Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory

F. Nisa Bostancı K. Kanellopoulos
A. Olgun A. G. Yaglikci I. E. Yuksel
N. Mansouri Ghiasi Z. Bingol M. Sadrosadati O. Mutlu

SAFARI

ETH zürich

Executive Summary

Motivation: Processing-in-Memory (PIM) architectures alleviate data movement bottleneck by bringing computation closer to where data resides and are being adopted in real products

Problem: *No* prior work analyzes and evaluates the security of PIM architectures against timing side and covert channel attacks

Key Observation: PIM architectures create opportunities for critical main memory-based timing attacks due to two reasons:

- PIM provides direct main memory access, a key building block for high-throughput attacks
- Defenses against these attacks are highly costly or inapplicable to PIM architectures

IMPACT: a set of high-throughput **I**n-**M**emory **P**rocessing-based timing **At**tacks that leverage direct and fast main memory accesses enabled by PiM architectures. IMPACT:

- eliminates expensive cache bypassing steps used in main memory-based timing attacks
- leverages the intrinsic parallelism of PIM operations

Case Studies:

- **Two high-throughput covert channel attacks** leveraging different PIM architectures
- A **side-channel attack** that leaks private information of concurrently running victim applications with low error rate

Mitigating IMPACT: We discuss and evaluate **four different countermeasures** against IMPACT, eventually concluding that mitigating IMPACT incurs high performance overheads

Outline

Motivation and Problem

Key Observation

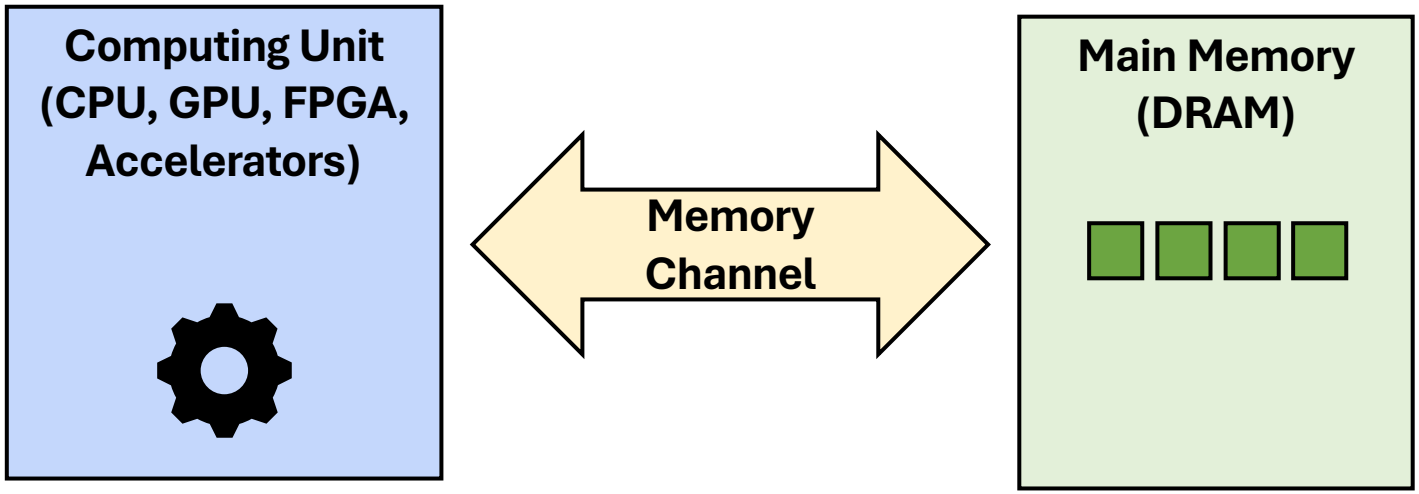
IMPACT

Mitigating IMPACT

Conclusion

Data Movement Bottleneck

- Today's computing systems are processor centric
- All data is processed in the processor → at great system cost



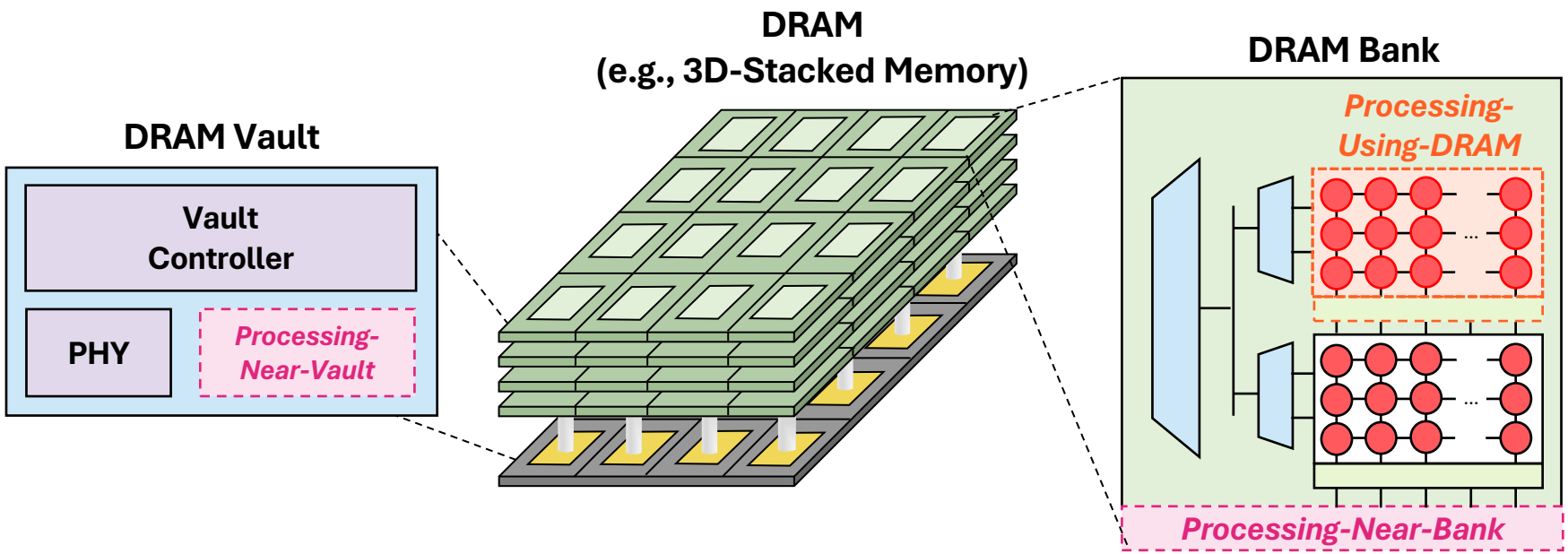
More than 60% of the total system energy is spent on data movement[1]

