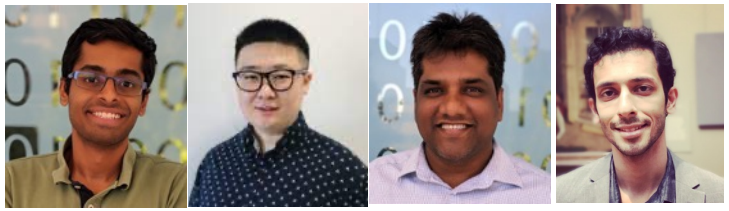
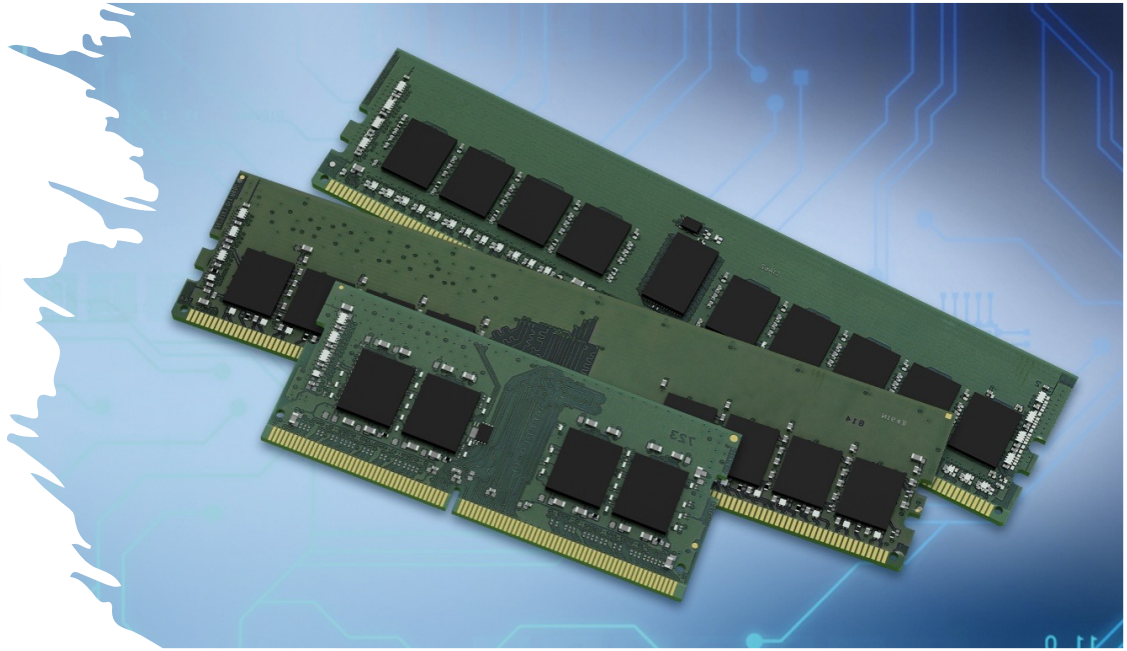
Randomized Row-Swap: Mitigating Row Hammer By Breaking Spatial Correlation Between Aggressor and Victim Rows

Gururaj Saileshwar

Bolin Wang, Moinuddin Qureshi,

Prashant Nair



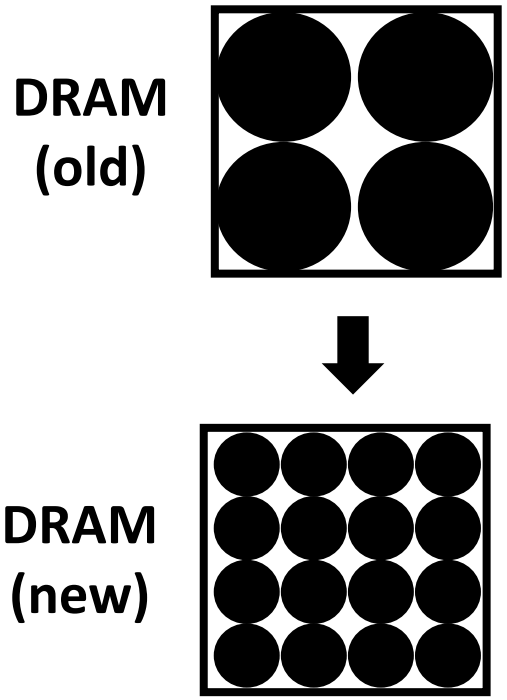


Georgia Tech

UBC THE UNIVERSITY OF BRITISH COLUMBIA

Rowhammer Attacks on DRAM

DRAM Scaling for Increased Capacity

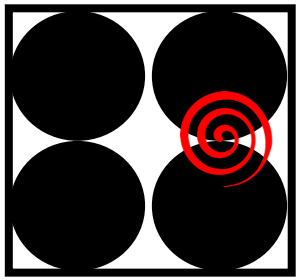


Rowhammer Attacks on DRAM

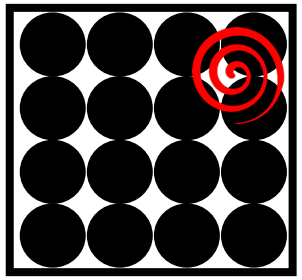
DRAM Scaling for Increased Capacity

More Inter-Cell Interference

DRAM (old)



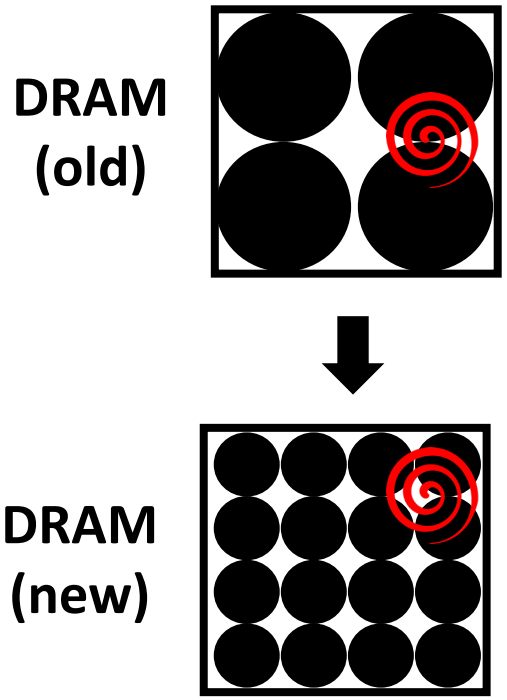
DRAM (new)

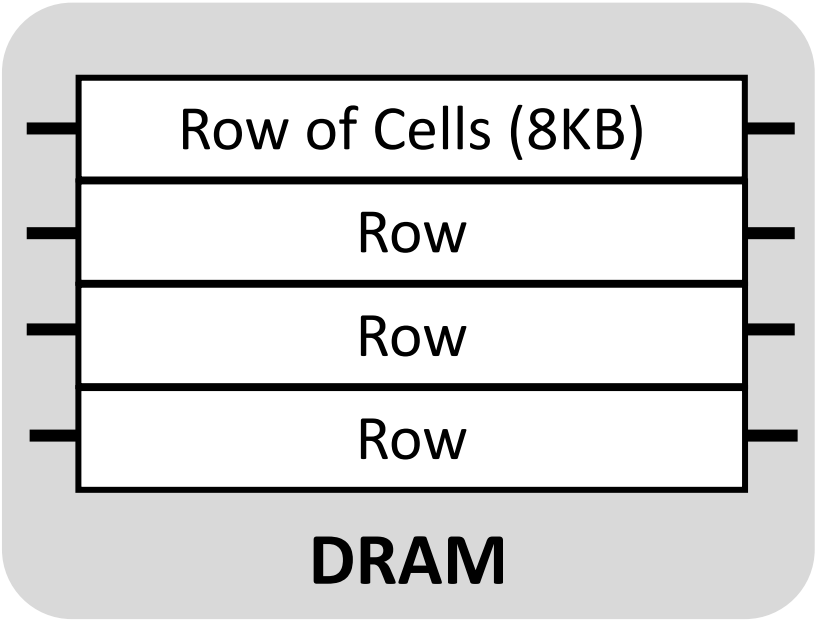


Rowhammer Attacks on DRAM

DRAM Scaling for Increased Capacity

More Inter-Cell Interference

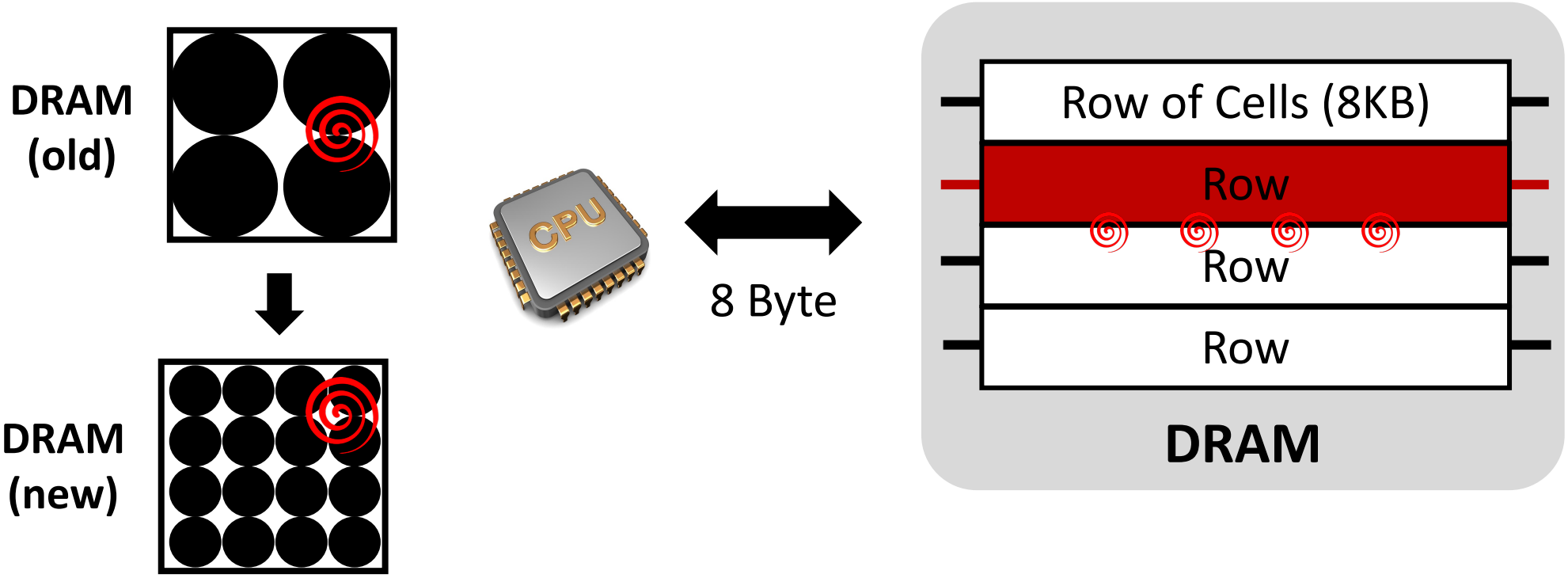




Rowhammer Attacks on DRAM

DRAM Scaling for Increased Capacity

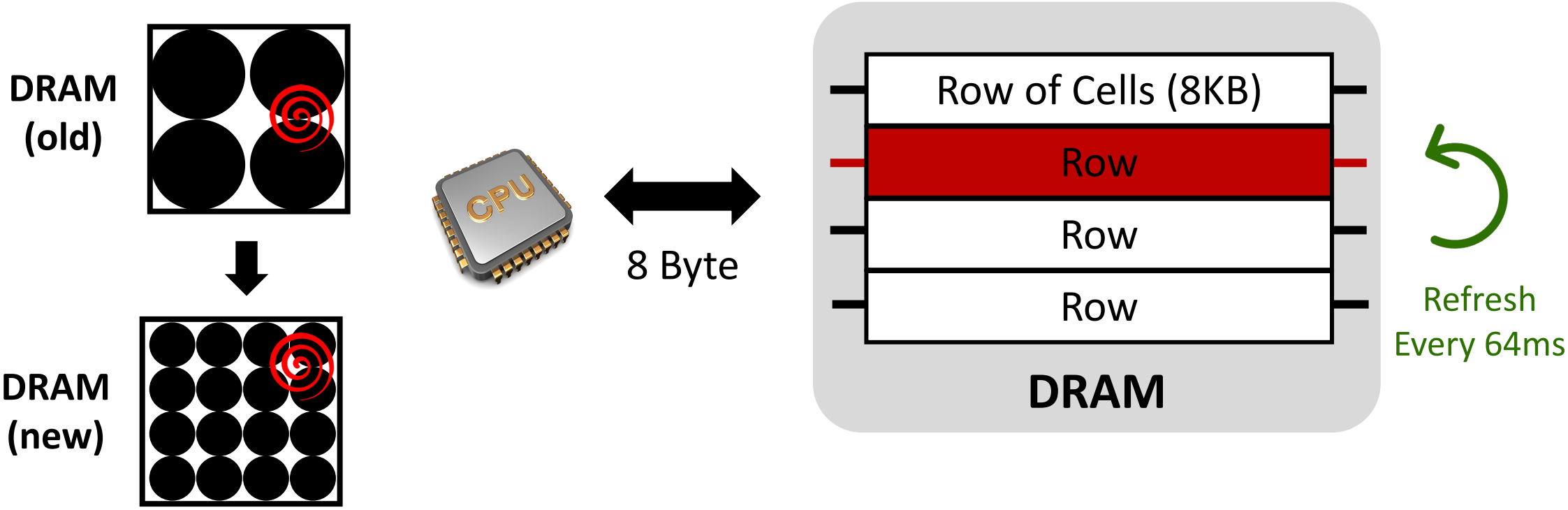
More Inter-Cell Interference



Rowhammer Attacks on DRAM

DRAM Scaling for Increased Capacity

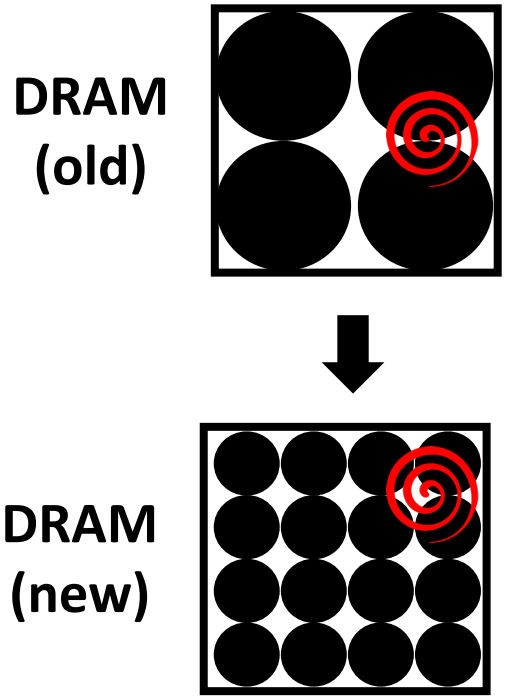
More Inter-Cell Interference

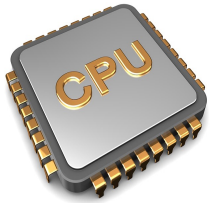


Rowhammer Attacks on DRAM

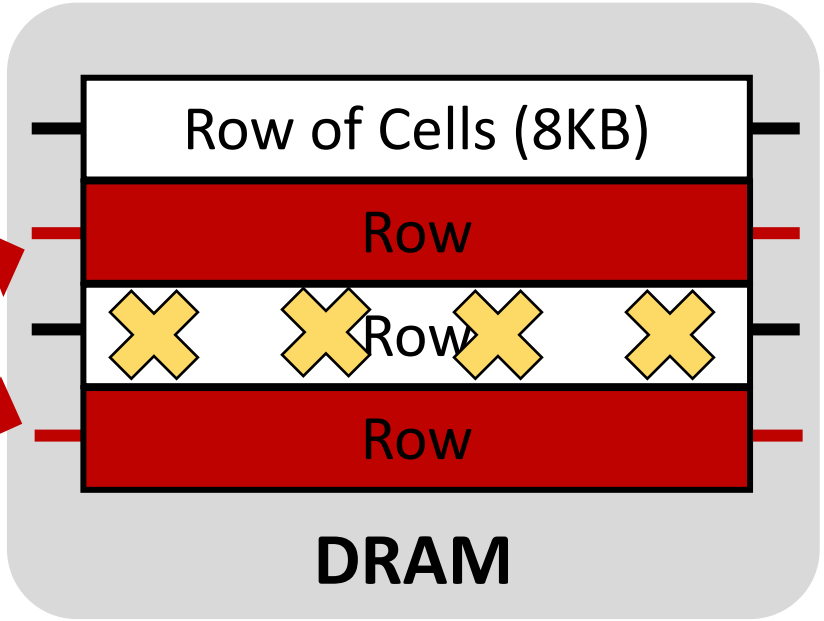
DRAM Scaling for Increased Capacity

More Inter-Cell Interference





Rapid Accesses



Rowhammer Attack

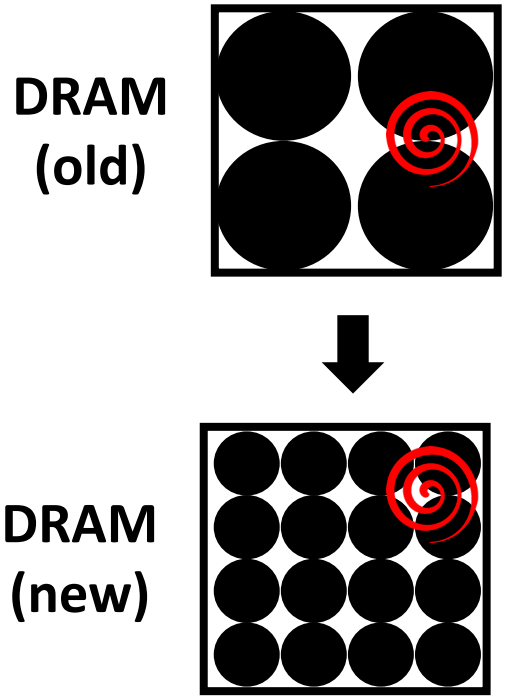
Bit-Flips in Neighboring Rows

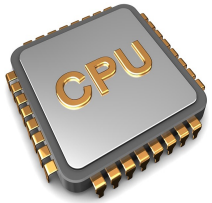
[Kim+, ISCA'15]

