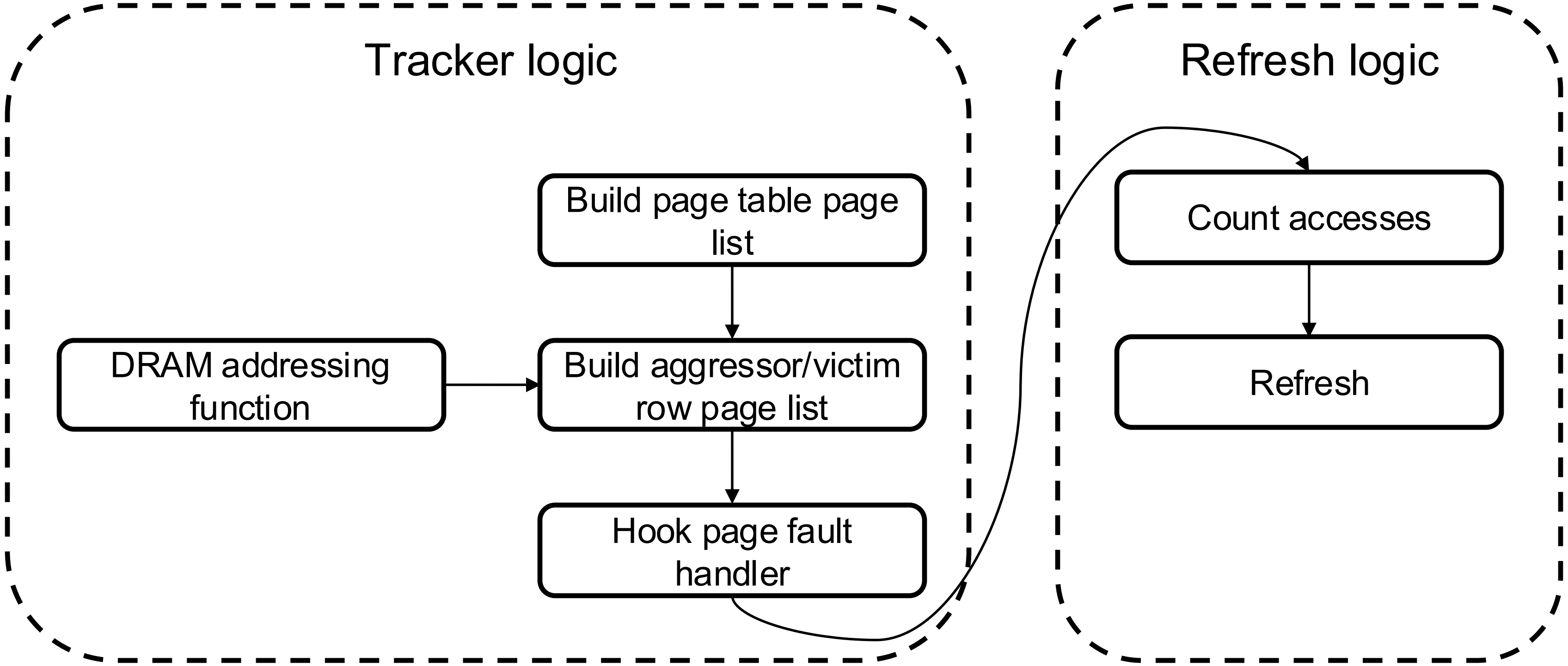
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based



# Capabilities of Marionette

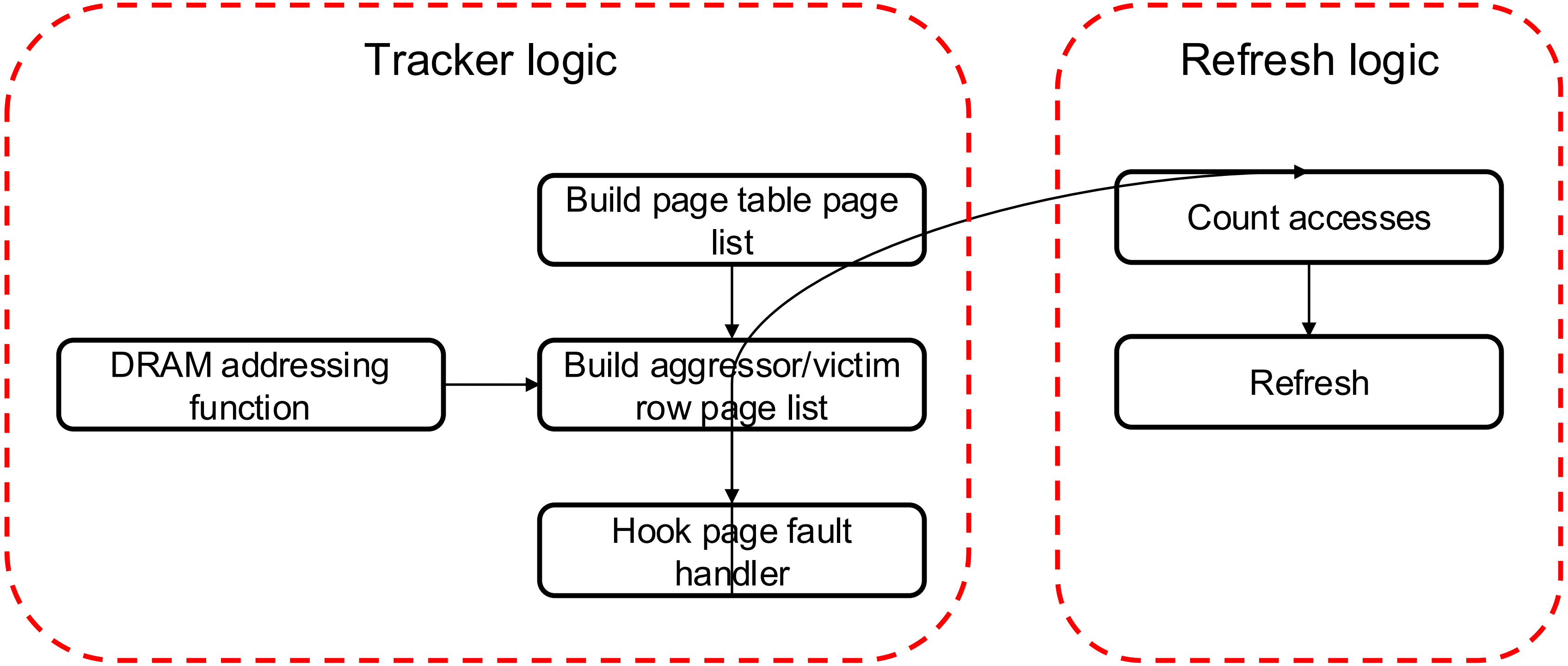
- Software-based mitigations
  - Tracking-based
    - = E.g., SoftTRR[10]



[10] Z. Zhang et al., “SoftTRR: Protect Page Tables against RowHammer Attacks using Software-only Target Row Refresh," USENIX ATC, 2022.