[1] A. Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS'18

Processing-In-Memory (PIM)

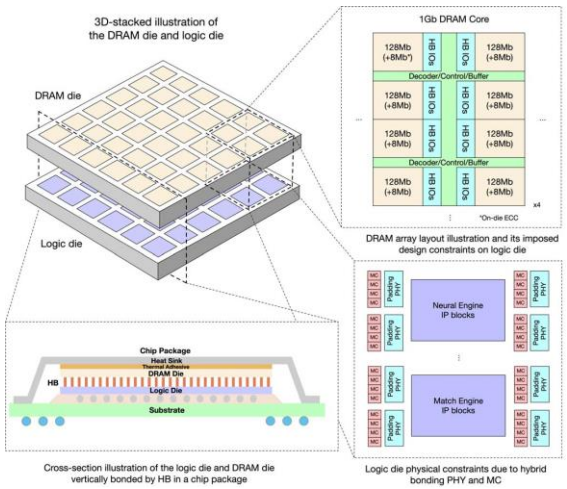
Two main approaches for Processing-In-Memory:

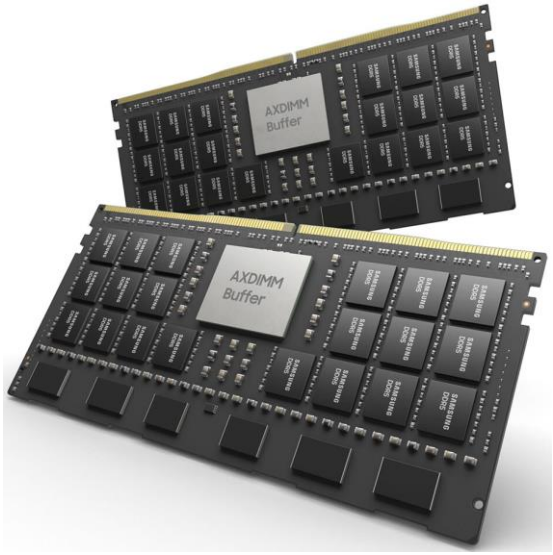
1. **Processing-Near-Memory**: PIM logic is added near the memory arrays or to the logic layer of 3D-stacked memory
2. **Processing-Using-Memory**: uses the analog operational principles of memory cells to perform computation

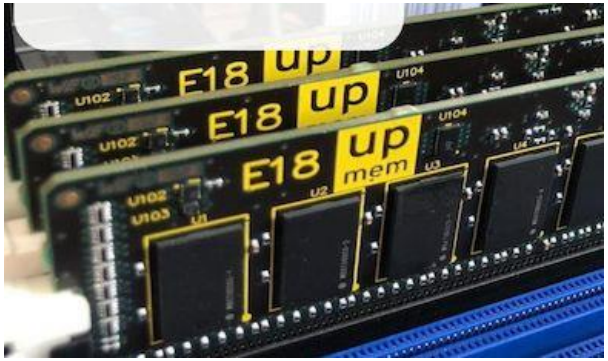


Problem

- A set of PIM techniques are already implemented in real products







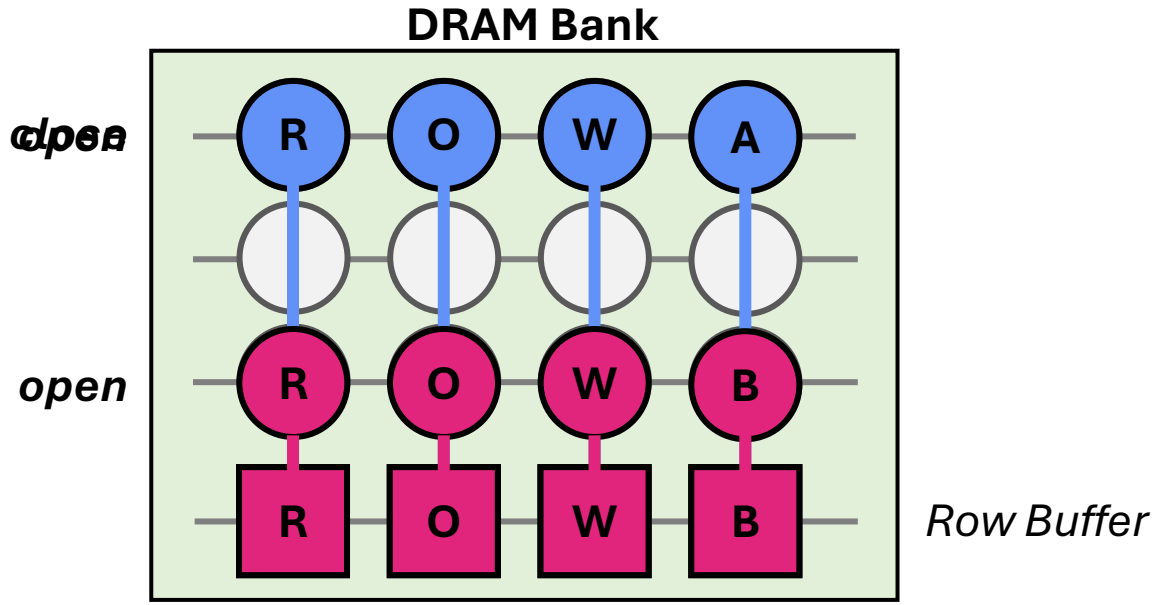
and many more are expected to be adopted