Rowhammer Attacks on DRAM

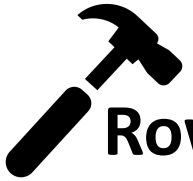
DRAM Scaling for Increased Capacity

More Inter-Cell Interference

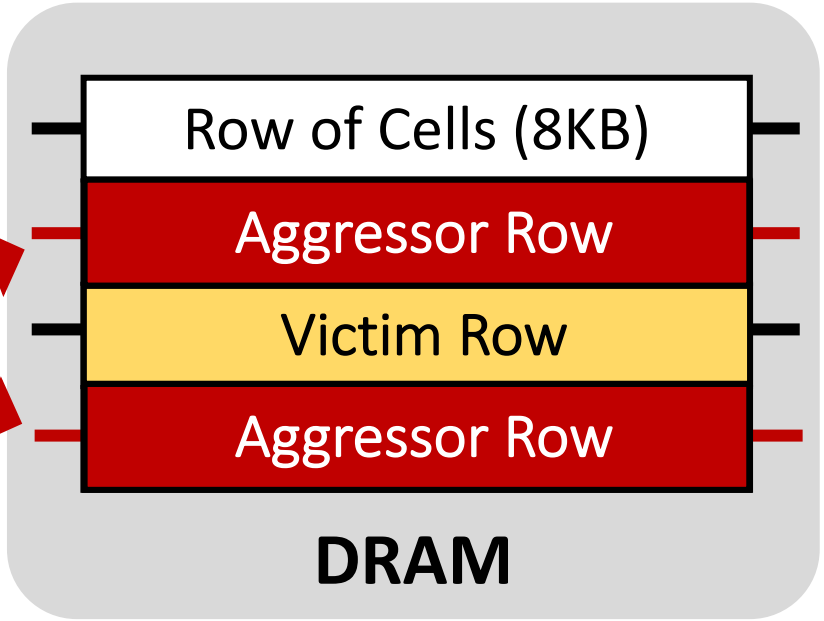




Rapid Accesses



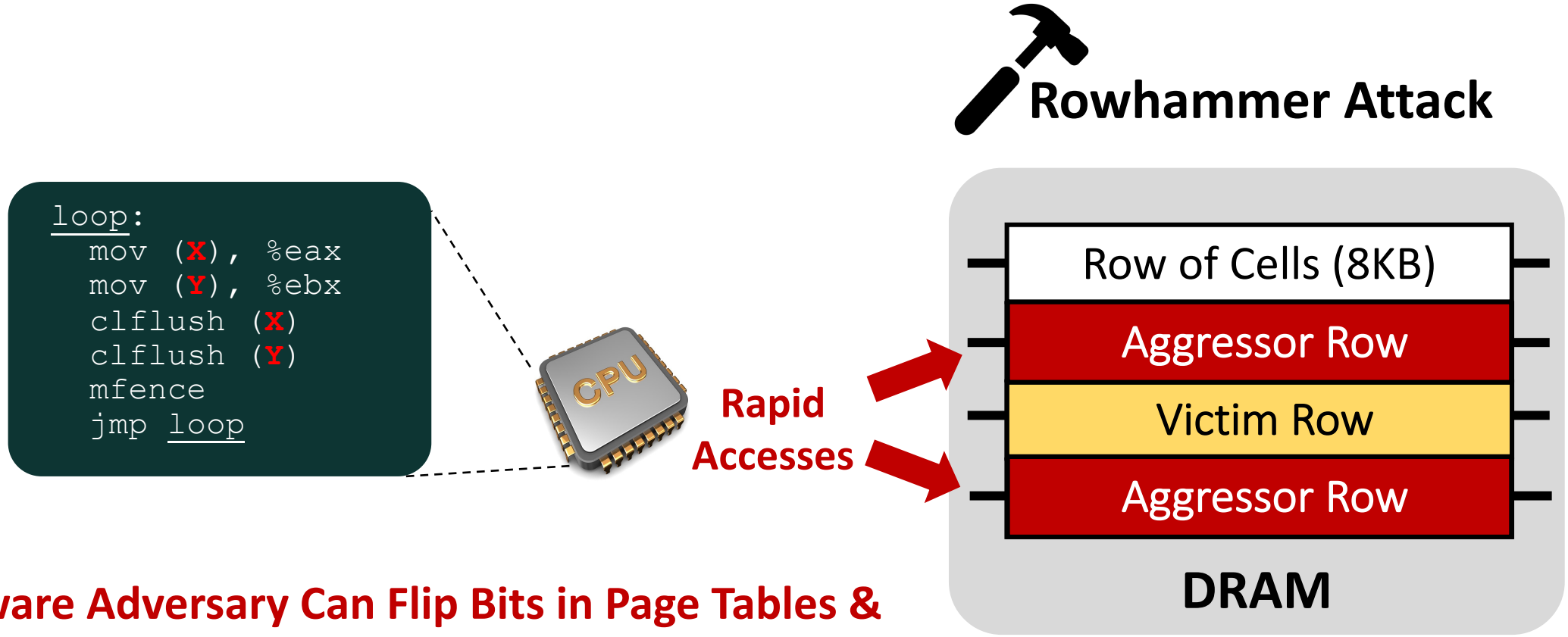
Rowhammer Attack



Bit-Flips in Neighboring Rows

[Kim+, ISCA'15]

Rowhammer Attacks on DRAM

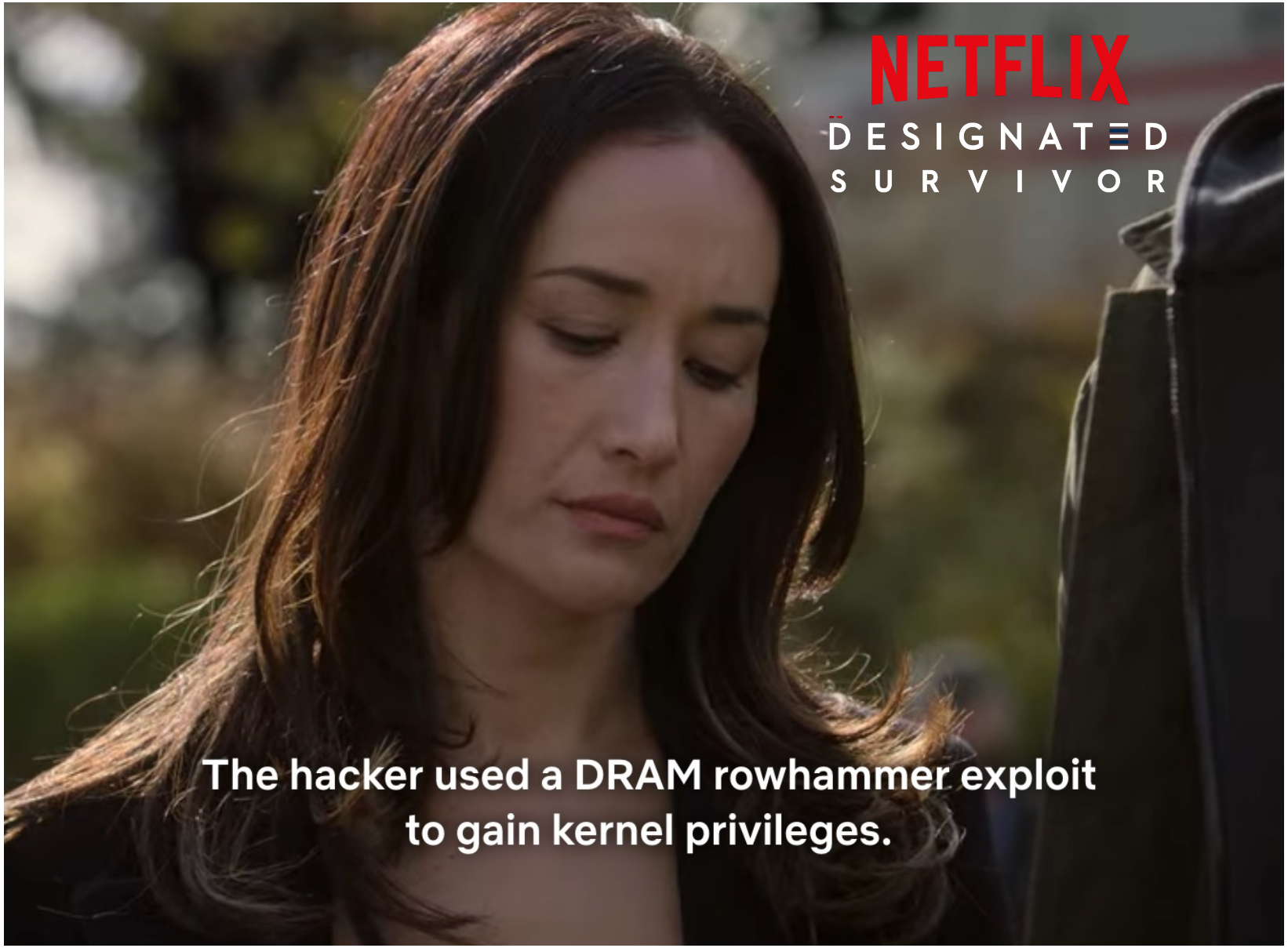


Software Adversary Can Flip Bits in Page Tables & Gain Kernel Privileges (Take Over a System)

[Seaborn+, Blackhat'15]

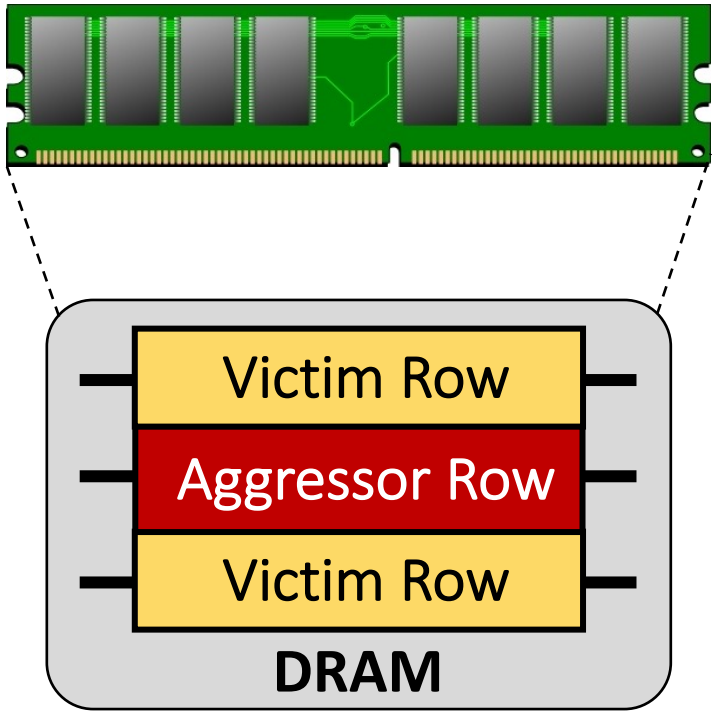
Bit-Flips in Neighboring Rows

[Kim+, ISCA'15]


NETFLIX
DESIGNATED
SURVIVOR
The hacker used a DRAM rowhammer exploit
to gain kernel privileges.

Mitigation in Commercial DDR4

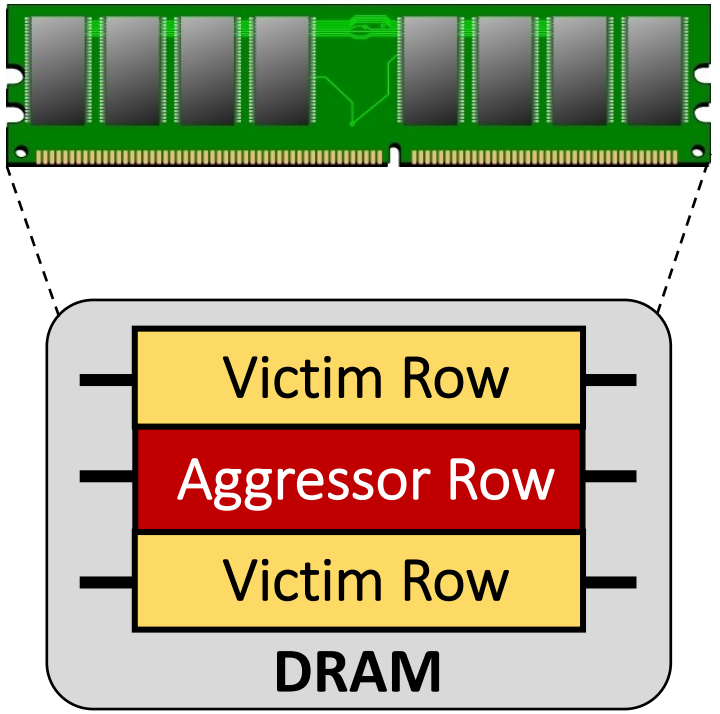
Targeted Row Refresh (TRR) in DDR4 (2015)



Mitigation in Commercial DDR4

Targeted Row Refresh (TRR) in DDR4 (2015)

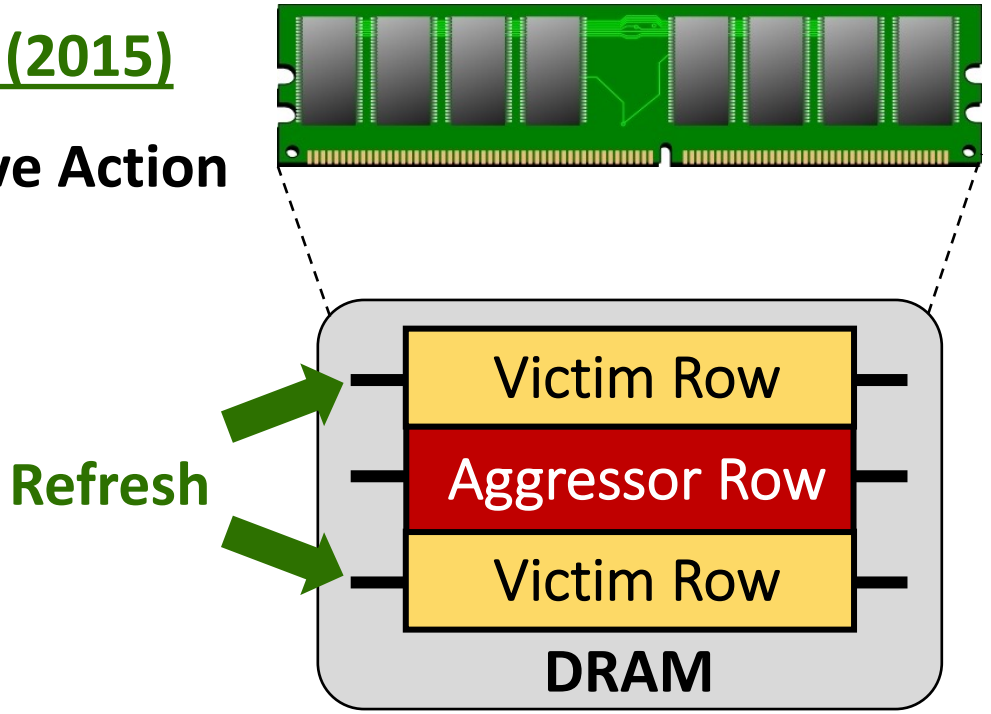
1. **Track Aggressor Rows**



Mitigation in Commercial DDR4

Targeted Row Refresh (TRR) in DDR4 (2015)