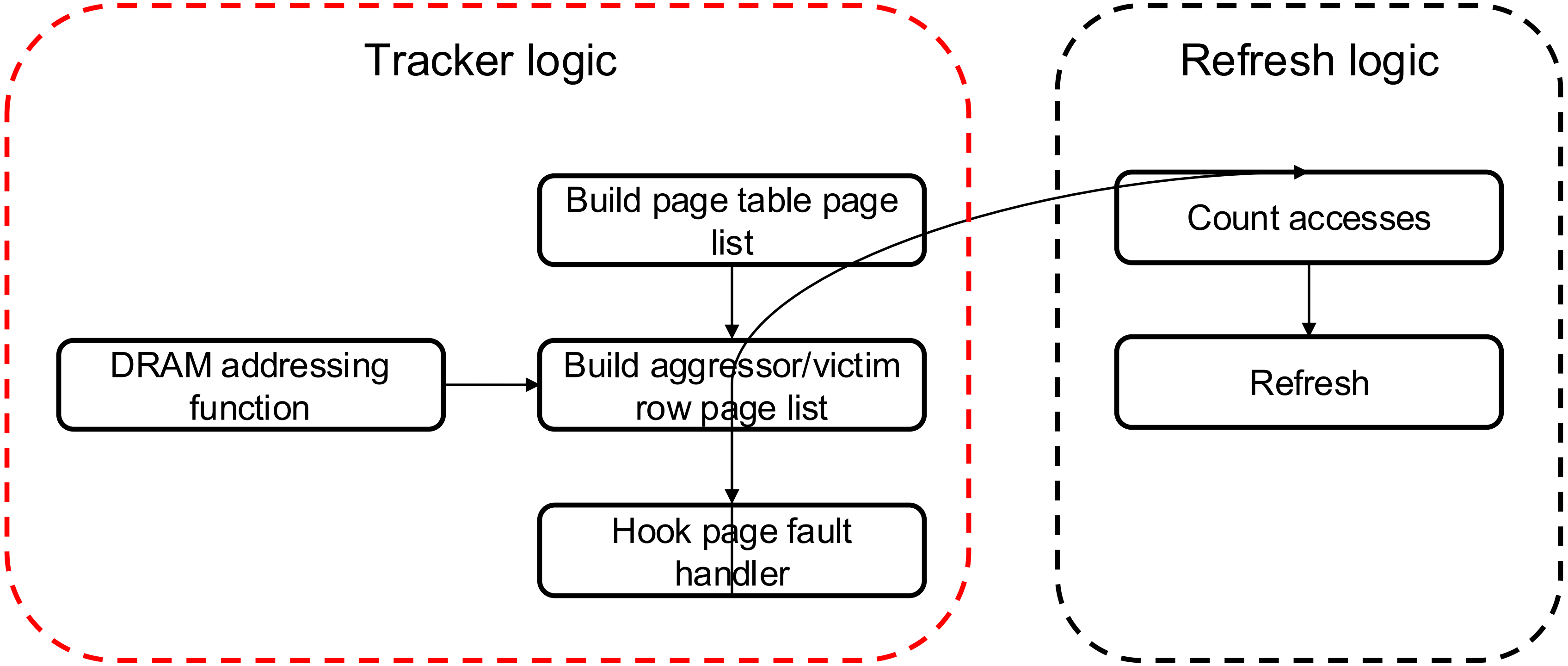
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based
    - E.g., SoftTRR



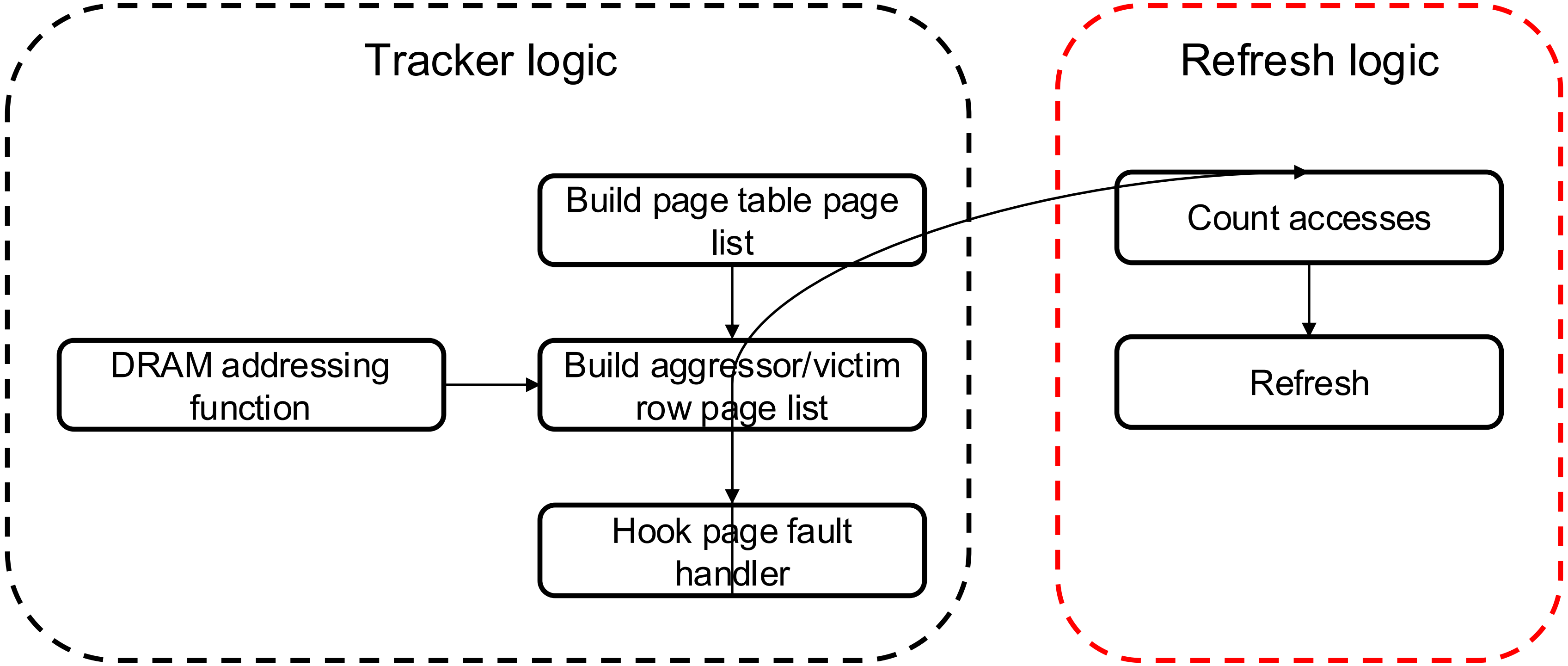
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based
    - E.g., SoftTRR



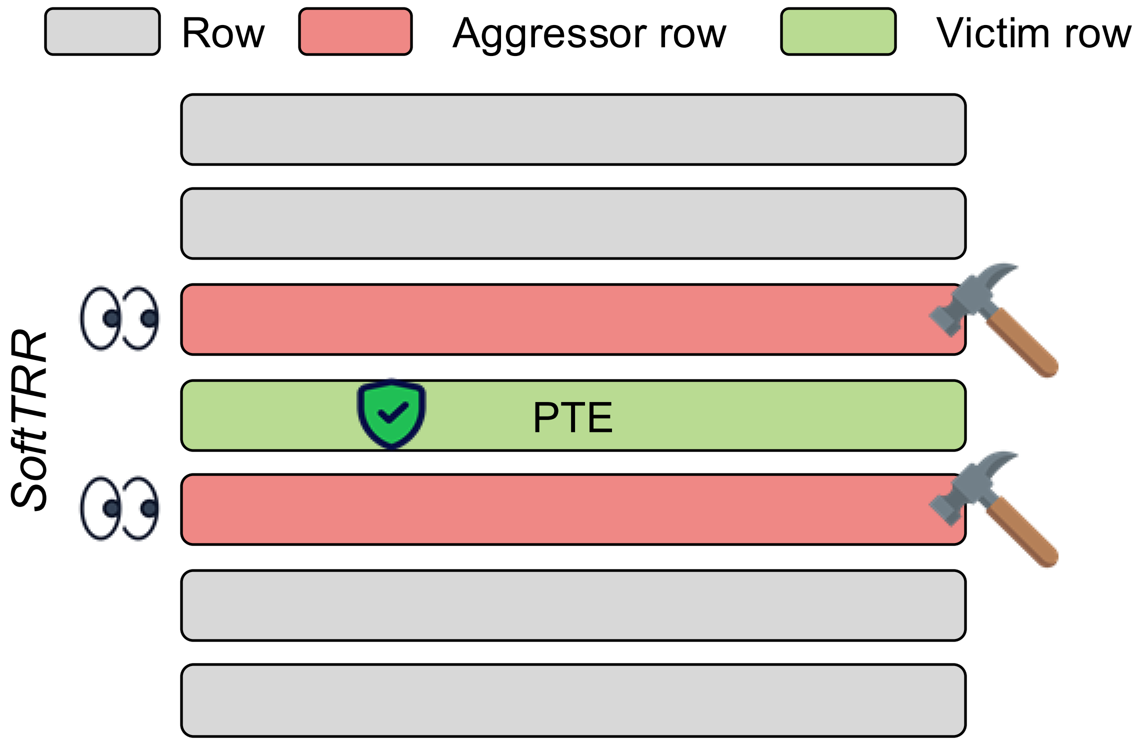
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based
    - = E.g., SoftTRR