No prior work analyzes and evaluates the security of emerging PIM architectures against timing covert- and side-channel attacks

[1] Niu et al., 184QPS/W 64Mb/mm2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System, ISSCC 2022
[2] Ke et al. "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM", IEEE Micro (2021)
[3] https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem

Main Memory-Based Timing Channels

- The attacker exploits **the shared main memory states**
  - An example: **DRAM row buffer-based attacks** [Pessl+, USENIX Sec'14]

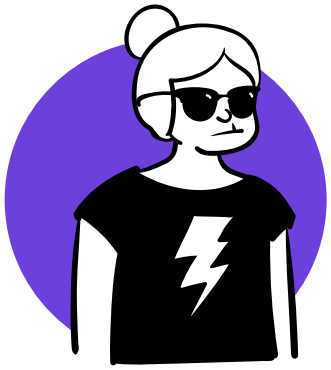


An access to an open row is served from the row buffer with lower latency

An access to any other row is served with higher latency

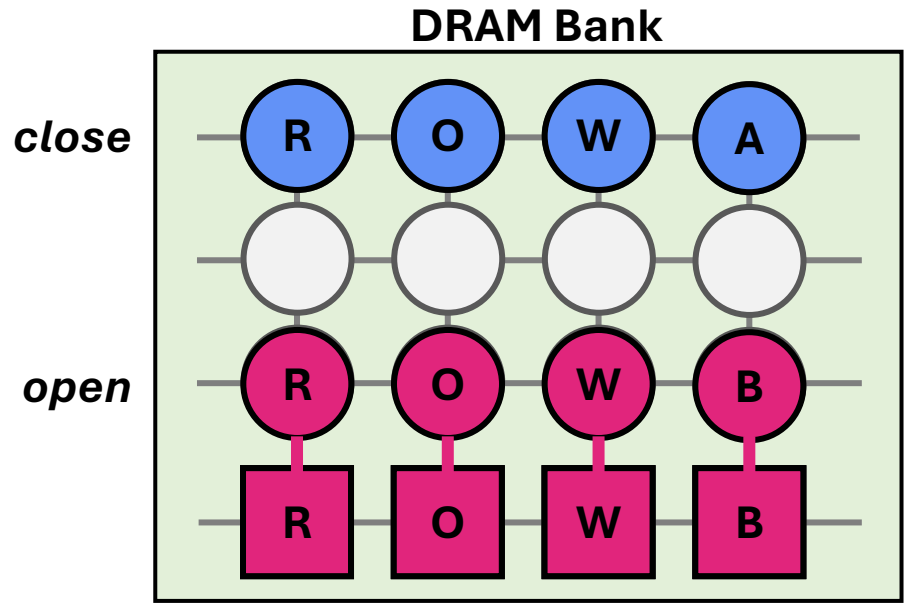
Main Memory-Based Timing Channels

- The attacker exploits **the shared main memory states**
  - An example: **DRAM row buffer-based attacks** [Pessl+, USENIX Sec'14]