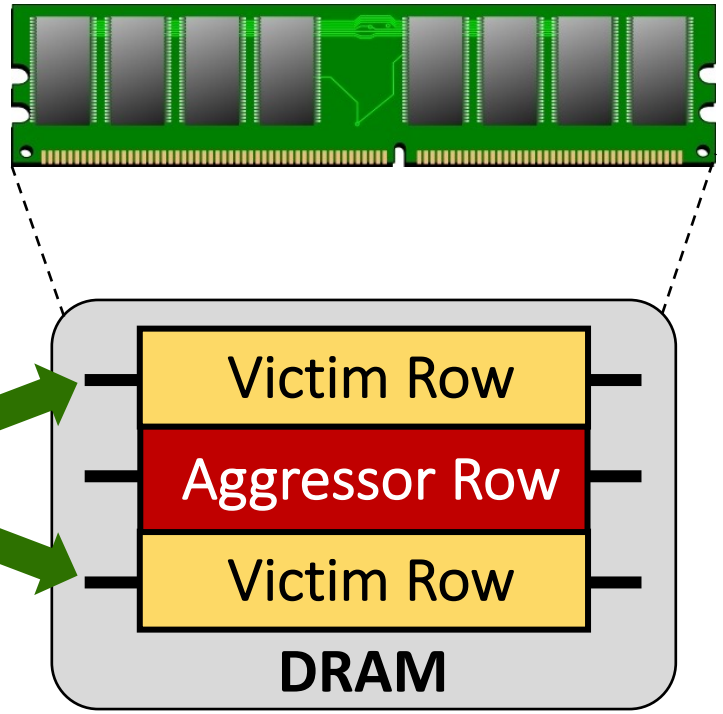
❶ Track Aggressor Rows ❷ Mitigative Action



Mitigation in Commercial DDR4: Broken!

Targeted Row Refresh (TRR) in DDR4 (2014)

❶ ~~Track Aggressor Rows~~ ❷ Mitigative Action



TRResspass Breaks TRR Tracker [Frigo+, SP'20]

> **Poor Rowhammer Fixes On DDR4 DRAM Chips Re-Enable Bit Flipping Attacks**
>
> Source: The Hacker News

Blacksmith Attack: All DRAM Shown Vulnerable [Jattke+, SP'22]

> **When the world ends, all that will be left are cockroaches and new Rowhammer attacks: RAM defenses broken again**
>
> Blacksmith is latest hammer horror
>
> Source: The Register

Recent Victim Focused Mitigations

✓ Track Aggressor Rows ❷ Mitigative Action

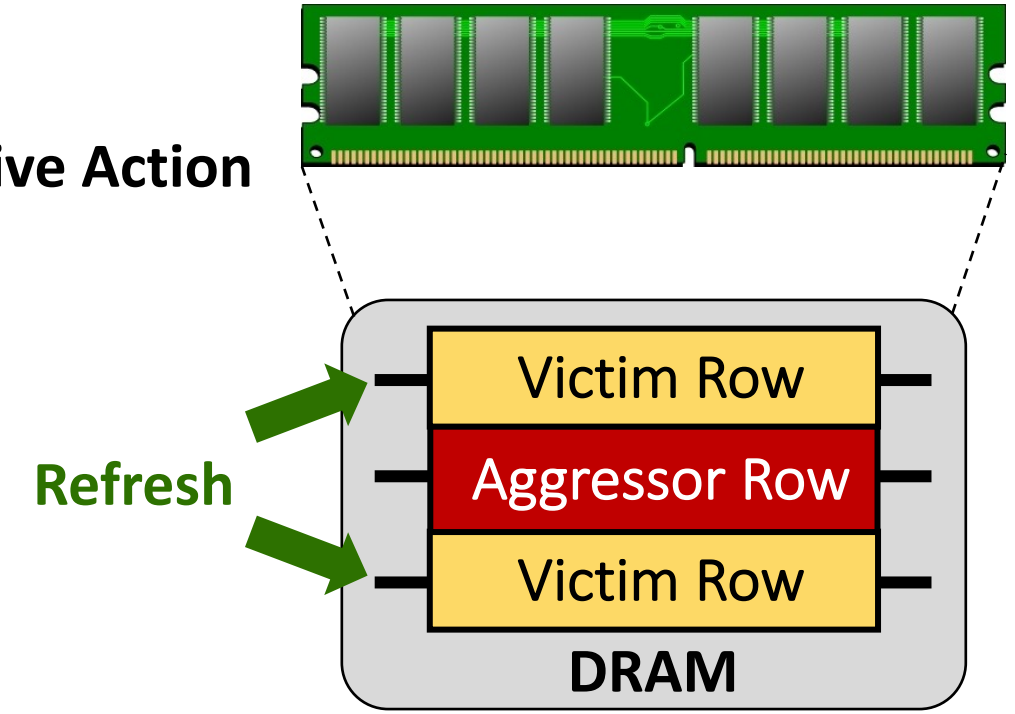
Principled Tracking

Graphene [MICRO'20]

TWICE [ISCA'19]

CRA [CAL'14]

and others …

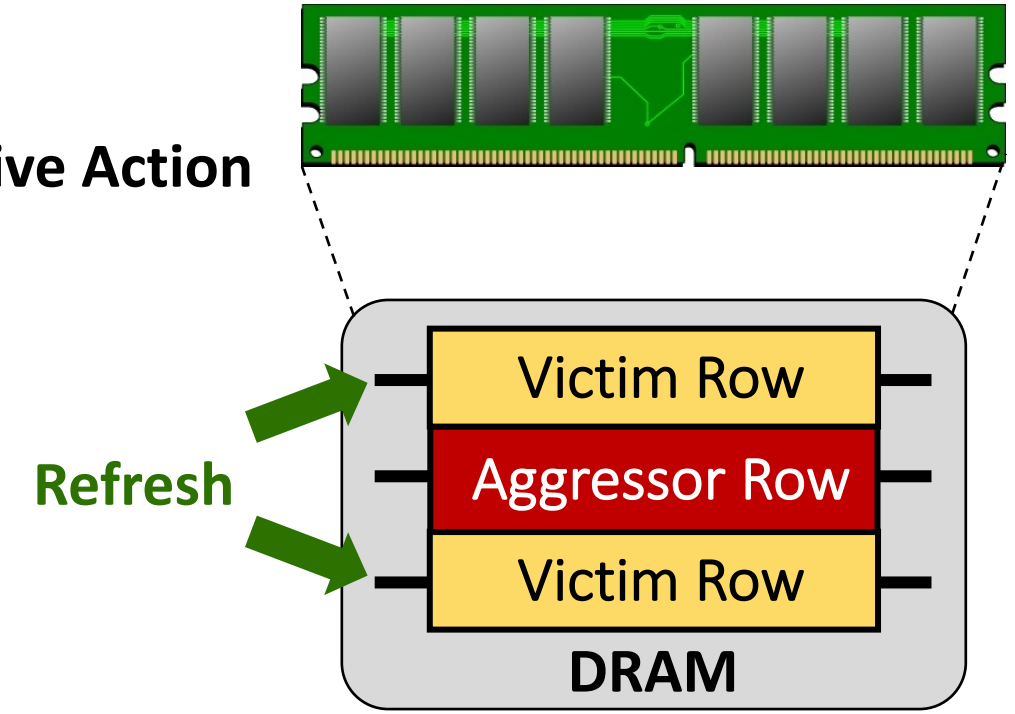


Recent Victim Focused Mitigations

Track Aggressor Rows

❷ Mitigative Action

?



Arms Race in Rowhammer

Track Aggressor Rows ❷ Mitigative Action

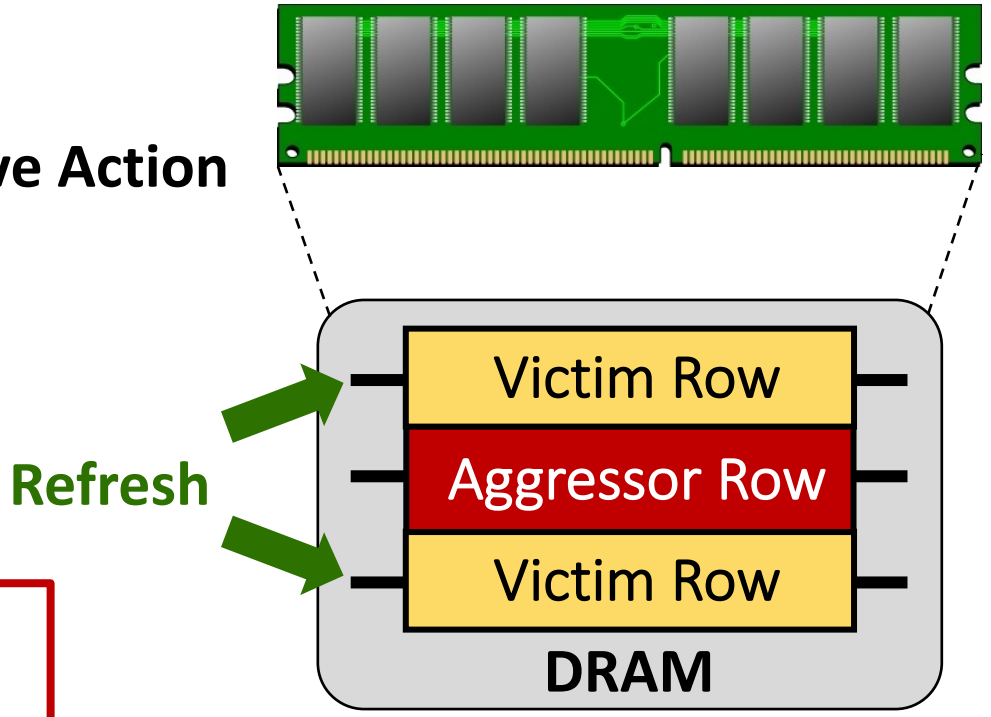
?

Google's Half-Double Attack: Exploits Mitigative Refresh [2021]

As Chips Shrink, Rowhammer Attacks Get Harder to Stop

A full fix for the "Half-Double" technique will require rethinking how memory semiconductors are designed.

Source: ArsTechnica



Arms Race in Rowhammer

Track Aggressor Rows ❷ Mitigative Action

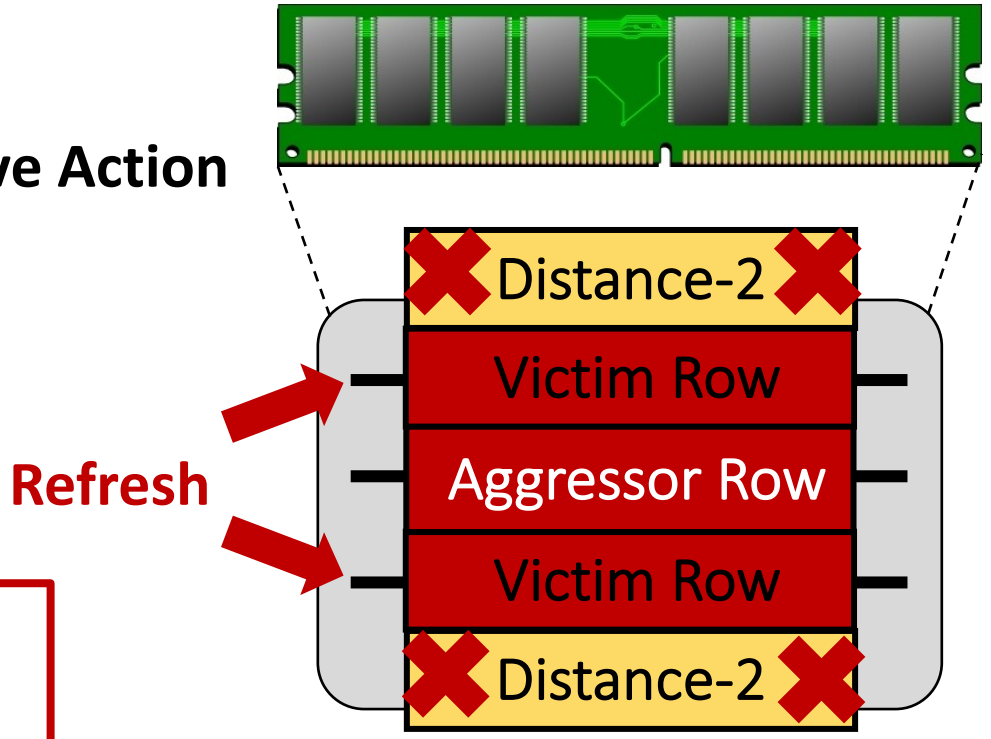
?

Google's Half-Double Attack: Exploits Mitigative Refresh [2021]

As Chips Shrink, Rowhammer Attacks Get Harder to Stop

A full fix for the "Half-Double" technique will require rethinking how memory semiconductors are designed.

Source: ArsTechnica



Arms Race in Rowhammer

Track Aggressor Rows ❷ Mitigative Action

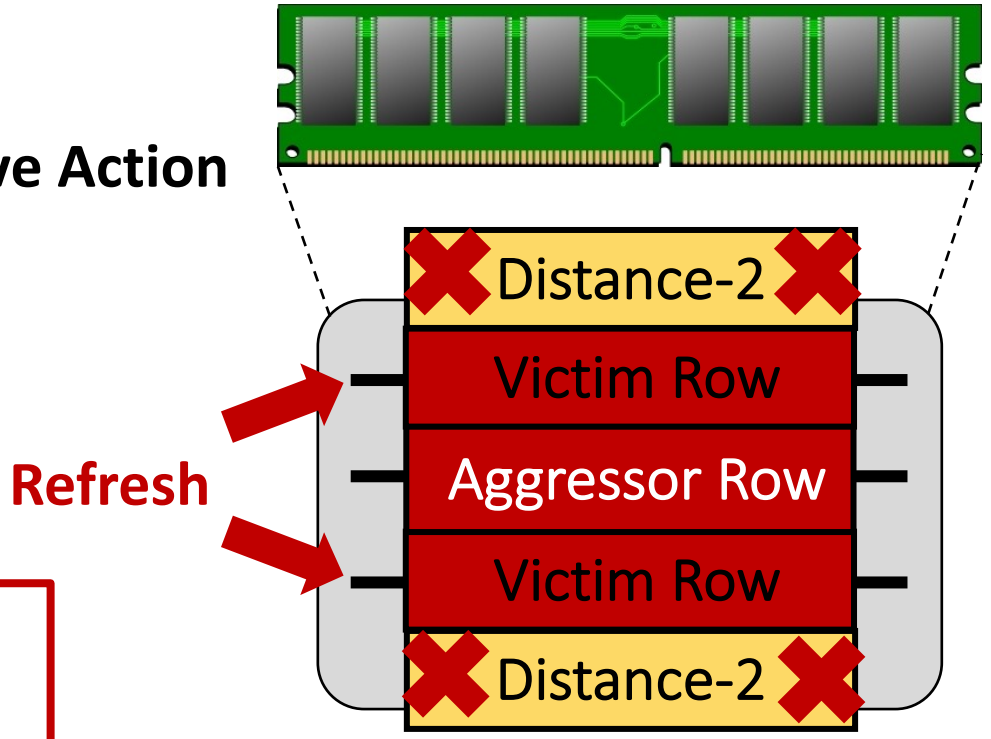
?