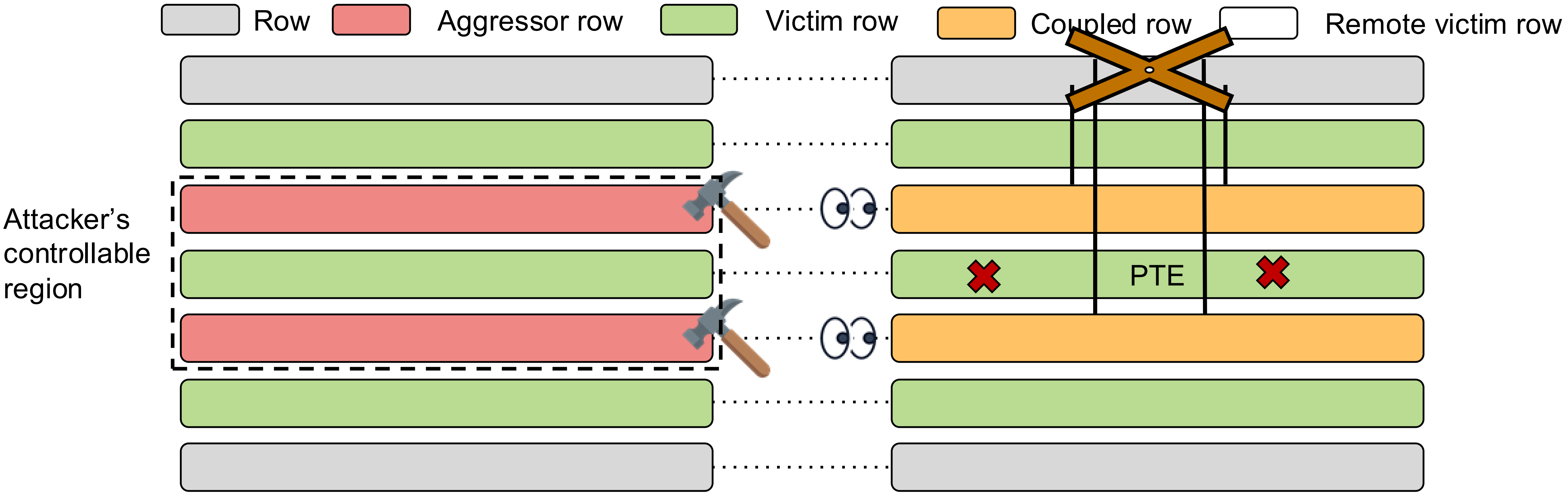
# Capabilities of Marionette

- Software-based mitigations
  - Tracking-based
    - E.g., SoftTRR



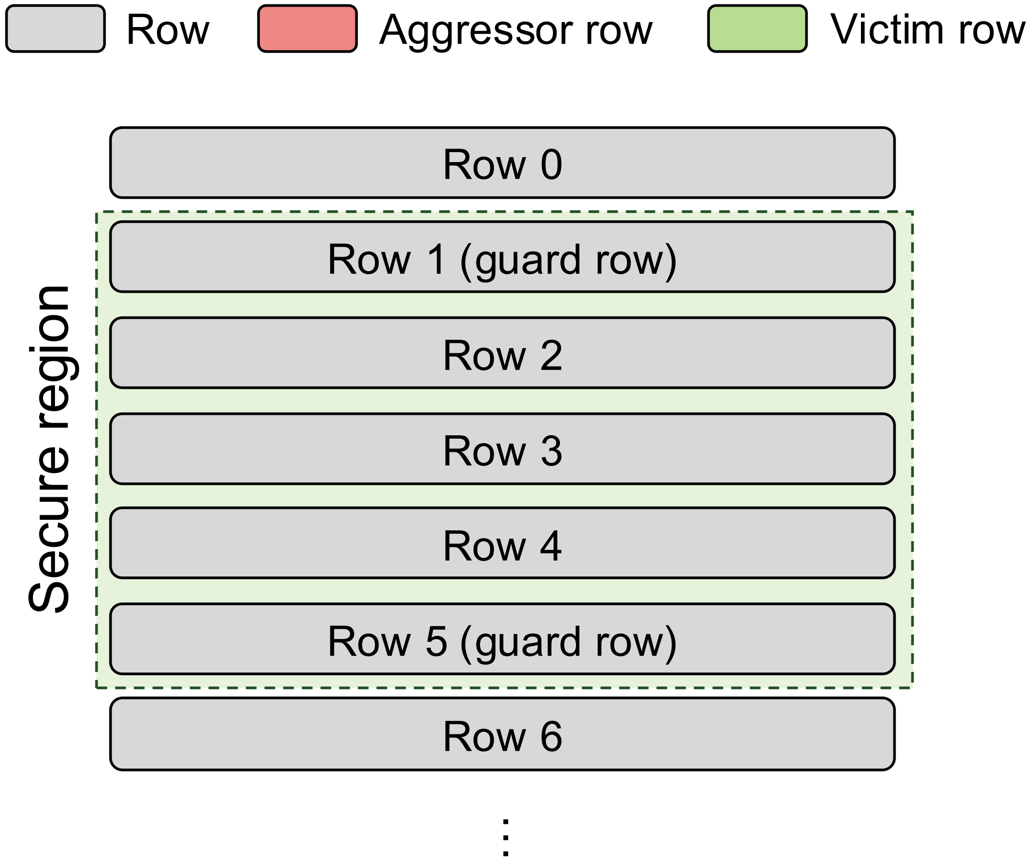
# Capabilities of Marionette

- Software-based mitigations
  - – Tracking-based
    - = E.g., SoftTRR



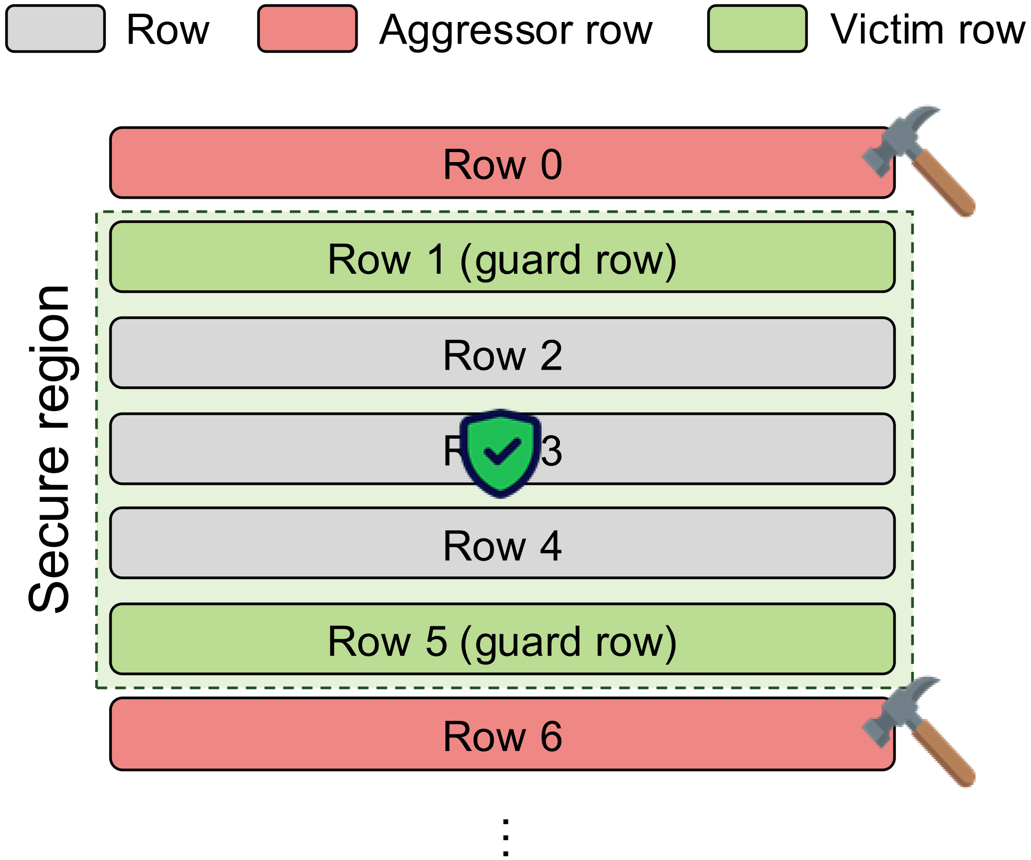
# Capabilities of Marionette

- Software-based mitigations
  - Isolation-based



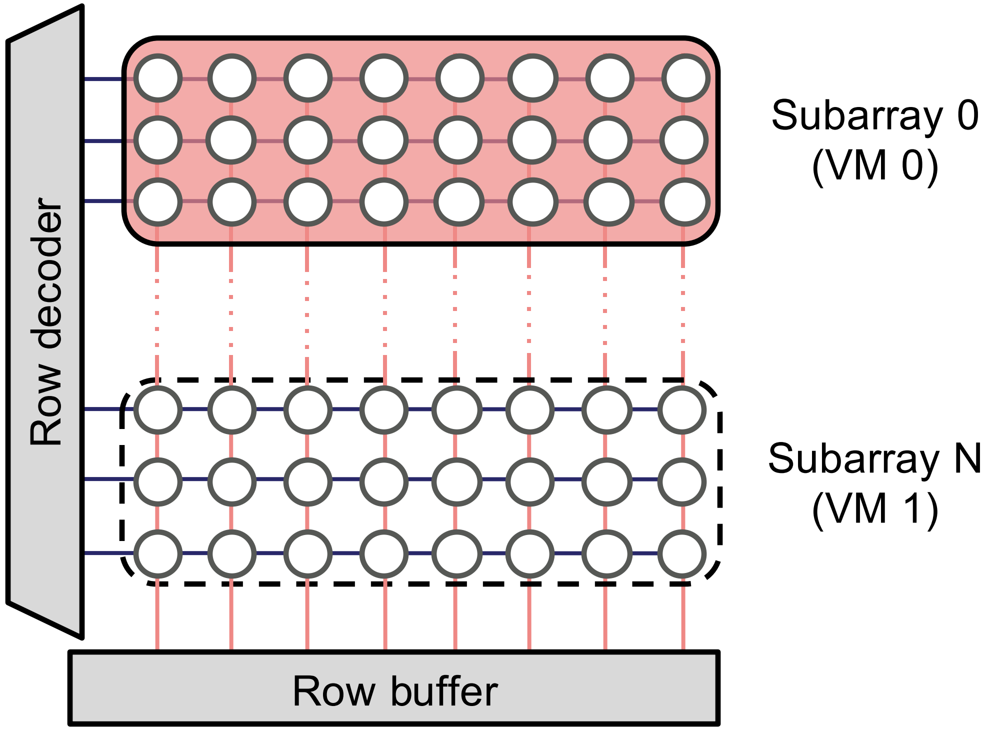
# Capabilities of Marionette

- Software-based mitigations
  - Isolation-based



# Capabilities of Marionette

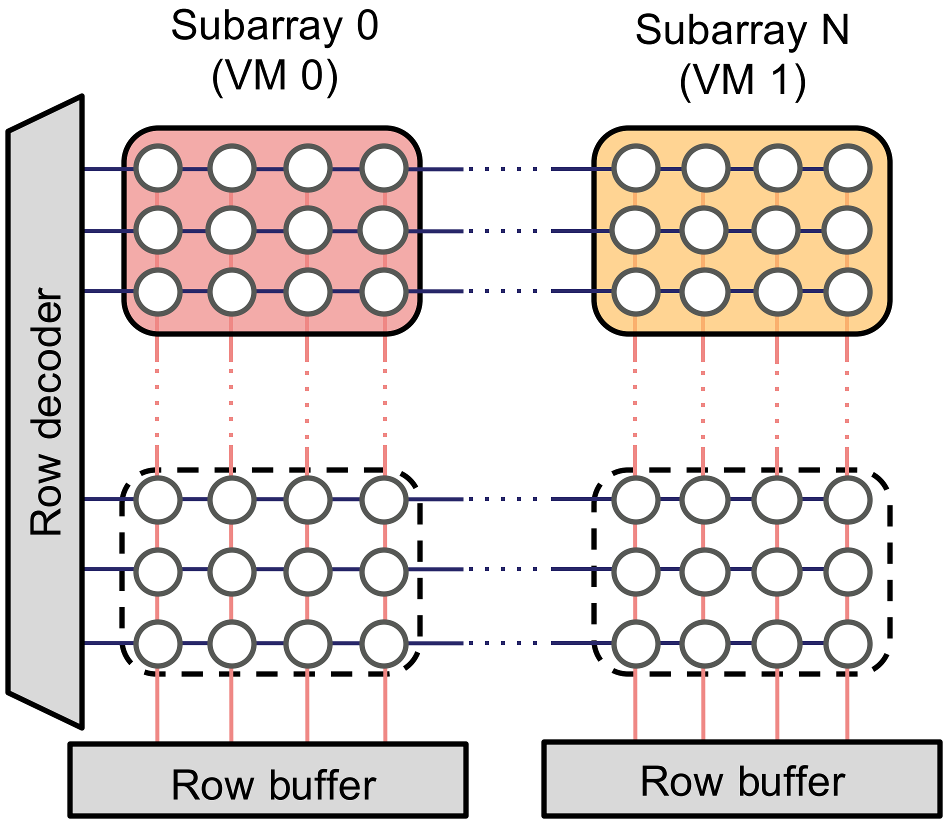