Attacker

Send message: 10010100



Row Buffer

Victim

e.g., cross-CPU attacks

An access to an open row is served from the row buffer with lower latency

An access to any other row is served with higher latency

Observable from userspace applications

SAFARI

Outline

Motivation and Problem

Key Observation

IMPACT

Mitigating IMPACT

Conclusion

Key Observation

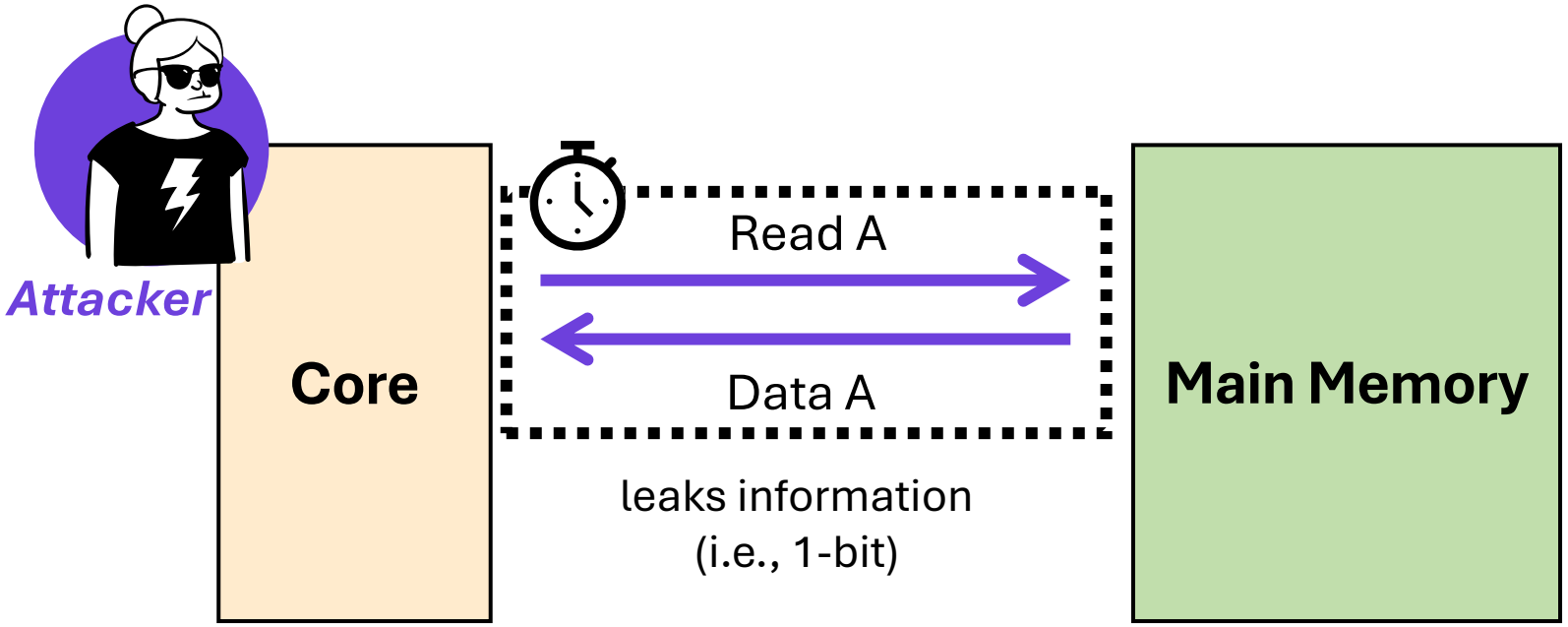
- PIM architectures create opportunities for critical main memory-based timing attacks due to two reasons:

1 PIM architectures provide **direct access to main memory**, **a key building block** for **high-throughput main memory-based timing attacks**

2 Defenses against these attacks either **incur high performance overheads** or **are *not* applicable to PIM architectures**

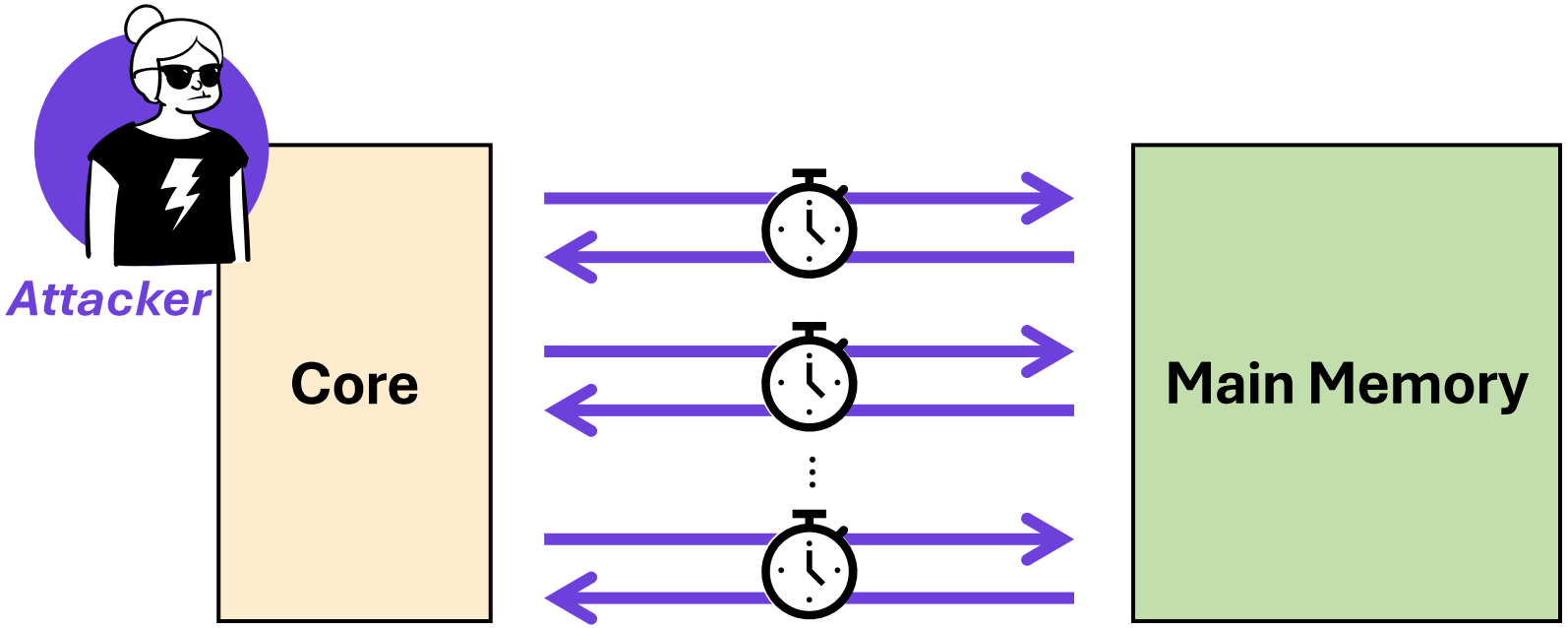
1. Direct Access to Main Memory - (I)