Google's Half-Double Attack: Exploits Mitigative Refresh [2021]

As Chips Shrink, Rowhammer Attacks Get Harder to Stop

A full fix for the "Half-Double" technique will require rethinking how memory semiconductors are designed.

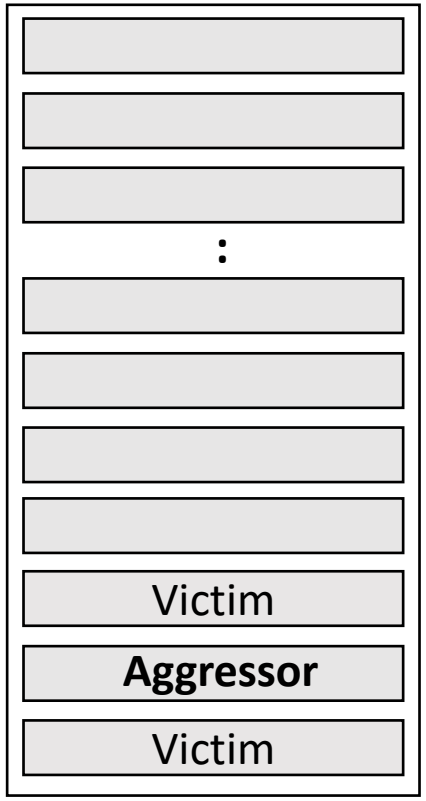
Source: ArsTechnica



Need New Mitigative Action Resilient to Current and Emerging Attack Patterns (preferably without requiring knowledge of DRAM mapping function)

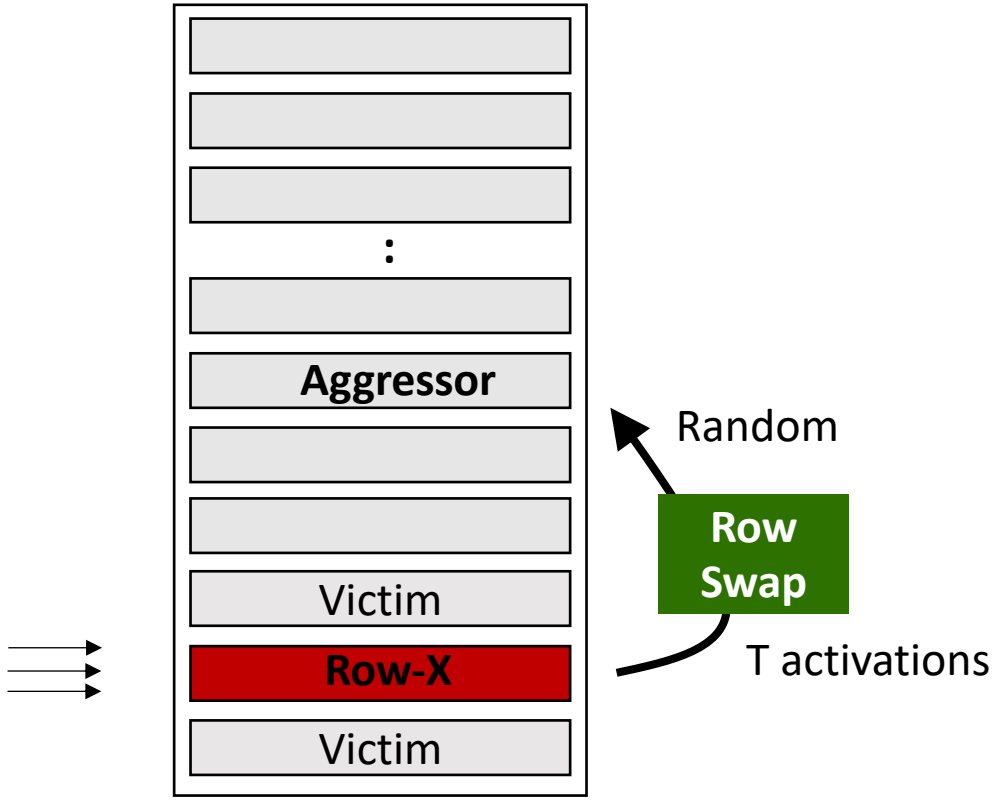
Aggressor Focused Mitigation: Randomized Row-Swap

Key Idea: Remap Aggressor Rows to Break Spatial Correlation with Victim Rows



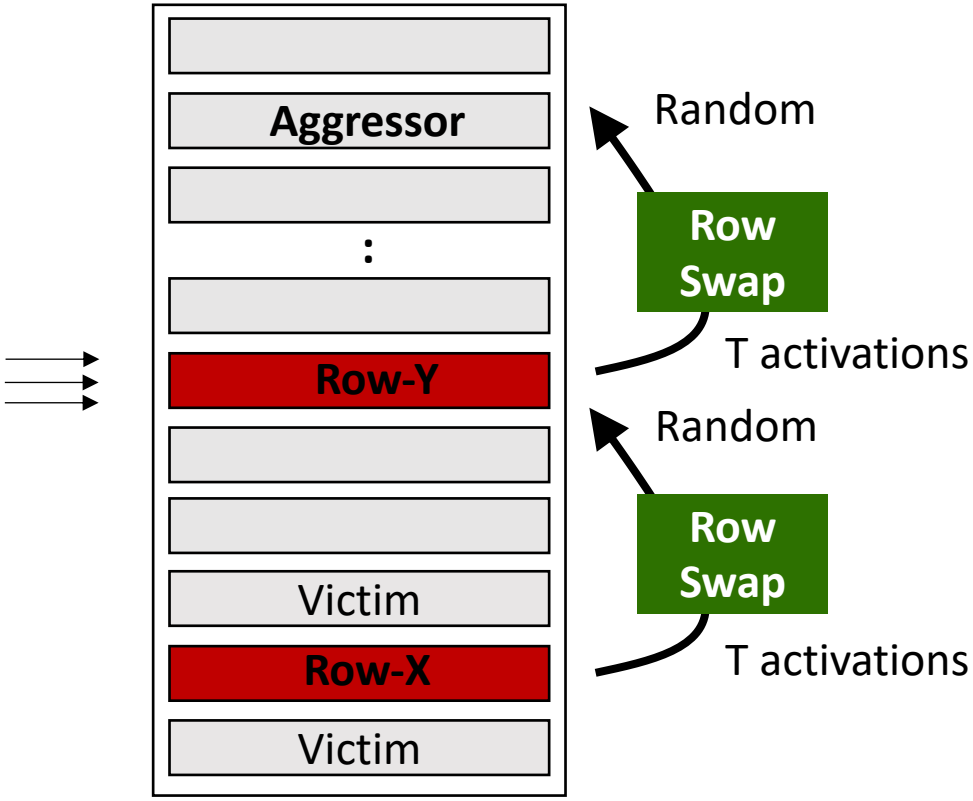
Aggressor Focused Mitigation: Randomized Row-Swap

Key Idea: Remap Aggressor Rows to Break Spatial Correlation with Victim Rows



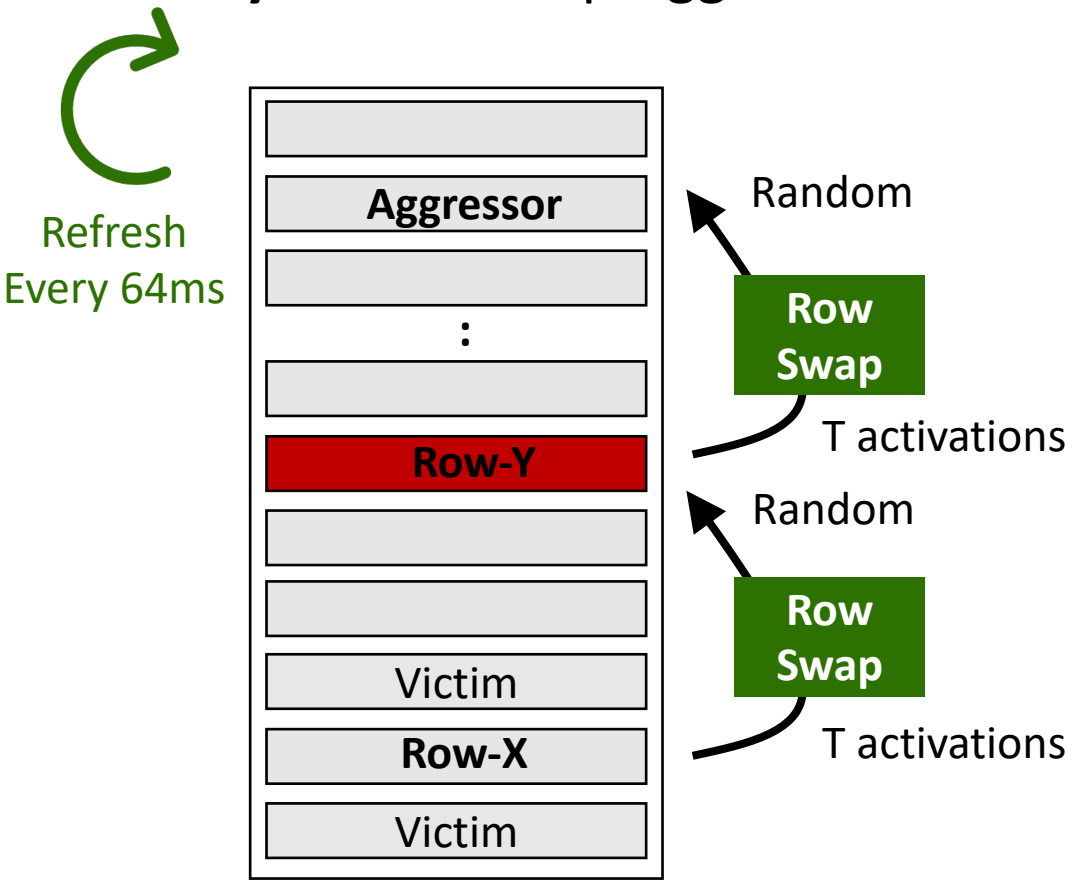
Aggressor Focused Mitigation: Randomized Row-Swap

Key Idea: Remap Aggressor Rows to Break Spatial Correlation with Victim Rows



Aggressor Focused Mitigation: Randomized Row-Swap

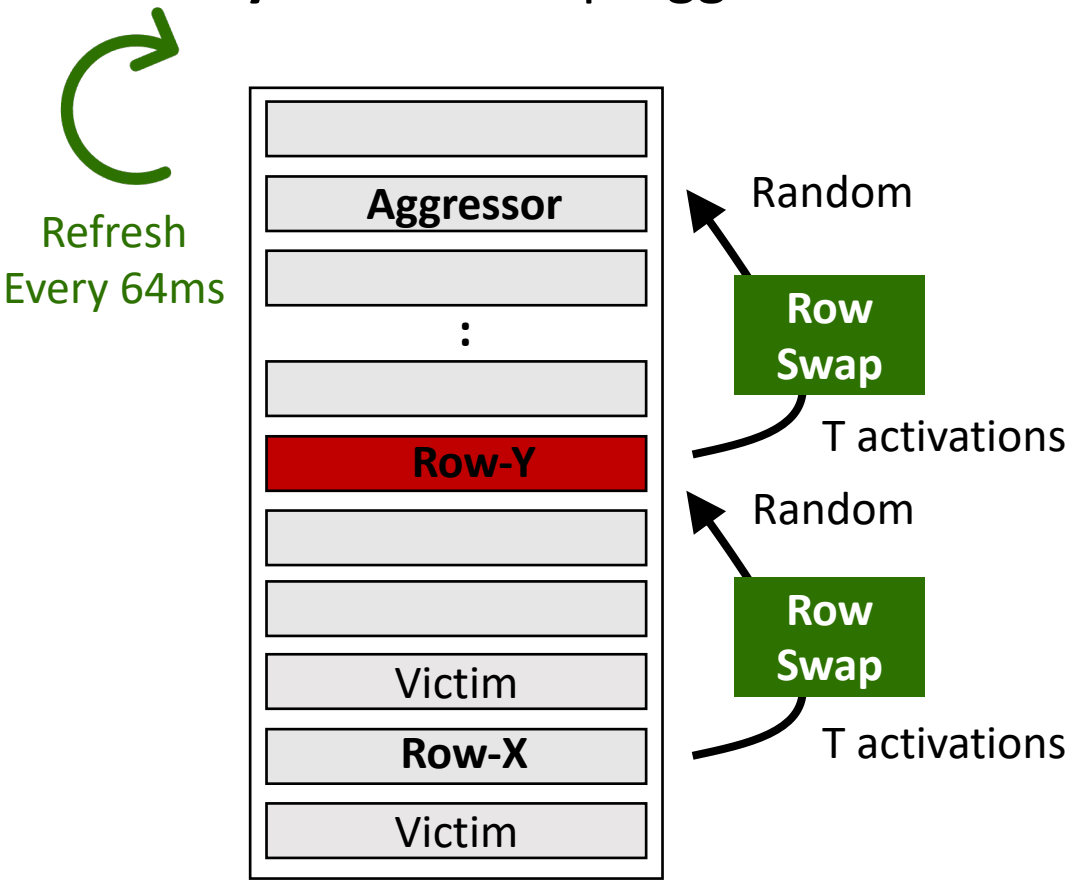
Key Idea: Remap Aggressor Rows to Break Spatial Correlation with Victim Rows



Security Guarantee: No Row Crosses Rowhammer Threshold Activations within 64ms

Aggressor Focused Mitigation: Randomized Row-Swap

Key Idea: Remap Aggressor Rows to Break Spatial Correlation with Victim Rows



Security Guarantee: No Row Crosses Rowhammer Threshold Activations within 64ms

Lower T (Swap Threshold) → Better Security

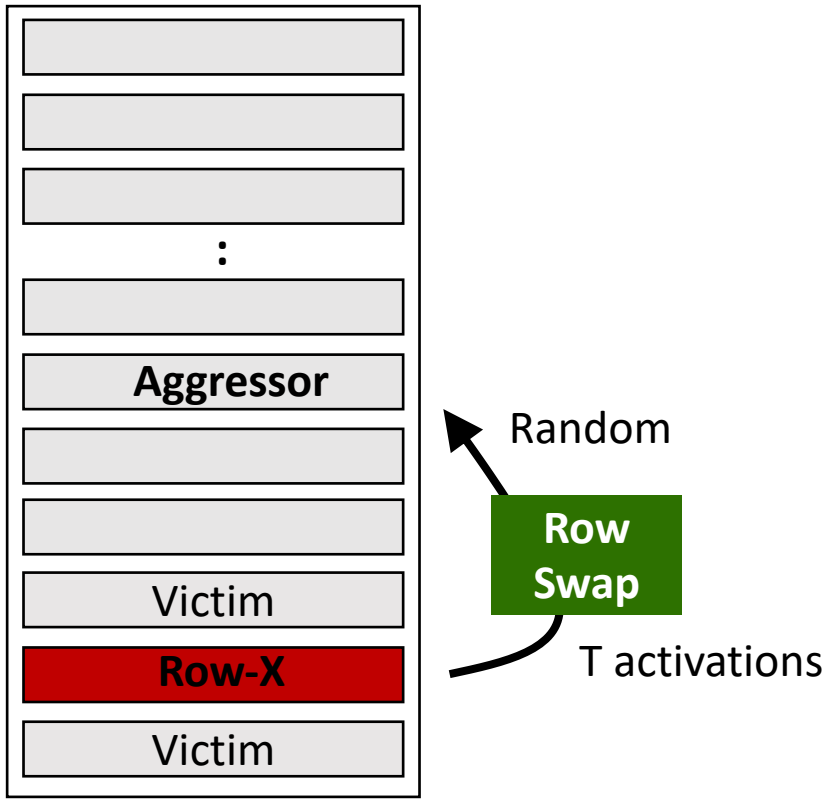
Security Analysis

TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern

(Single-sided, Double-Sided, Half-Double)