- Software-based mitigations
  - Isolation-based
    - E.g., Siloz[11]



[11] K. Loughlin et al., “Siloz: Leveraging DRAM Isolation Domains to Prevent Inter-VM Rowhammer,” SOSP, 2023.

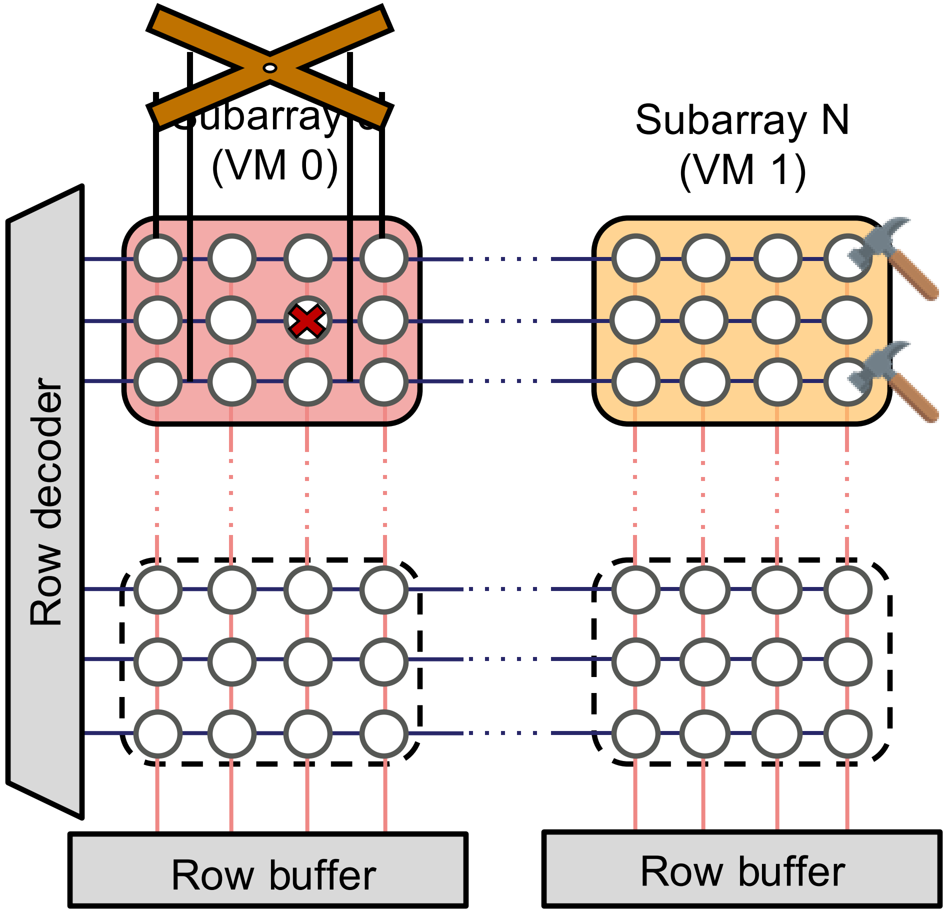
# Capabilities of Marionette

- Software-based mitigations
  - – Isolation-based
    - = E.g., Siloz



# Capabilities of Marionette

- Software-based mitigations
  - – Isolation-based
    - = E.g., Siloz



# Capabilities of Marionette

- Kernel privilege escalation[12]
  - – Exploiting the DRAM RowHammer bug to gain kernel privileges