- Attacker leaks information by timing a memory access



1. Direct Access to Main Memory - (I)

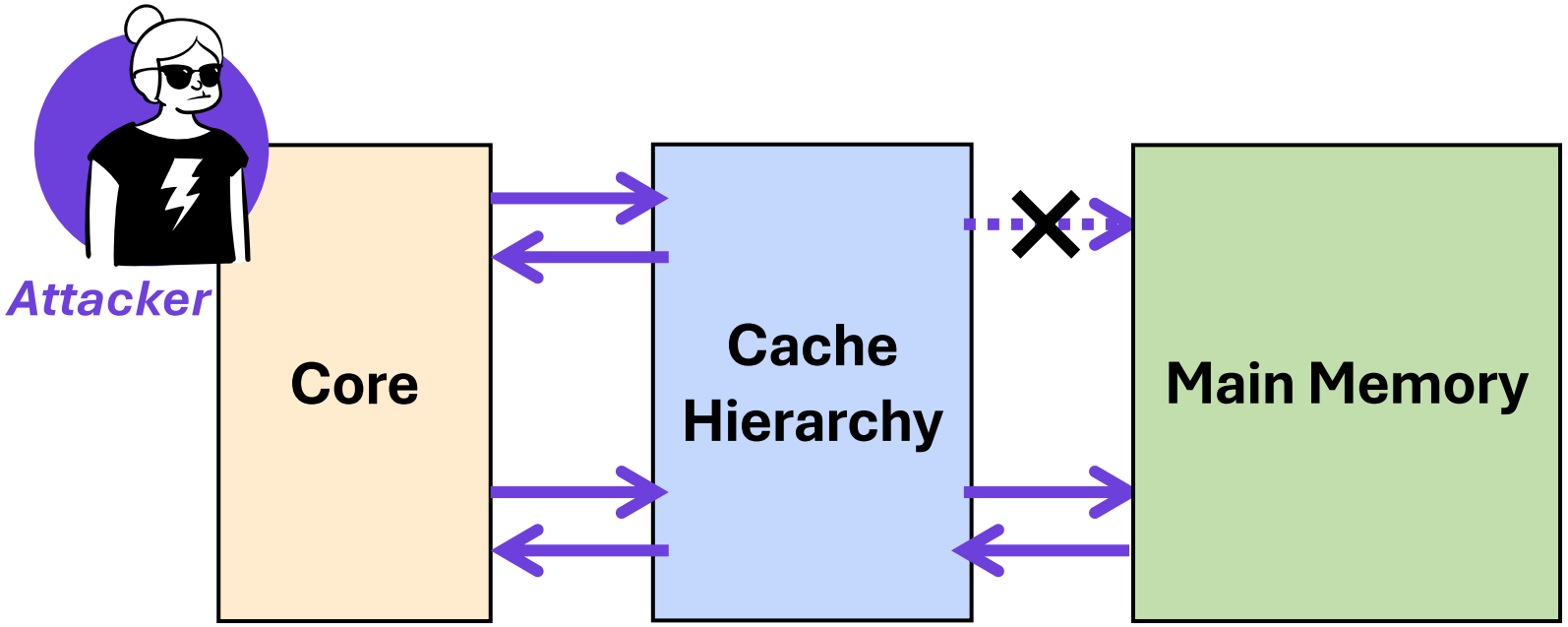
- Attacker leaks information by timing a memory access



- High throughput attacks require **reliable** and **fast access** to main memory