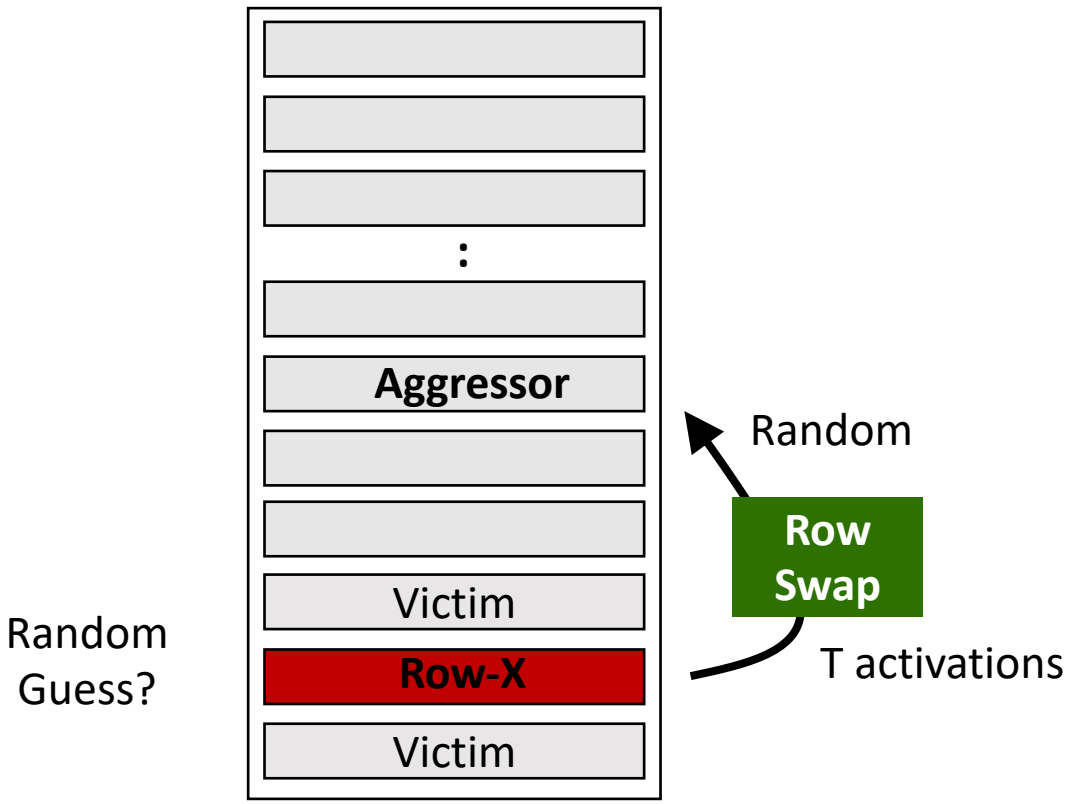
Security Analysis

TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern
(Single-sided, Double-Sided, Half-Double)



Security Analysis

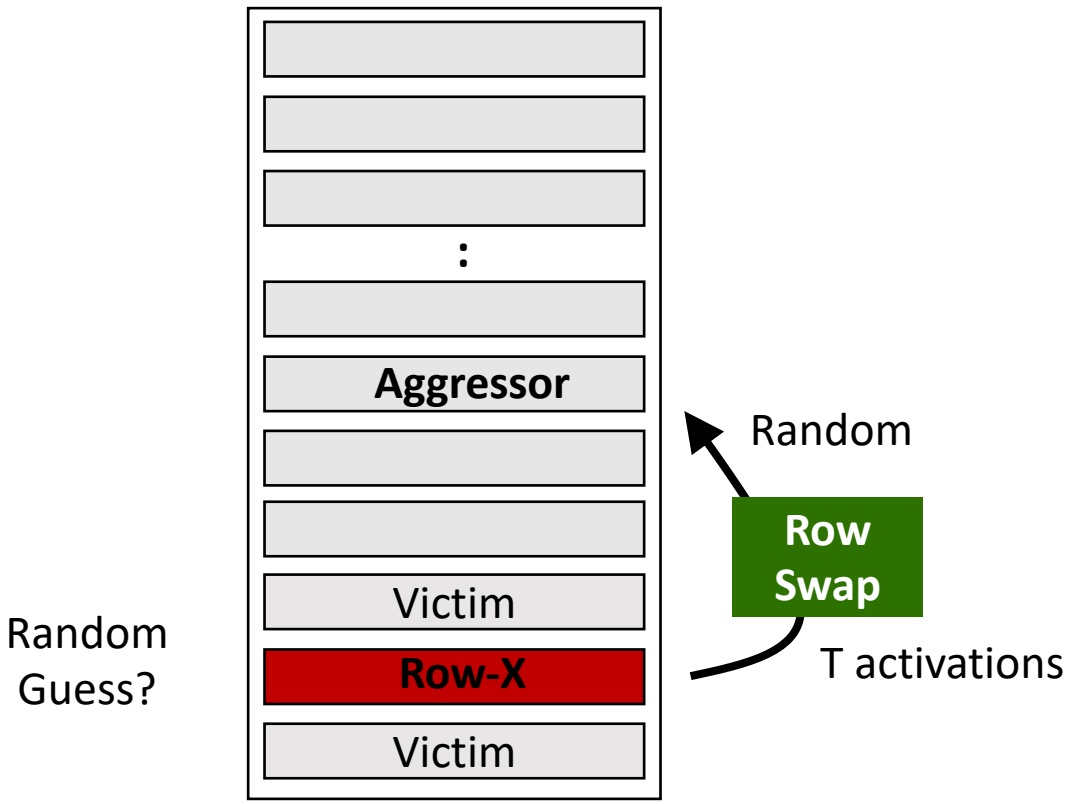
TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern
(Single-sided, Double-Sided, Half-Double)



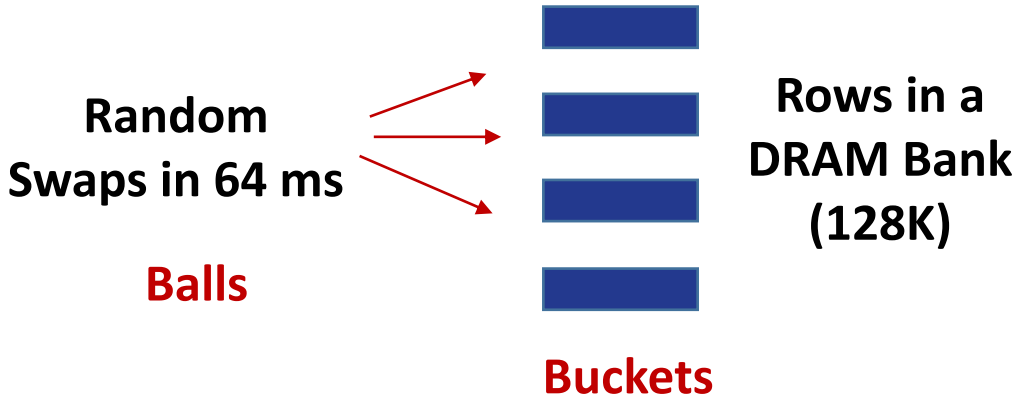
Security Analysis

TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern

(Single-sided, Double-Sided, Half-Double)

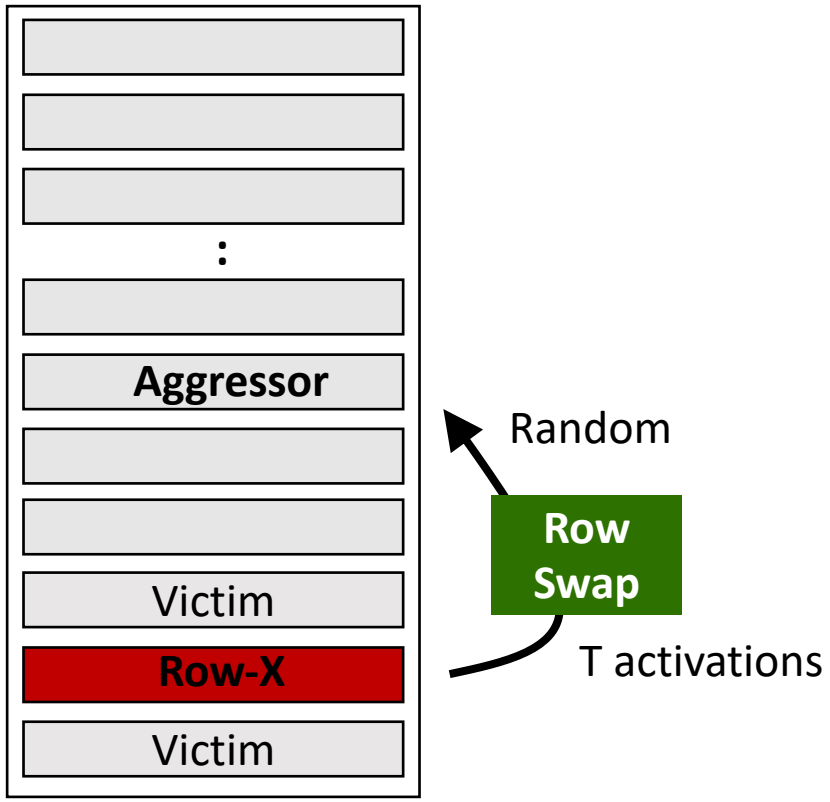


Buckets and Balls Problem

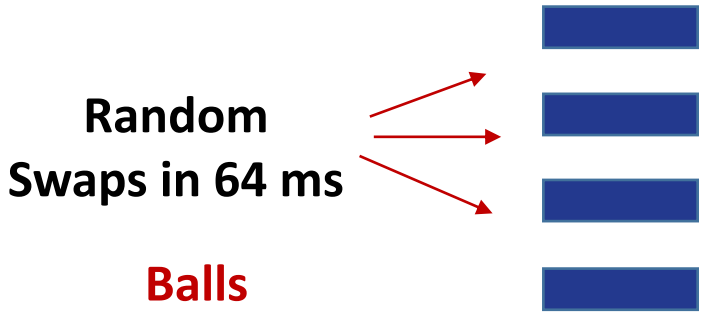


Security Analysis

TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern
(Single-sided, Double-Sided, Half-Double)



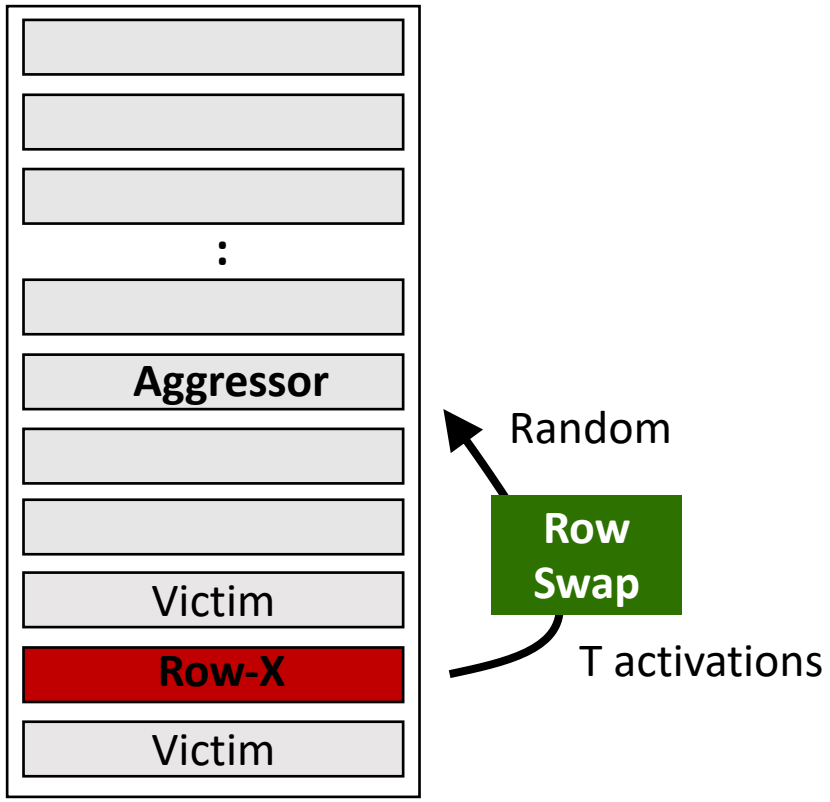
Buckets and Balls Problem



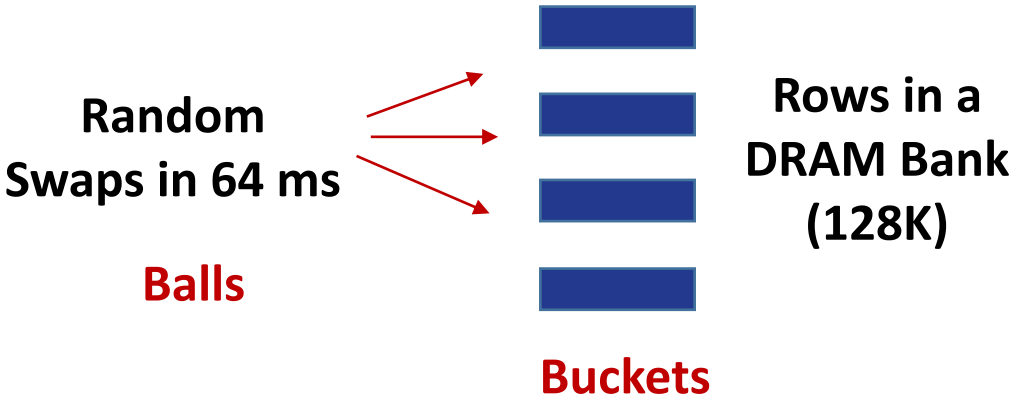
Swap Every T Activations	Attack Time
T = TRH/5	6.9 days
T = TRH/6	**3.8 years**
T = TRH/7	762 years

Security Analysis

TRH=4800 → Minimum Activations in 64ms on Row for Rowhammer via Any Pattern
(Single-sided, Double-Sided, Half-Double)



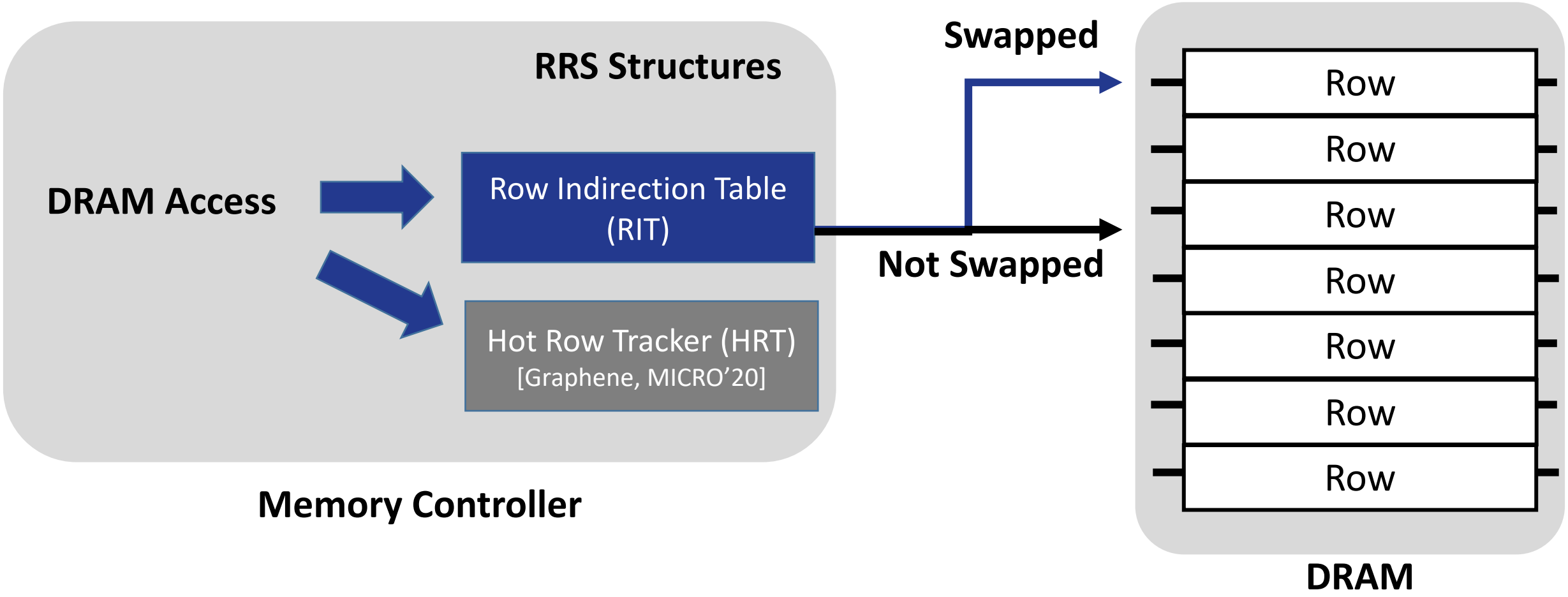
Buckets and Balls Problem



Swap Every T Activations	Attack Time
T = TRH/5	6.9 days
T = TRH/6	**3.8 years**
T = TRH/7	762 years