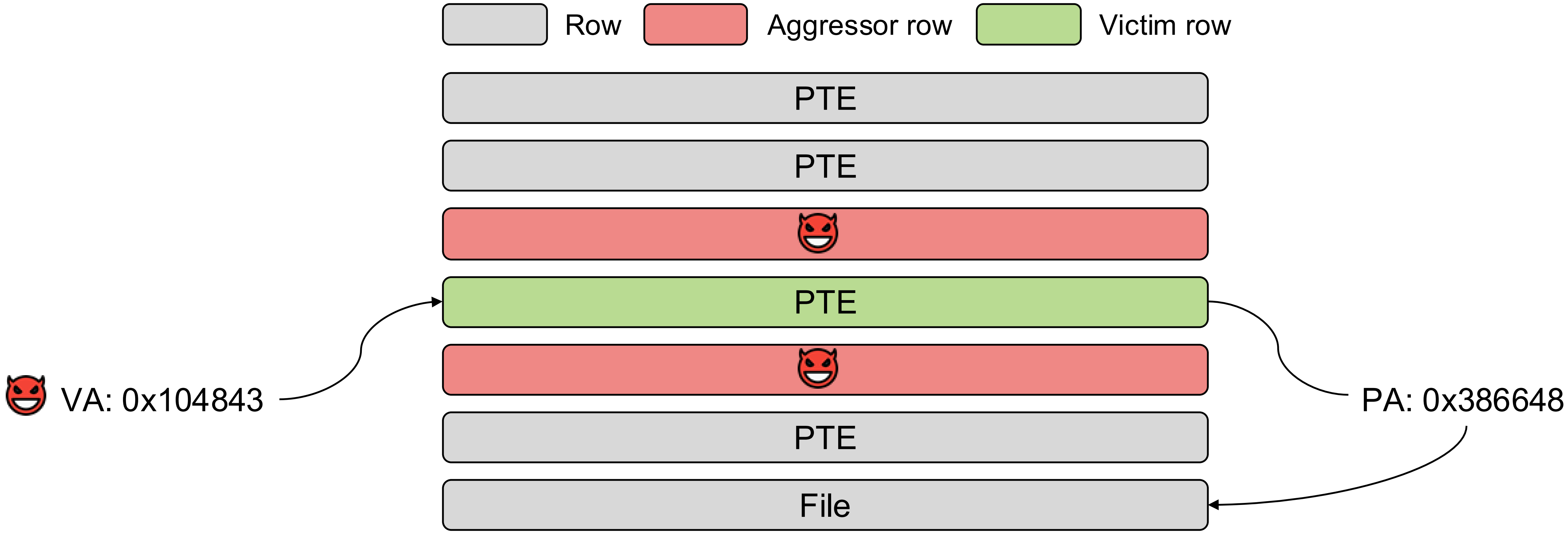
[12] M. Seaborn et al., “Exploiting the DRAM Rowhammer bug to gain kernel privileges,” https://googleprojectzero.blogspot.com /2015/03/ exploiting-dram-rowhammer-bug-to-gain.html, 2015.

# Capabilities of Marionette

- Kernel privilege escalation
  - Exploiting the DRAM RowHammer bug to gain kernel privileges
  - Attack flow
    - Step 1: Preparing Aggressor Rows
    - Step 2: Memory Templating
    - Step 3: Page Table Spraying
    - Step 4: RowHammer Attack