1. Direct Access to Main Memory - (II)

- High throughput attacks are difficult in compute-centric architectures

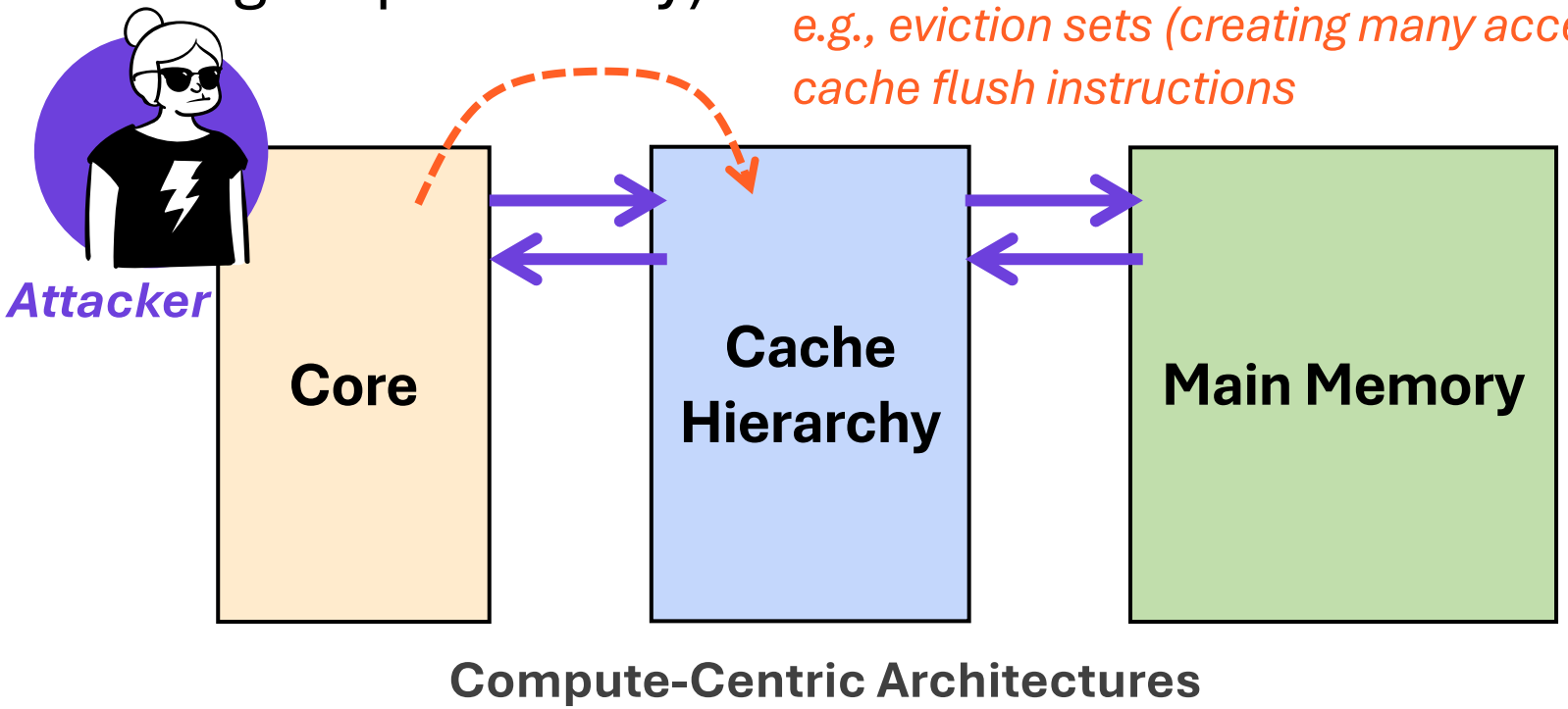


Compute-Centric Architectures

- Deep cache hierarchies
  - filter memory accesses and
  - incur additional latency

1. Direct Access to Main Memory - (II)

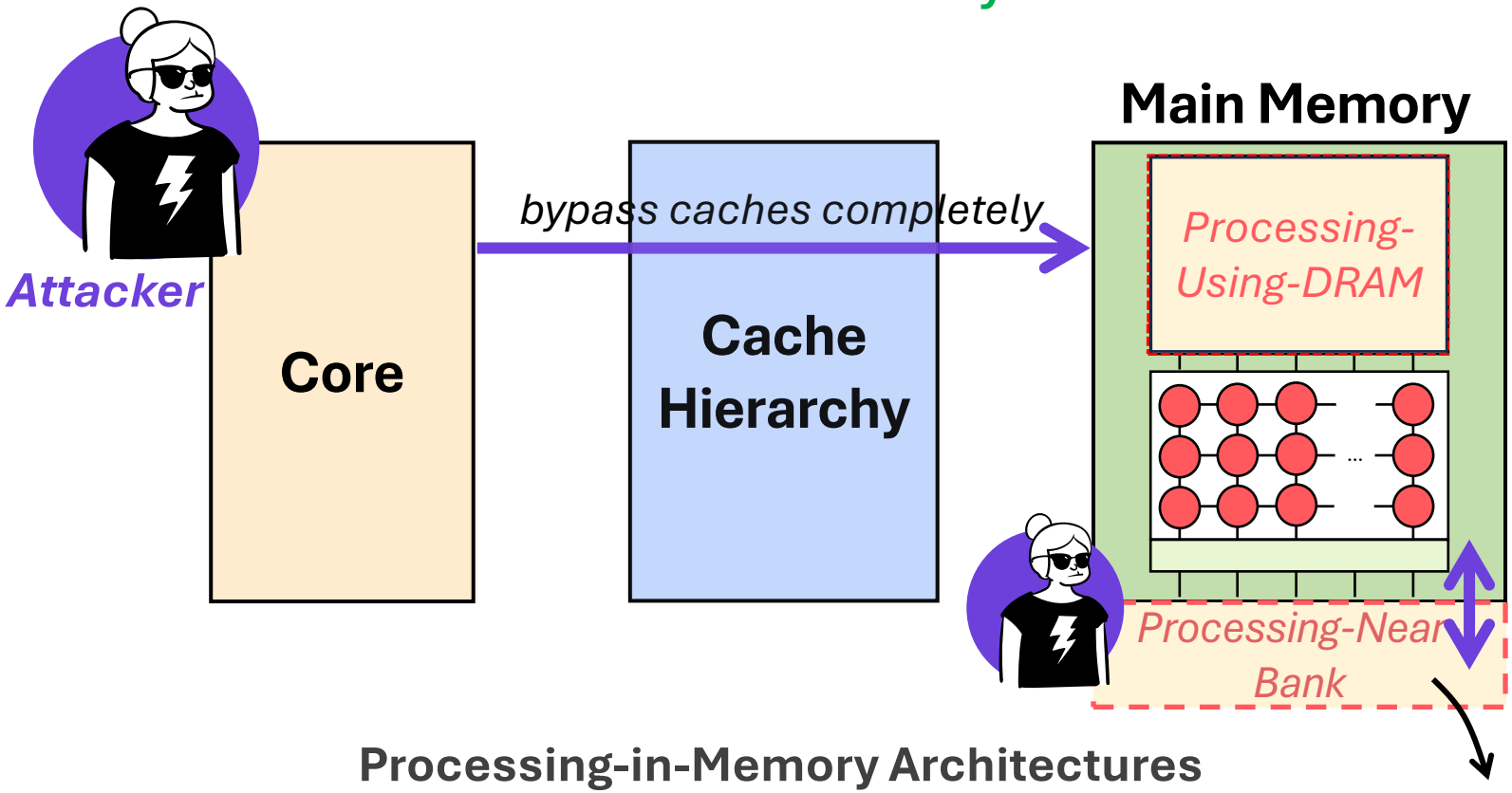
- Attackers manipulate the caches to access main memory (with a higher probability)