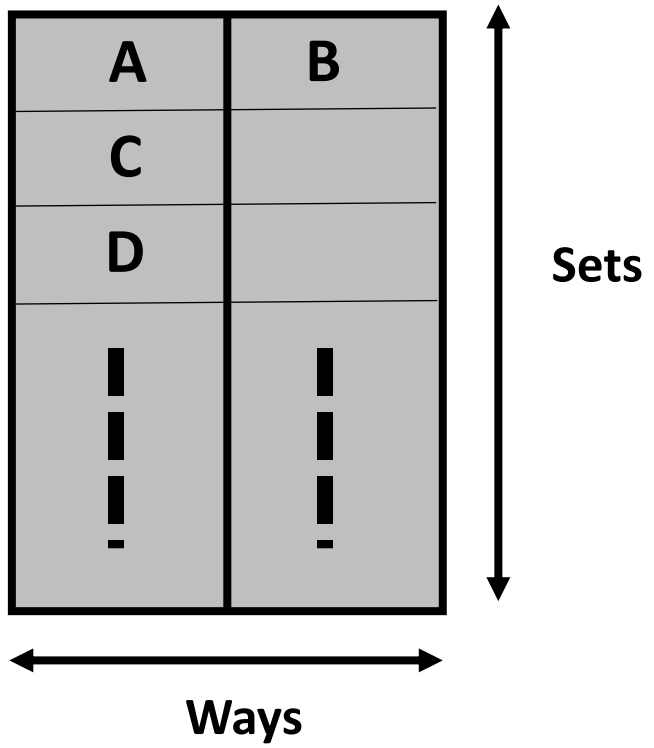
Continuous Remapping of Aggressor Rows Provides Principled Security for Years of Attack

Implementation of Randomized Row Swap



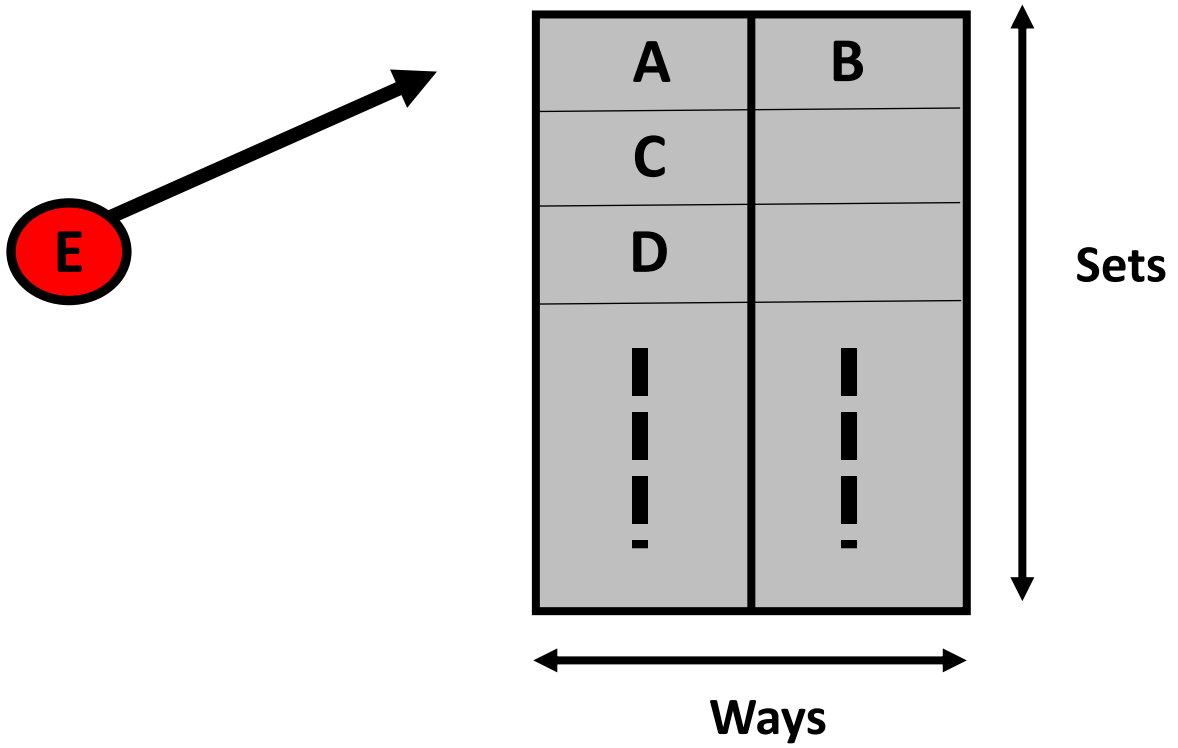
Collision Avoidance Table (CAT) for RIT and HRT

Problem: The tracking structures can be overwhelmed by repeated accesses



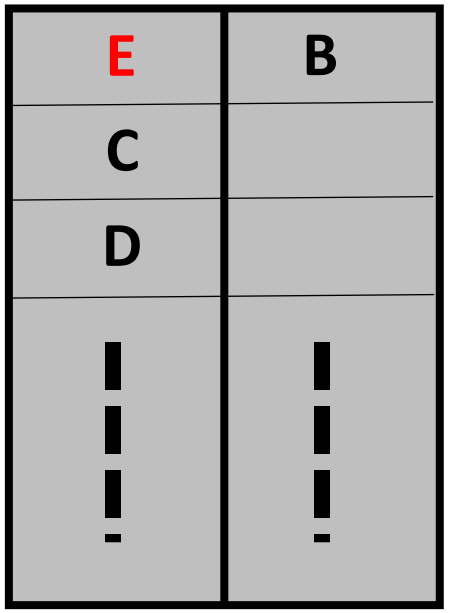
Collision Avoidance Table (CAT) for RIT and HRT

Problem: The tracking structures can be overwhelmed by repeated accesses



Collision Avoidance Table (CAT) for RIT and HRT

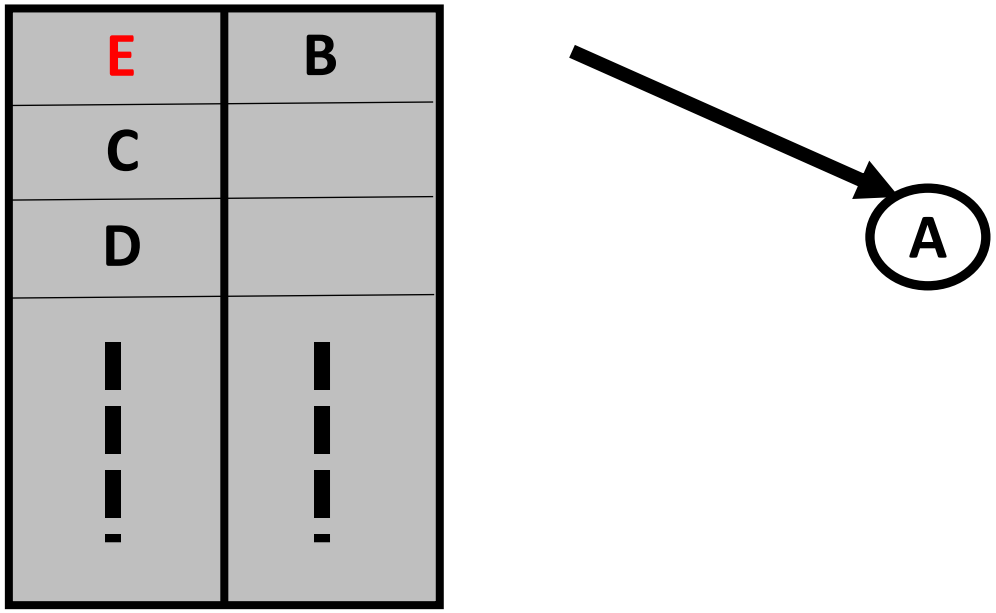
Problem: The tracking structures can be overwhelmed by repeated accesses



Conflicts in tracking structures → targeted row addresses to be expelled → Lose tracking of some rows

Collision Avoidance Table (CAT) for RIT and HRT

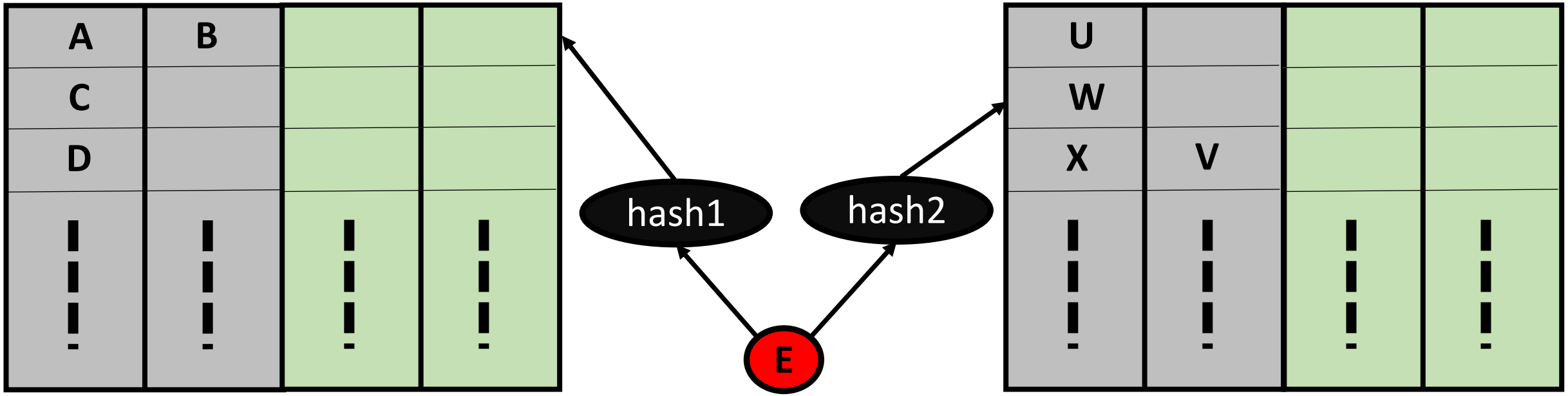
Problem: The tracking structures can be overwhelmed by repeated accesses



Conflicts in tracking structures → targeted row addresses to be expelled → Lose tracking of some rows

Collision Avoidance Table (CAT) for RIT and HRT

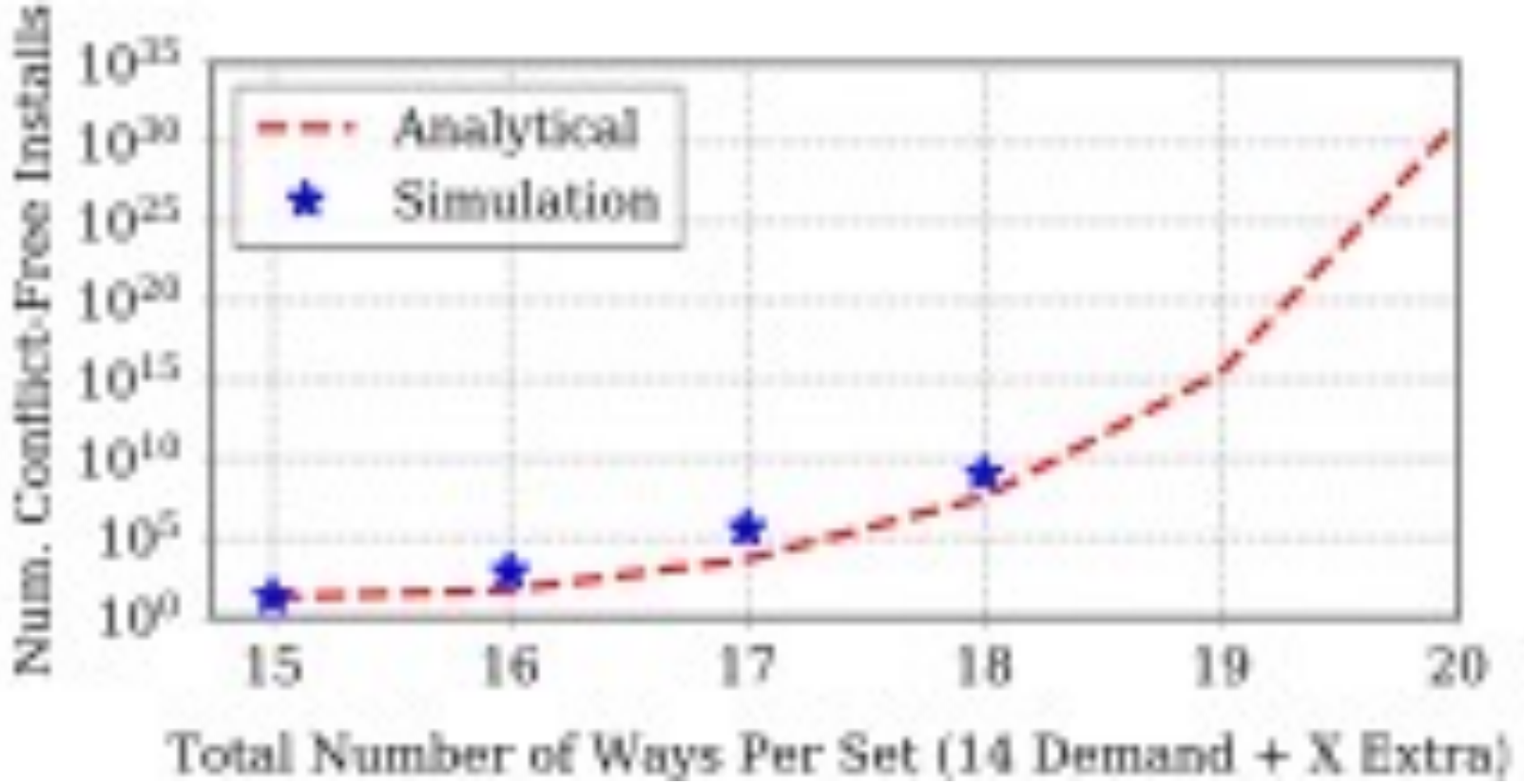
Solution: Leverage Power of Two Choices [MIRAGE – USENIX SEC'21]



Additional Ways + Multiple Hashes + Load Balancing + Random Replacement

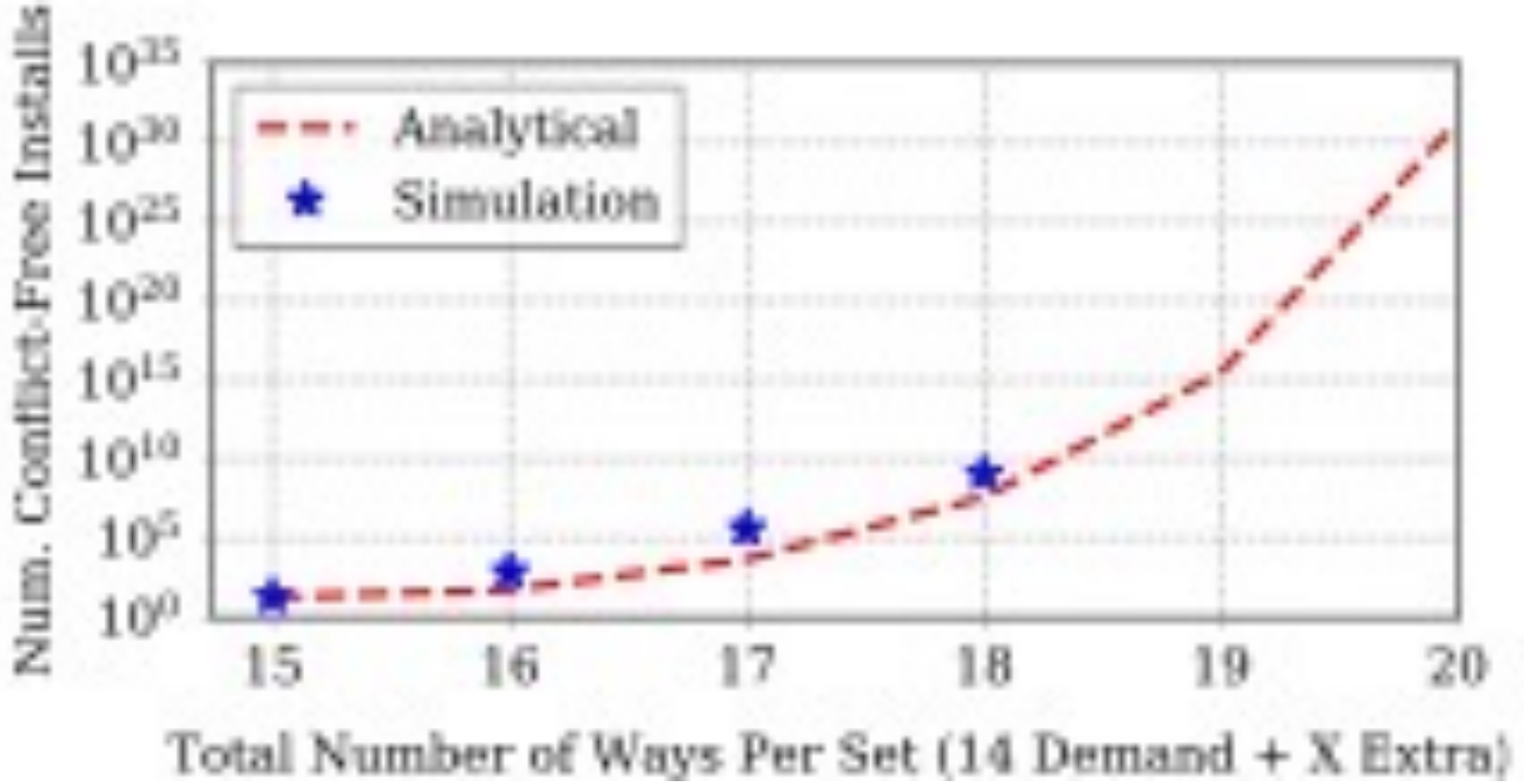
Collision Avoidance Table (CAT) for RIT and HRT