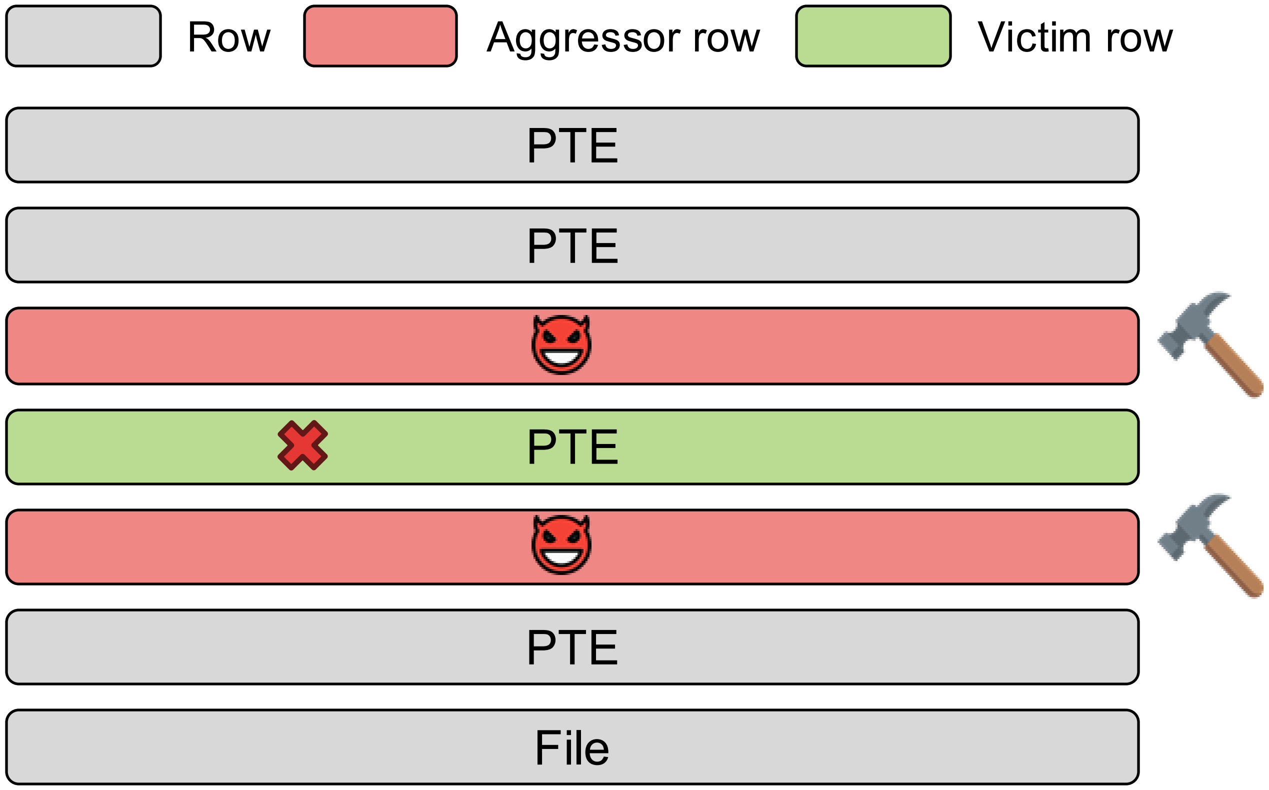
# Capabilities of Marionette

- Kernel privilege escalation



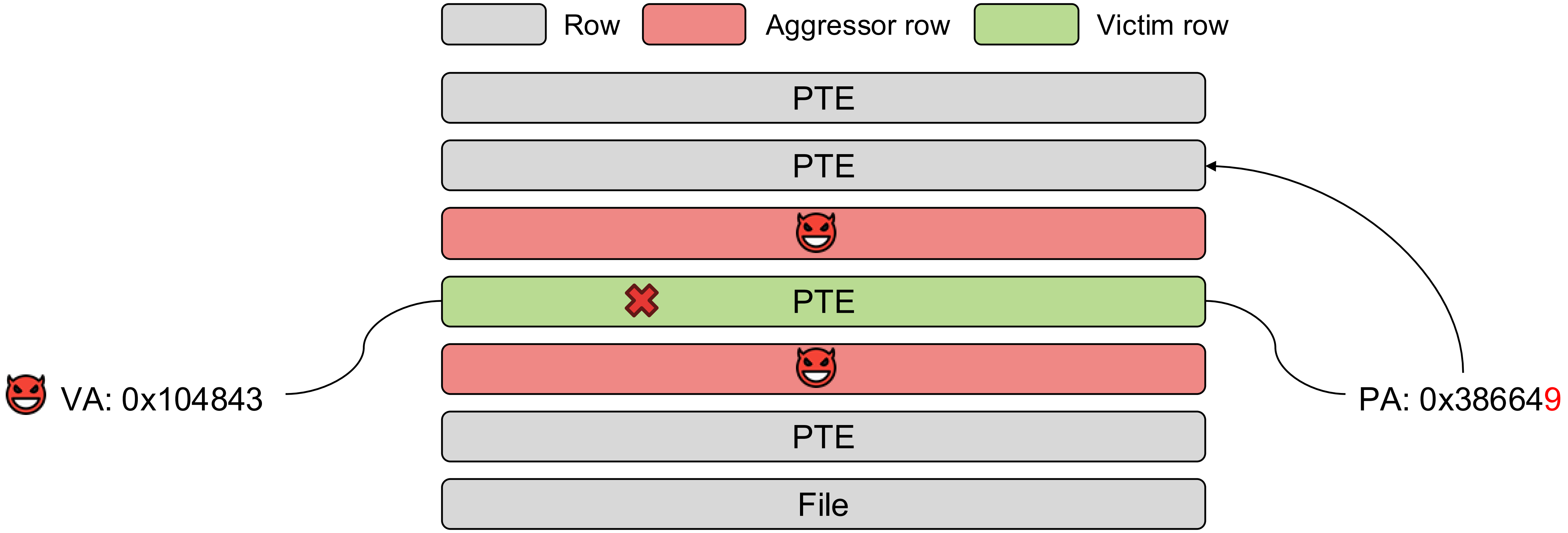
# Capabilities of Marionette

- Kernel privilege escalation



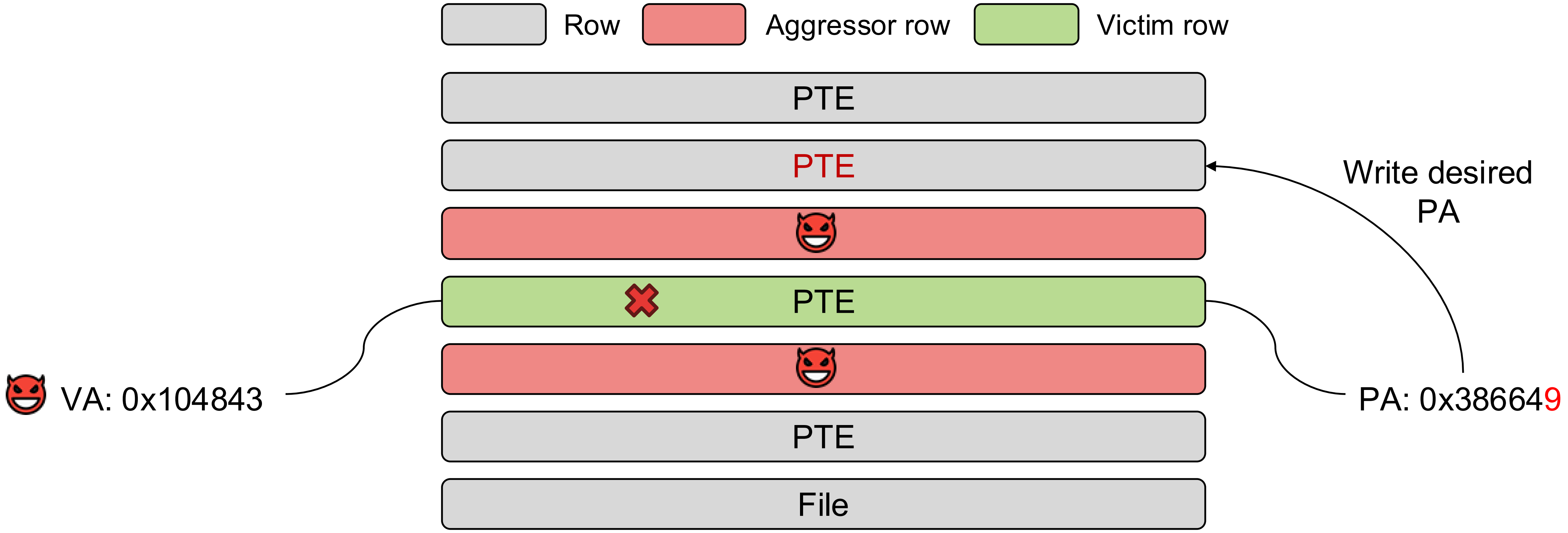
# Capabilities of Marionette

- Kernel privilege escalation



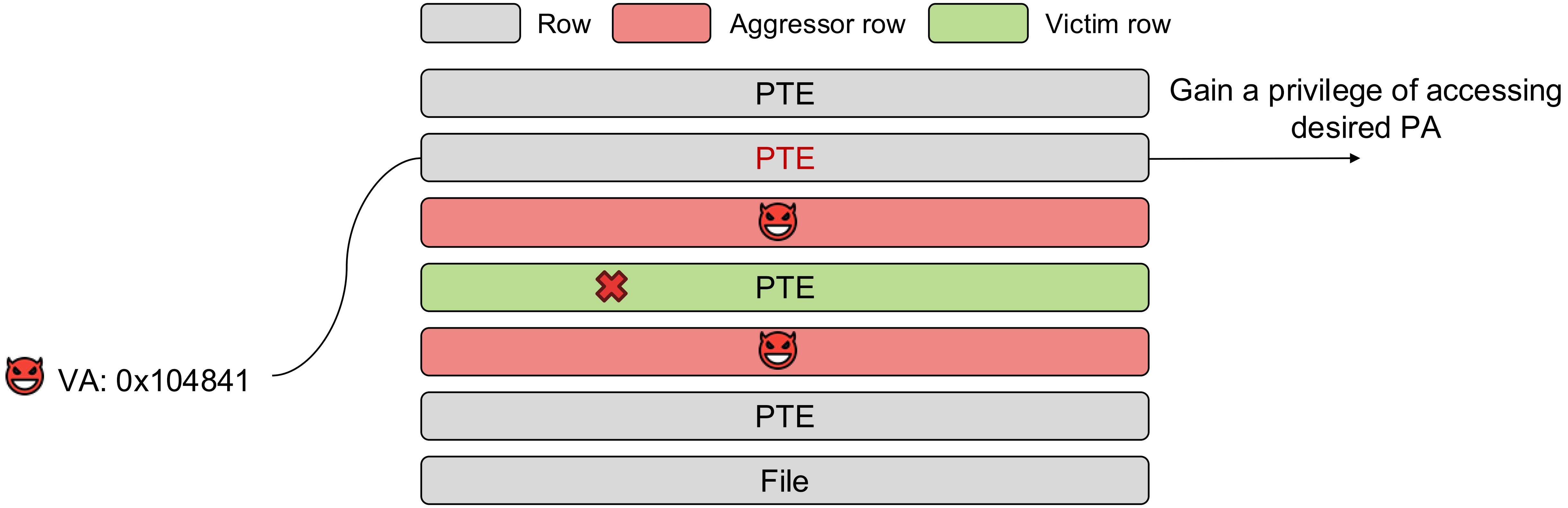
# Capabilities of Marionette

- Kernel privilege escalation



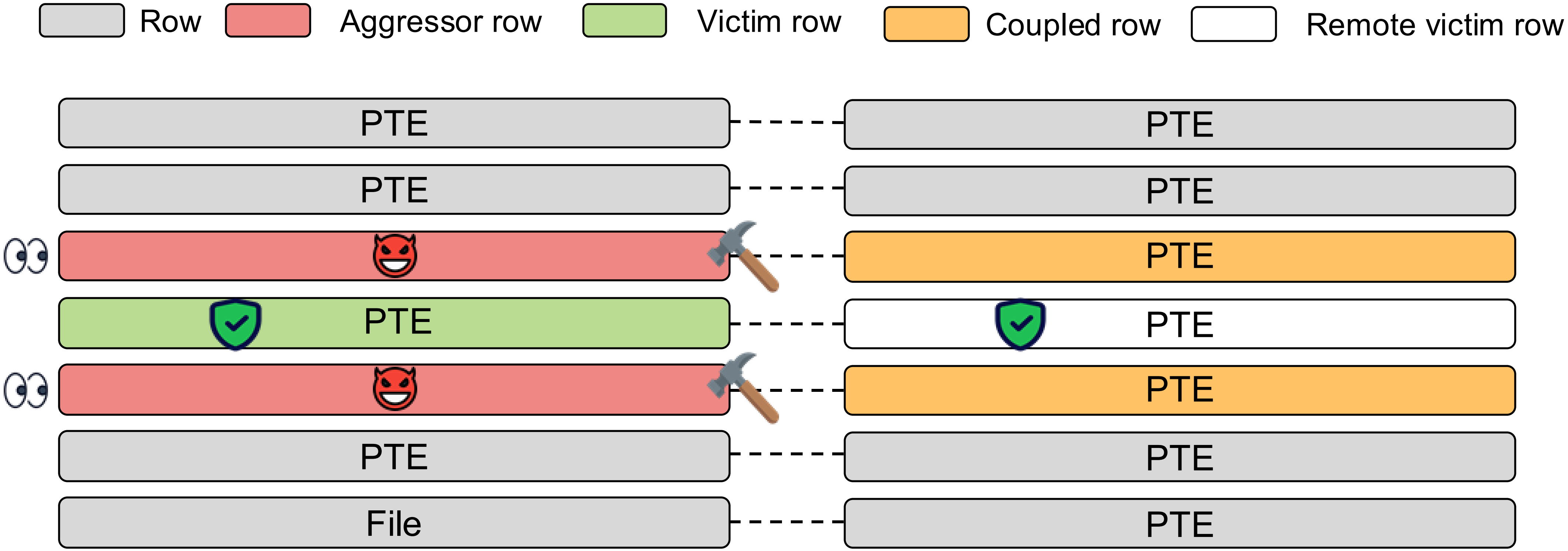
# Capabilities of Marionette

- Kernel privilege escalation



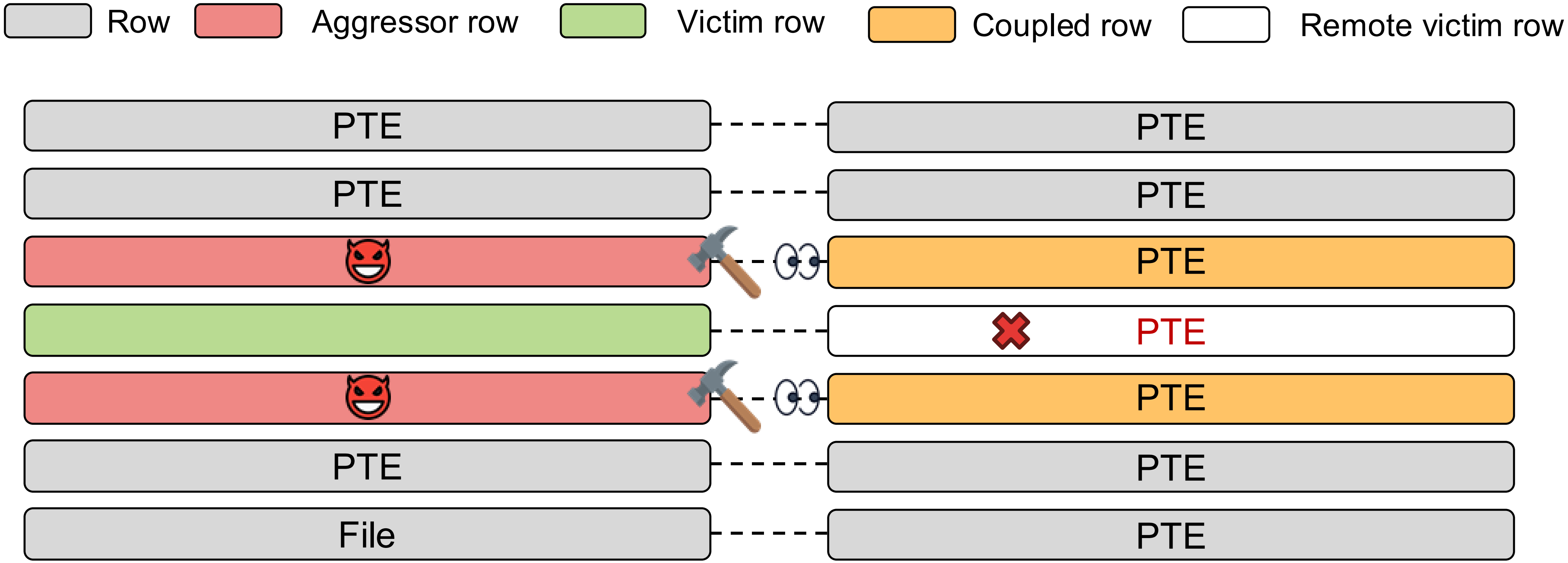
# Capabilities of Marionette

- Exploitation bypassing SoftTRR



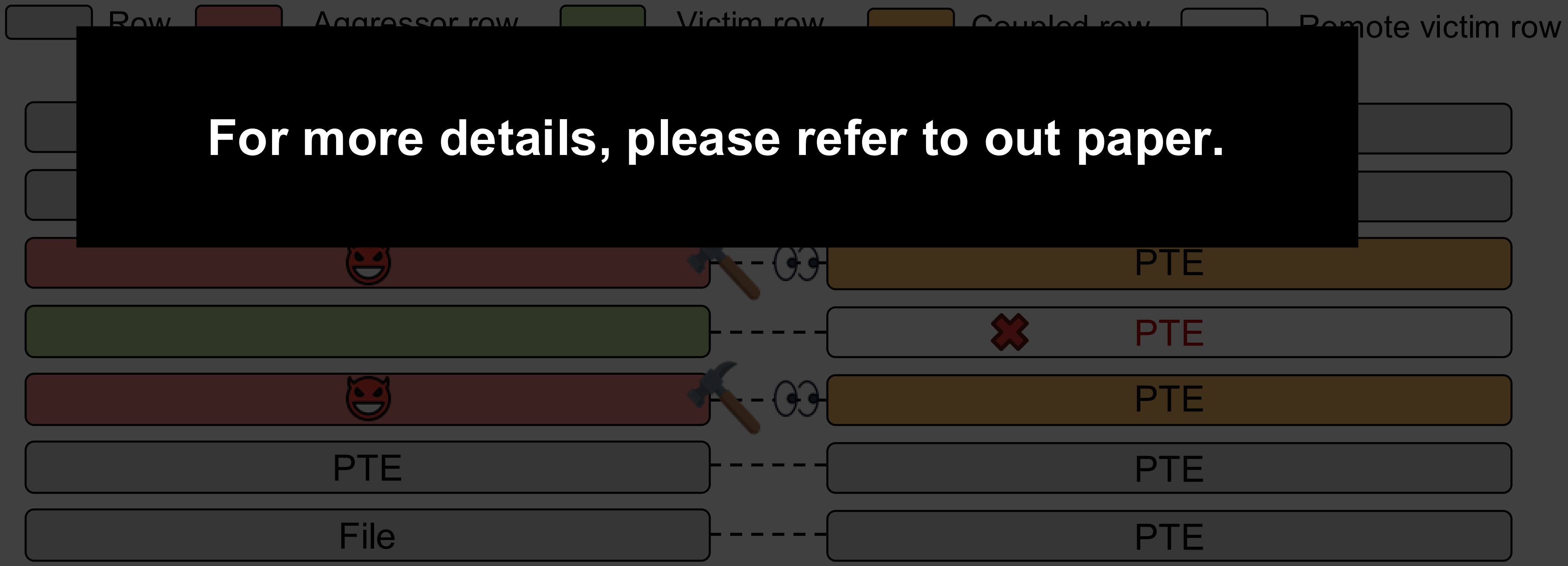
# Capabilities of Marionette

- Exploitation bypassing SoftTRR