Solution: Leverage Power of Two Choices [MIRAGE – USENIX SEC'21]



Collision Avoidance Table (CAT) for RIT and HRT

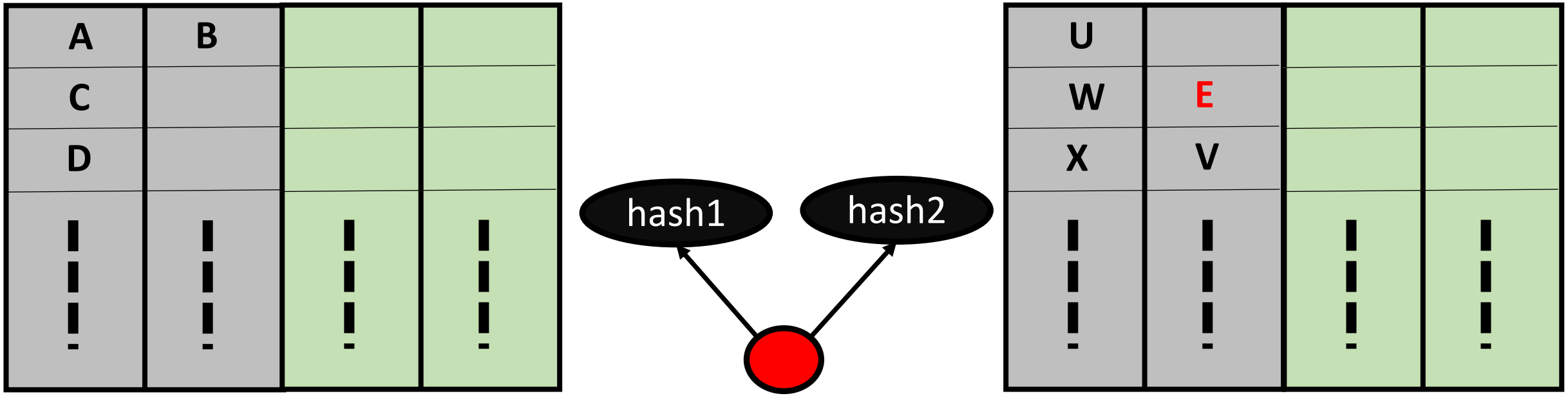
Solution: Leverage Power of Two Choices [MIRAGE – USENIX SEC'21]



With 14 Demand Ways and 6 Extra Ways → 10^{30} installs are needed to create a conflict

Collision Avoidance Table (CAT) for RIT and HRT

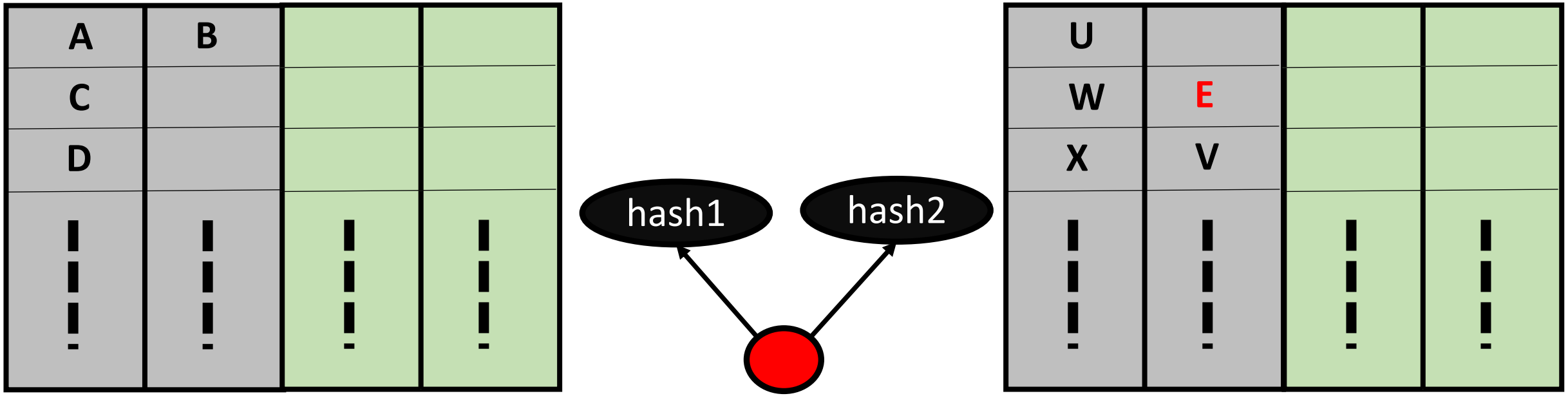
Solution: Leverage Power of Two Choices [MIRAGE – USENIX SEC'21]



Additional Ways + Multiple Hashes + Load Balancing + Random Replacement

Collision Avoidance Table (CAT) for RIT and HRT

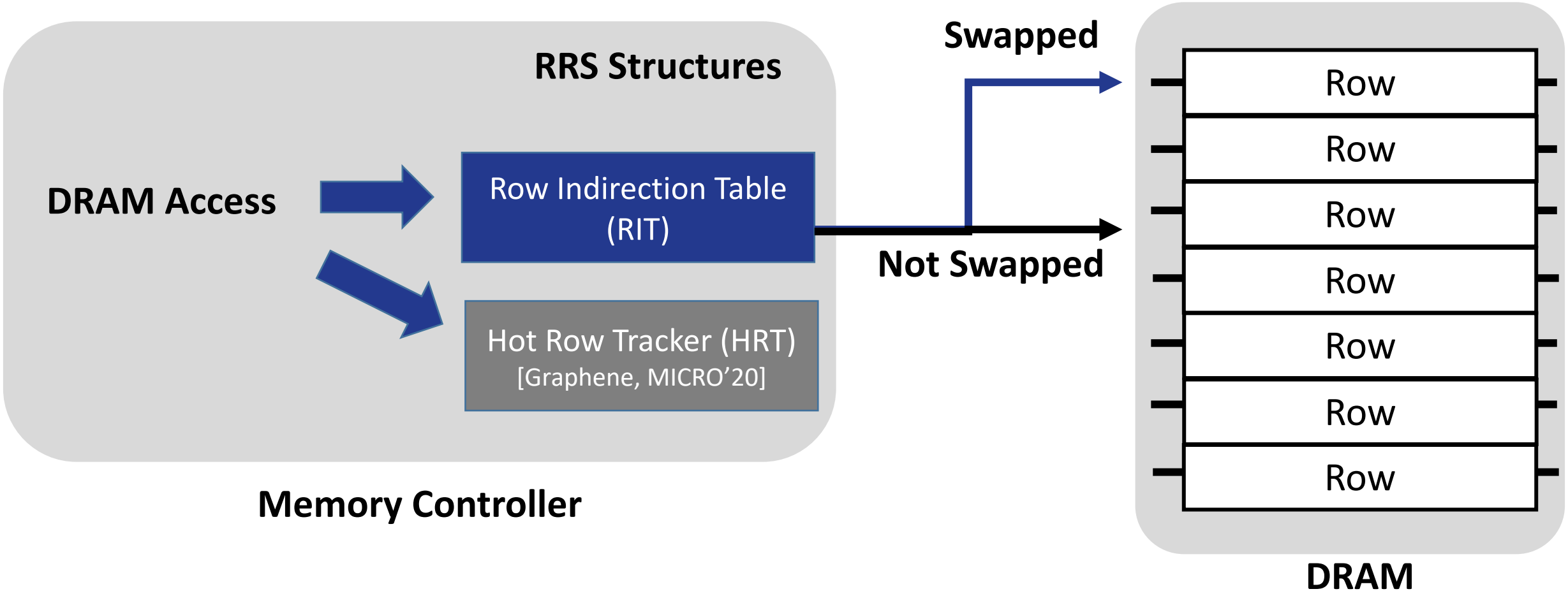
Solution: Leverage Power of Two Choices [MIRAGE – USENIX SEC'21]



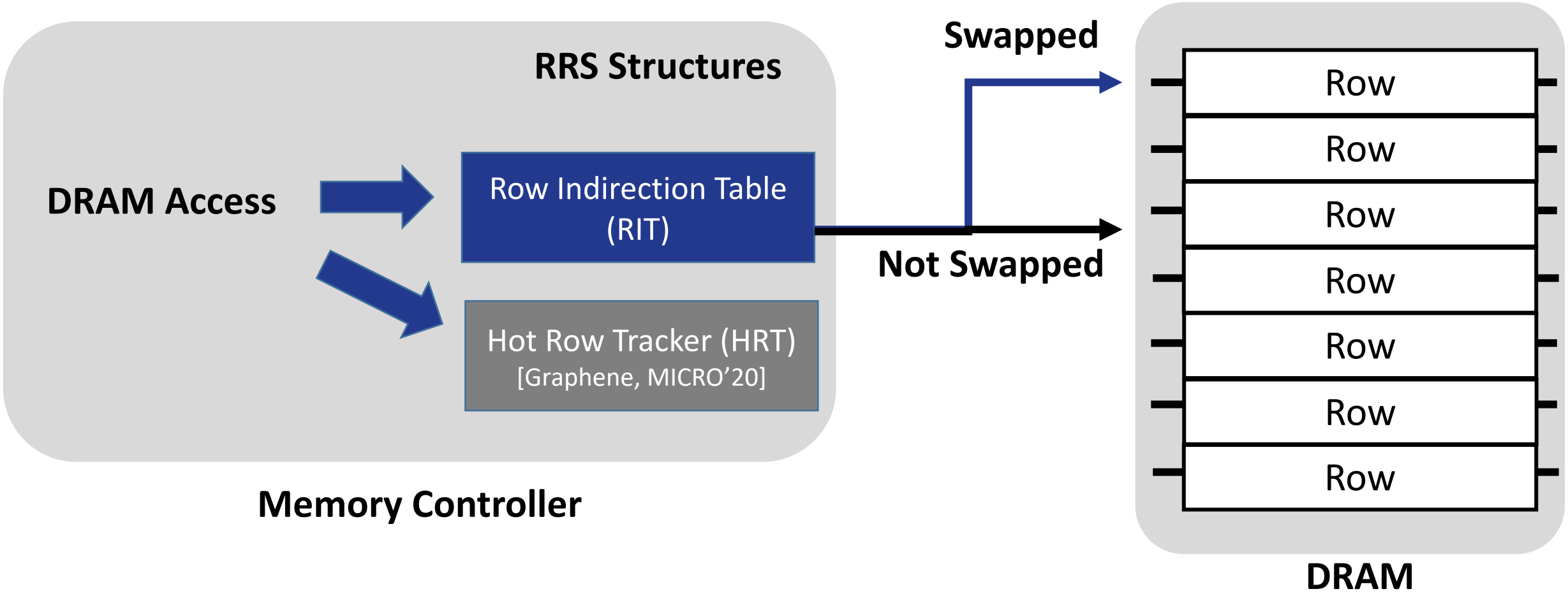
Additional Ways + Multiple Hashes + Load Balancing + Random Replacement

RIT and HRT use the CAT to avoid collisions → Prevents Conflict-Based Attacks

Implementation of Randomized Row Swap



Implementation of Randomized Row Swap

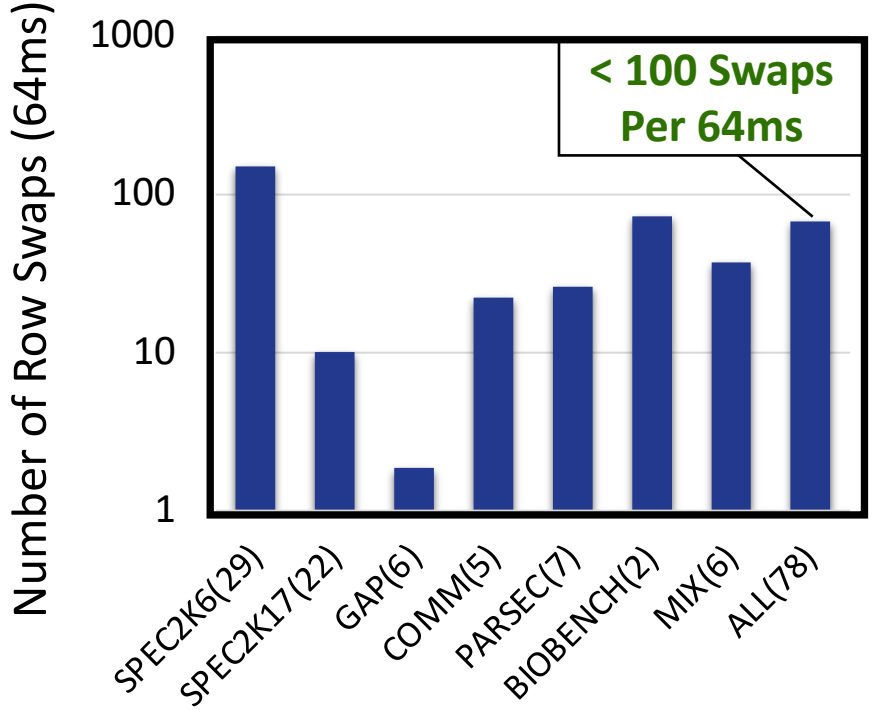


SRAM Storage Overheads (RIT, HRT) → 45 KB Per DRAM Bank x 16 Banks → 720 KB of SRAM

Performance Impact of Row Swaps

Frequency of Row Swaps Per 64ms

(1.5 microseconds per swap)

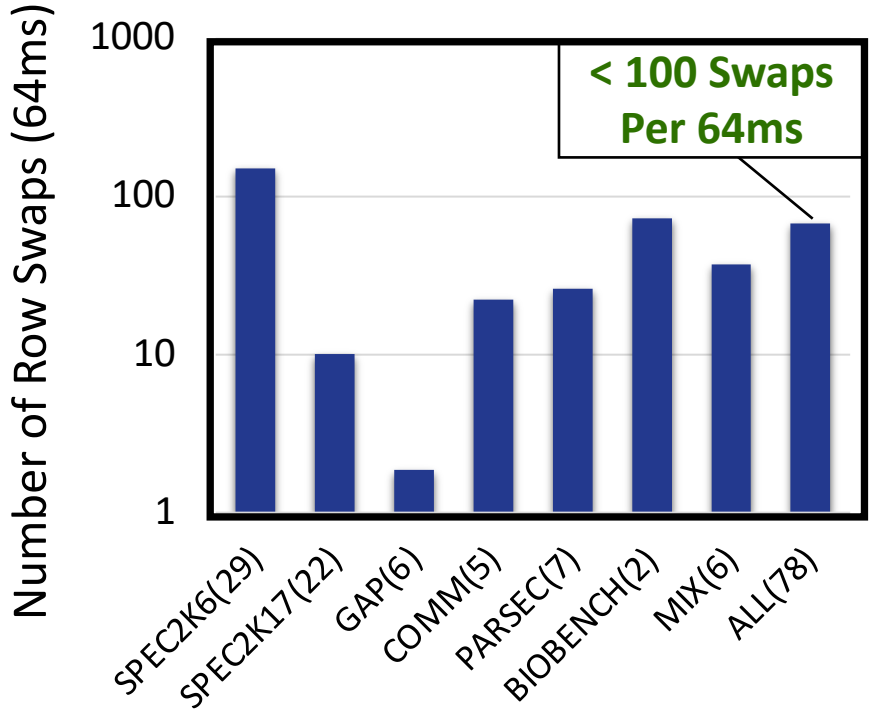


Saileshwar et al. Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation. In ASPLOS 2022

Performance Impact of Row Swaps

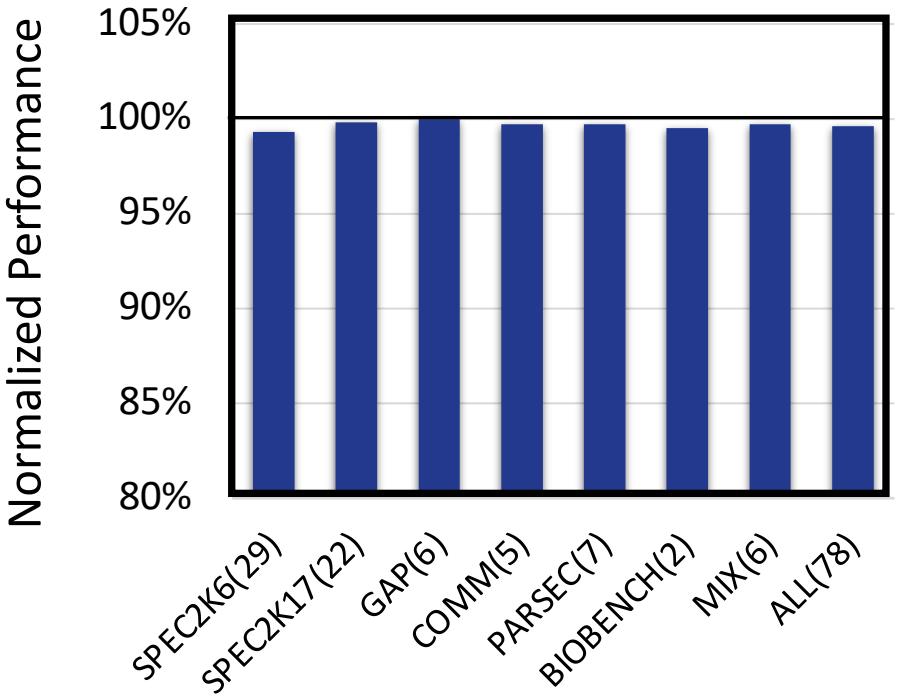
Frequency of Row Swaps Per 64ms

(1.5 microseconds per swap)



Negligible Performance Impact

(0.4% slowdown on average)

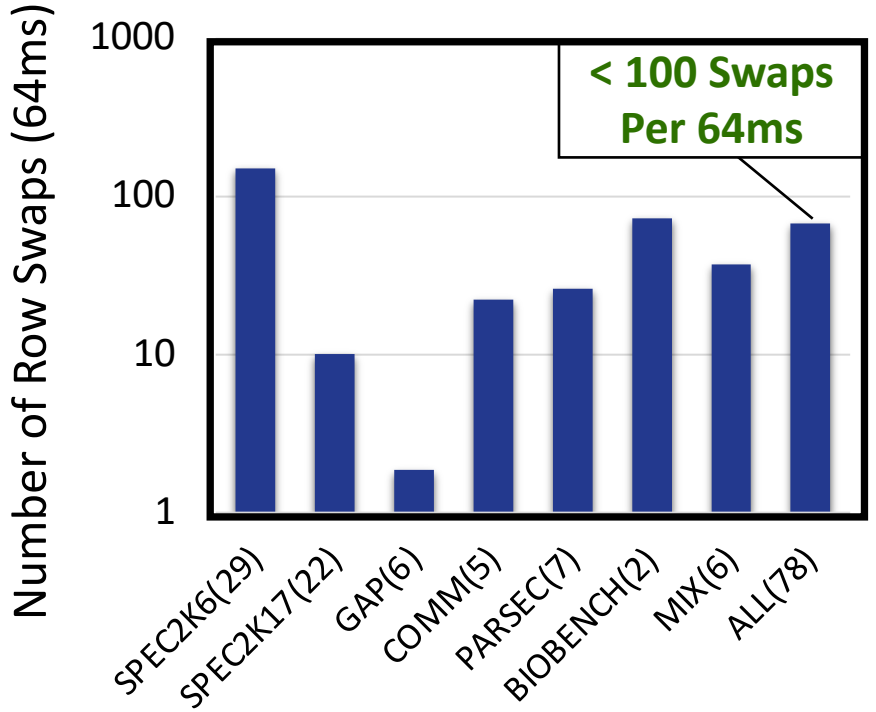


Saileshwar et al. Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation. In ASPLOS 2022

Performance Impact of Row Swaps

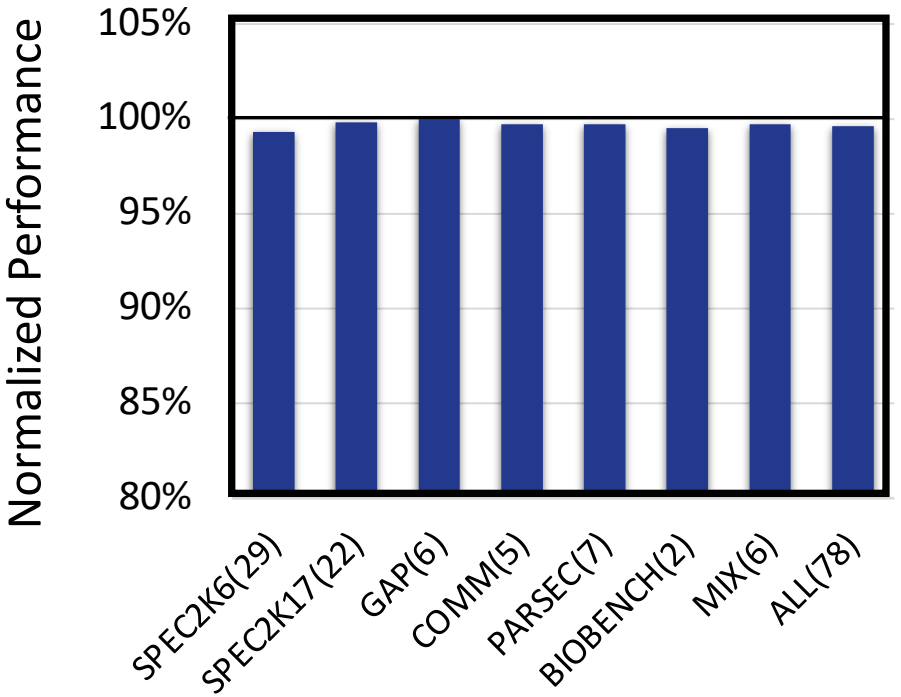
Frequency of Row Swaps Per 64ms

(1.5 microseconds per swap)



Negligible Performance Impact

(0.4% slowdown on average)

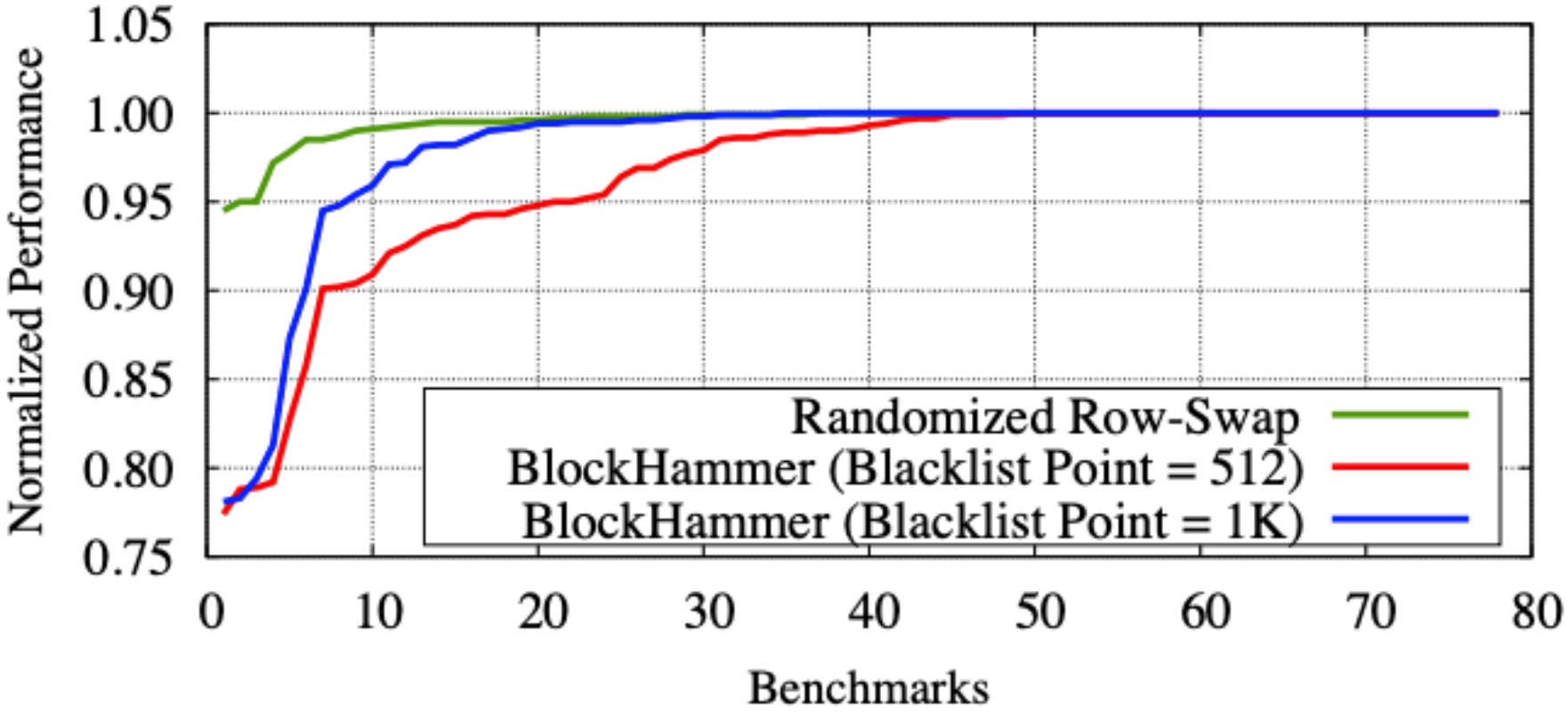


Randomized Row Swap has negligible performance impact due to infrequent swaps

Comparison with Prior Aggressor Focused Mitigation

Blockhammer [HPCA'21]

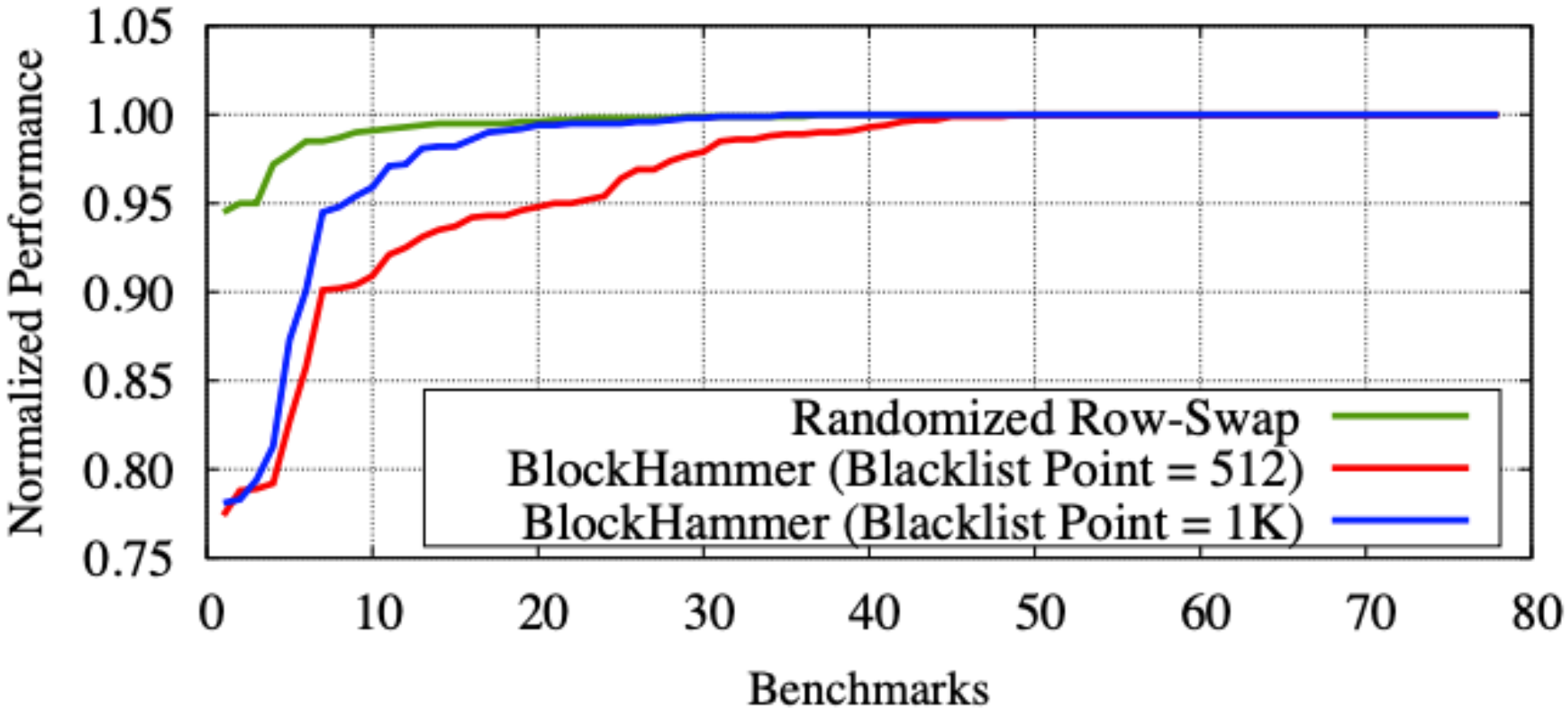
Rate-Throttles Aggressors at Blacklist Point (Earlier Than RTH) For Entire 64ms



Comparison with Prior Aggressor Focused Mitigation

Blockhammer [HPCA'21]

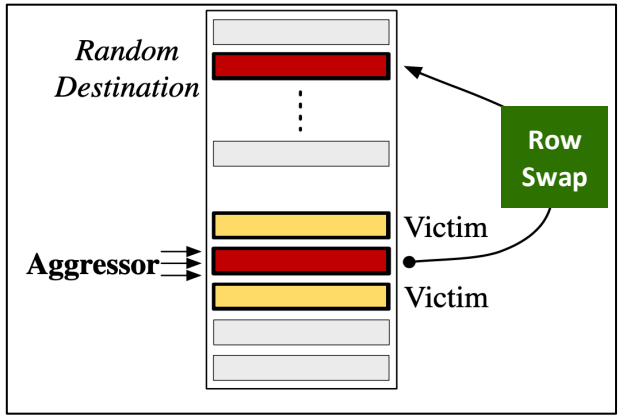
Rate-Throttles Aggressors at Blacklist Point (Earlier Than RTH) For Entire 64ms



Randomized Row Swap Incurs Much Lower Worst-Case Slowdowns vs Blockhammer

(mitigation is less expensive and hence adds less overheads in benign workloads)

Takeways from Randomized Row Swap



New Aggressor-Focused Mitigation – Swaps Aggressors with Random Destinations & Successively Remaps --> No Row Crosses Rowhammer Threshold

Randomized Row Swap incurs modest costs (0.4% slowdown, ~40KB SRAM/bank) while providing security against years of continuous attack

Resilient to TRResspass [SP'20], Half-Double (2021), Blacksmith [SP'22] attacks & potentially future attacks?