# Capabilities of Marionette

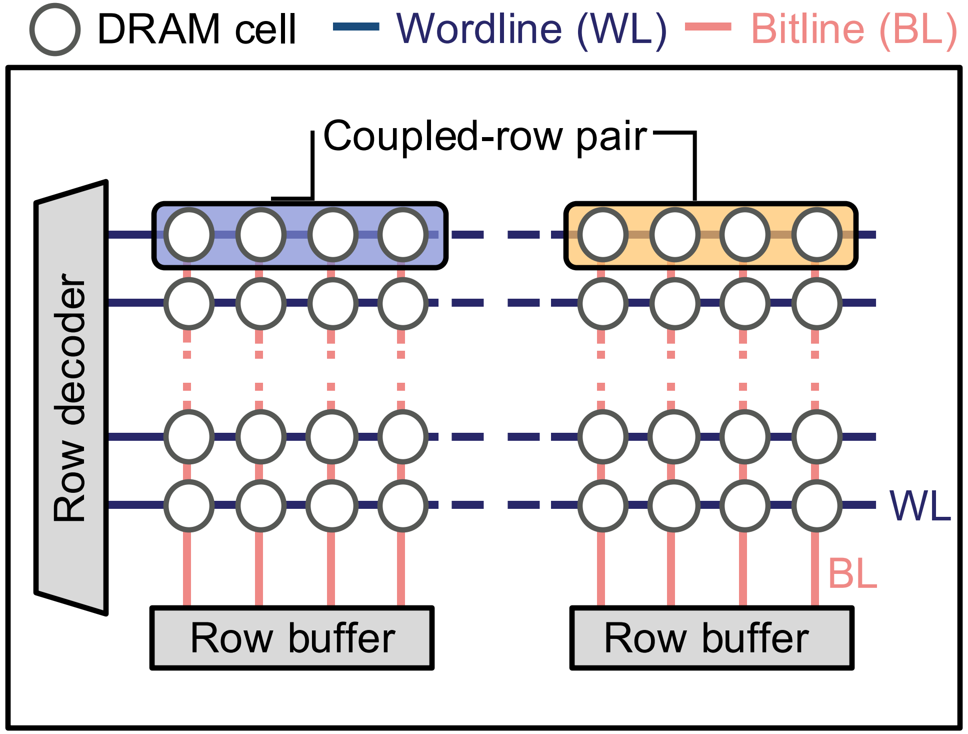
- Exploitation bypassing SoftTRR



**For more details, please refer to out paper.**

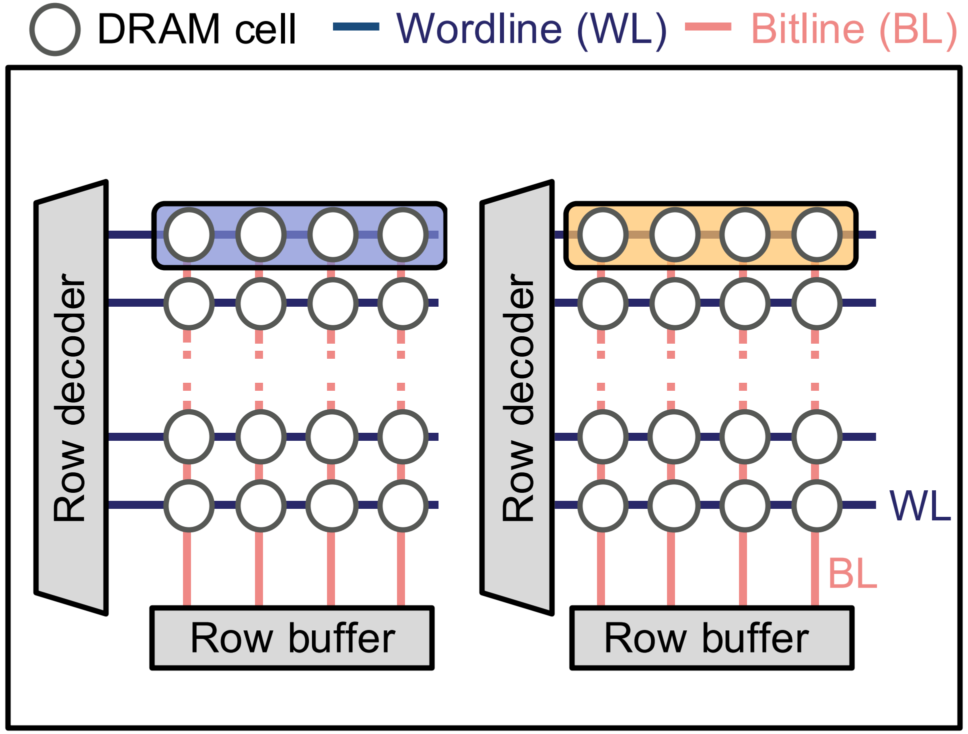
# Straightforward Marionette Mitigations

- Naïve approach
  - Close the gap between the system view and the DRAM view of coupled rows.



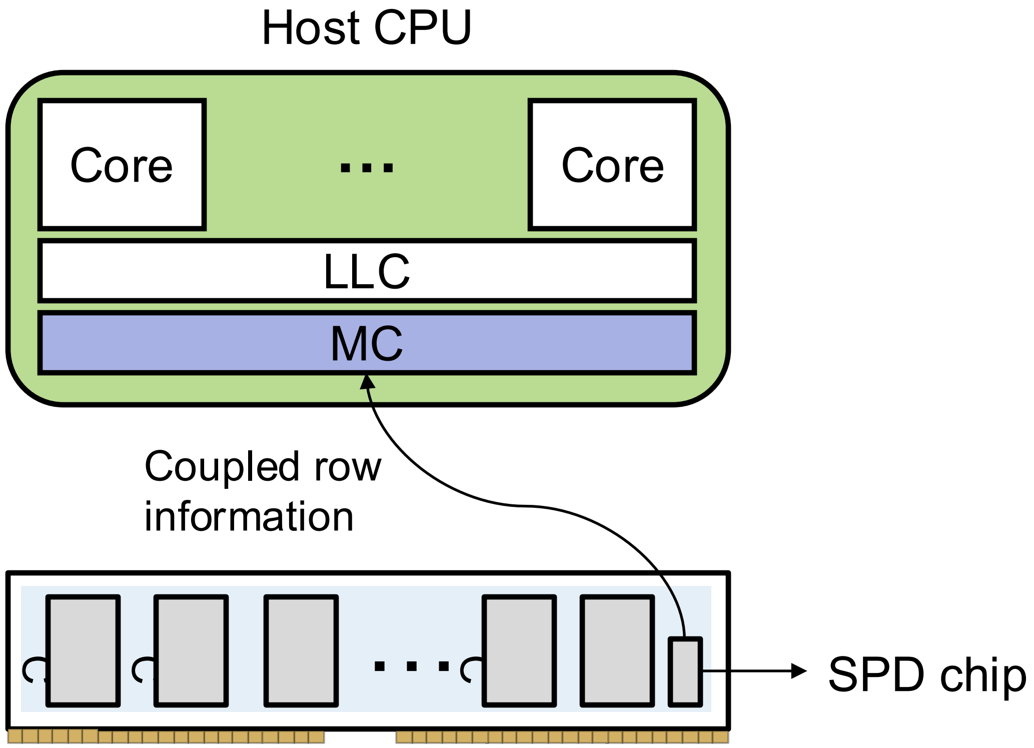
# Straightforward Marionette Mitigations

- Naïve approach
  - – Close the gap between the system view and the DRAM view of coupled rows.



# Straightforward Marionette Mitigations

- Expose a coupled row to the system
  - E.g., stores coupled row information in Serial Presence Detect (SPD) chips



# Summary

- Coupled row analysis
  - FPGA
    - = Coupled row has the hammering strength similar to conventional RowHammer attack
    - = In-DRAM TRR properly tracks coupled rows
  - System
    - = There is a distant gap between coupled rows in physical address space

# Summary

- Coupled row analysis
  - FPGA
    - = Coupled row has the hammering strength similar to conventional RowHammer attack
    - = In-DRAM TRR properly tracks coupled rows
  - System
    - = There is a distant gap between coupled rows in physical address space
- **Marionette**
  - We demonstrate that Marionette can bypass an existing software-based RowHammer mitigations.
  - We showcase an exploitation using Marionette under a SoftTRR-protected system.
  - We discuss simple yet effective patches against Marionette

# Thank you!