at the cost of a higher access latency

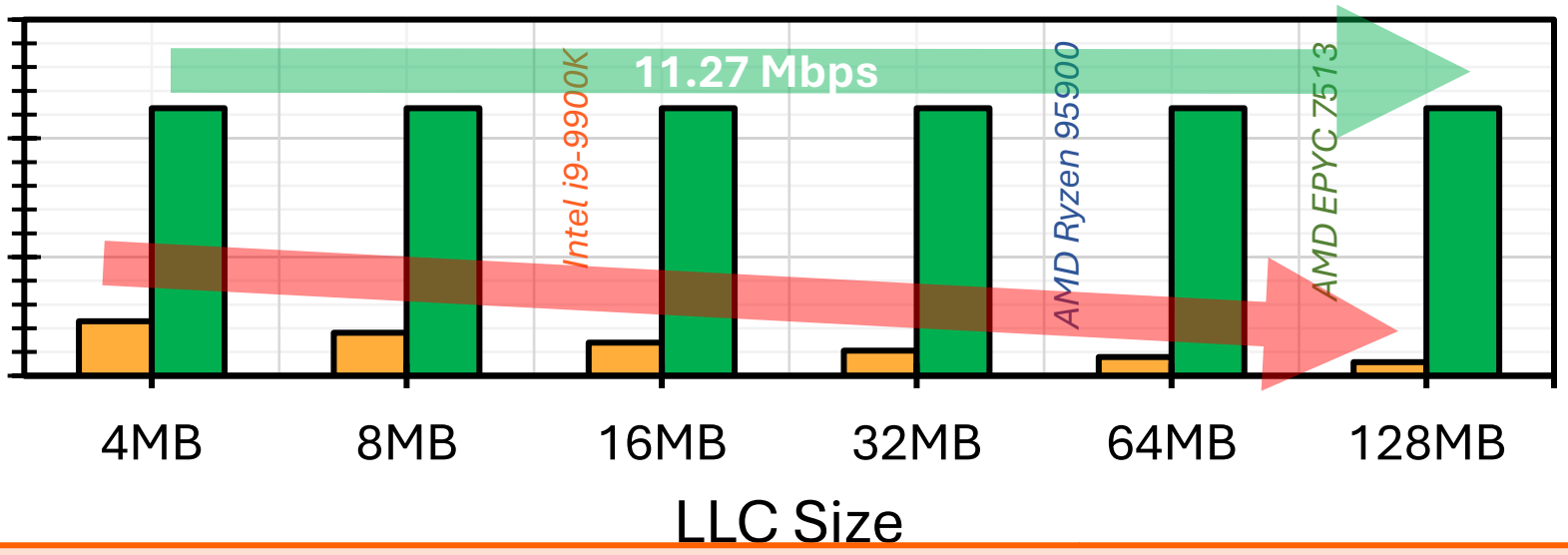
1. Direct Access to Main Memory - (III)

- PIM architectures provide userspace applications with fast and reliable direct main memory access



Impact of Direct Memory Access on Leakage Throughput

- Baseline Attack: State-of-the-art row buffer-based covert channel attack [1] with cache eviction sets [2]
- Direct Memory Access Attack: Row buffer-based covert channel attack bypassing the cache hierarchy



Direct Memory Access Attack maintains its leakage throughput regardless of the LLC size, in contrast to the baseline attack

[1] Pessl et al., "DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks," in USENIX Security, 2016.
[2] Liu et al., "Last-level cache side-channel attacks are practical," in SP, 2015.

Key Observation

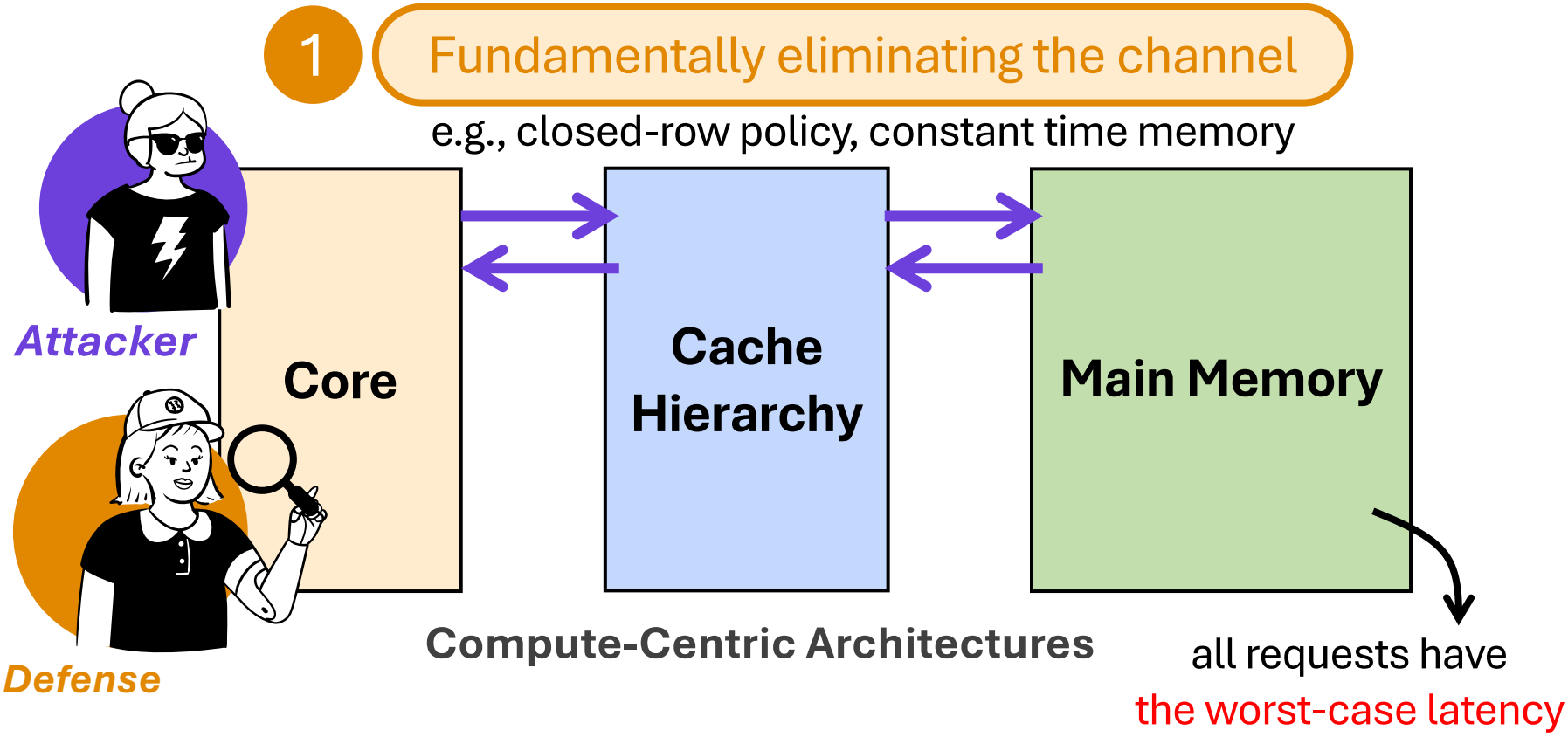
- PIM architectures create opportunities for critical main memory-based timing attacks due to two reasons:

1 PiM architectures provide **direct access to main memory**, **a key building block** for **high-throughput main memory-based timing attacks**

2 Defenses against these attacks either **incur high performance overheads** or **are *not* applicable to PIM architectures**

2. Hard-to-Mitigate with Practical Defenses

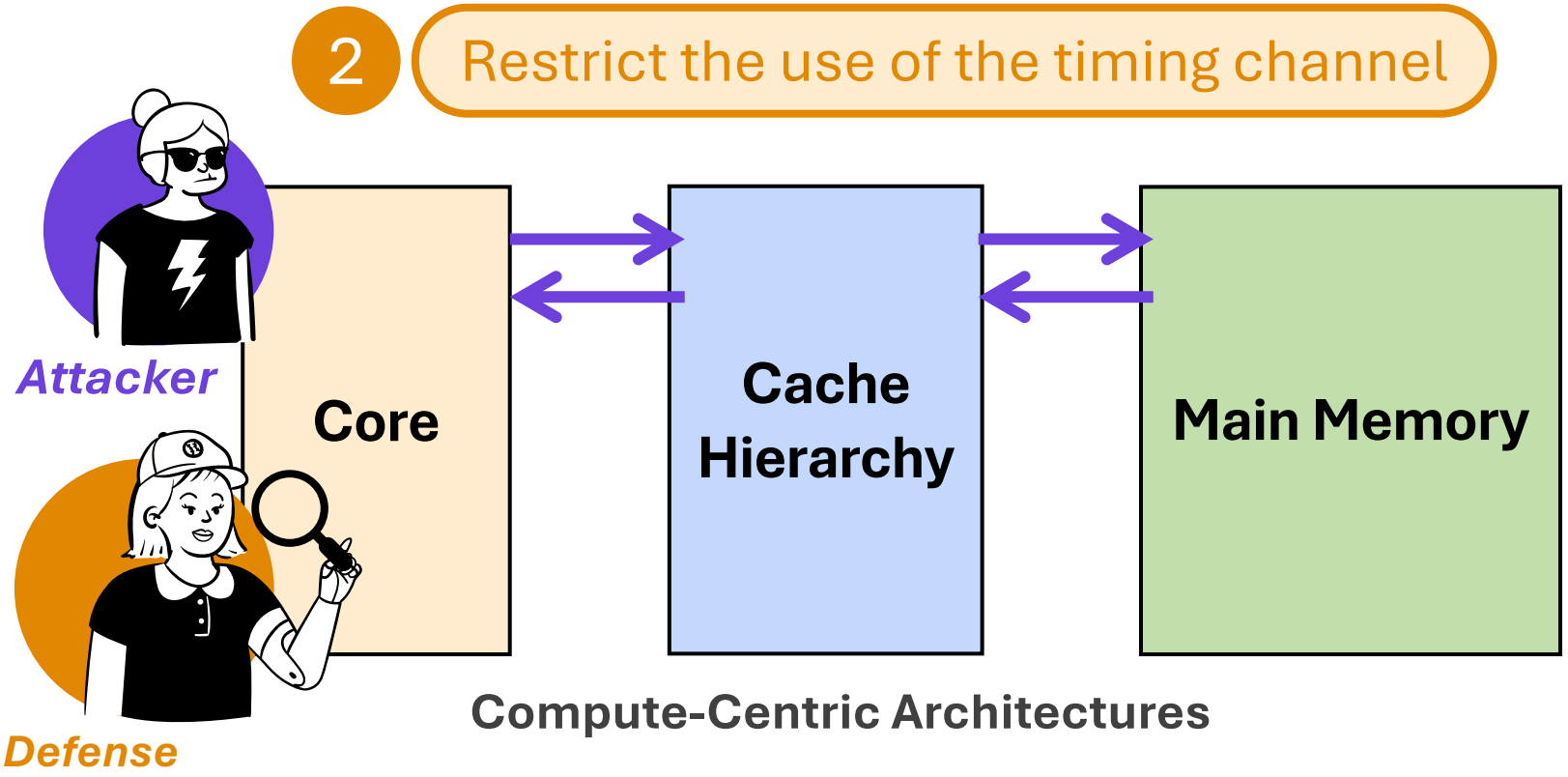
- Defenses have two options:



Defenses that eliminate the timing channel induce **high performance overheads** due to the increased memory access latency

2. Hard-to-Mitigate with Practical Defenses

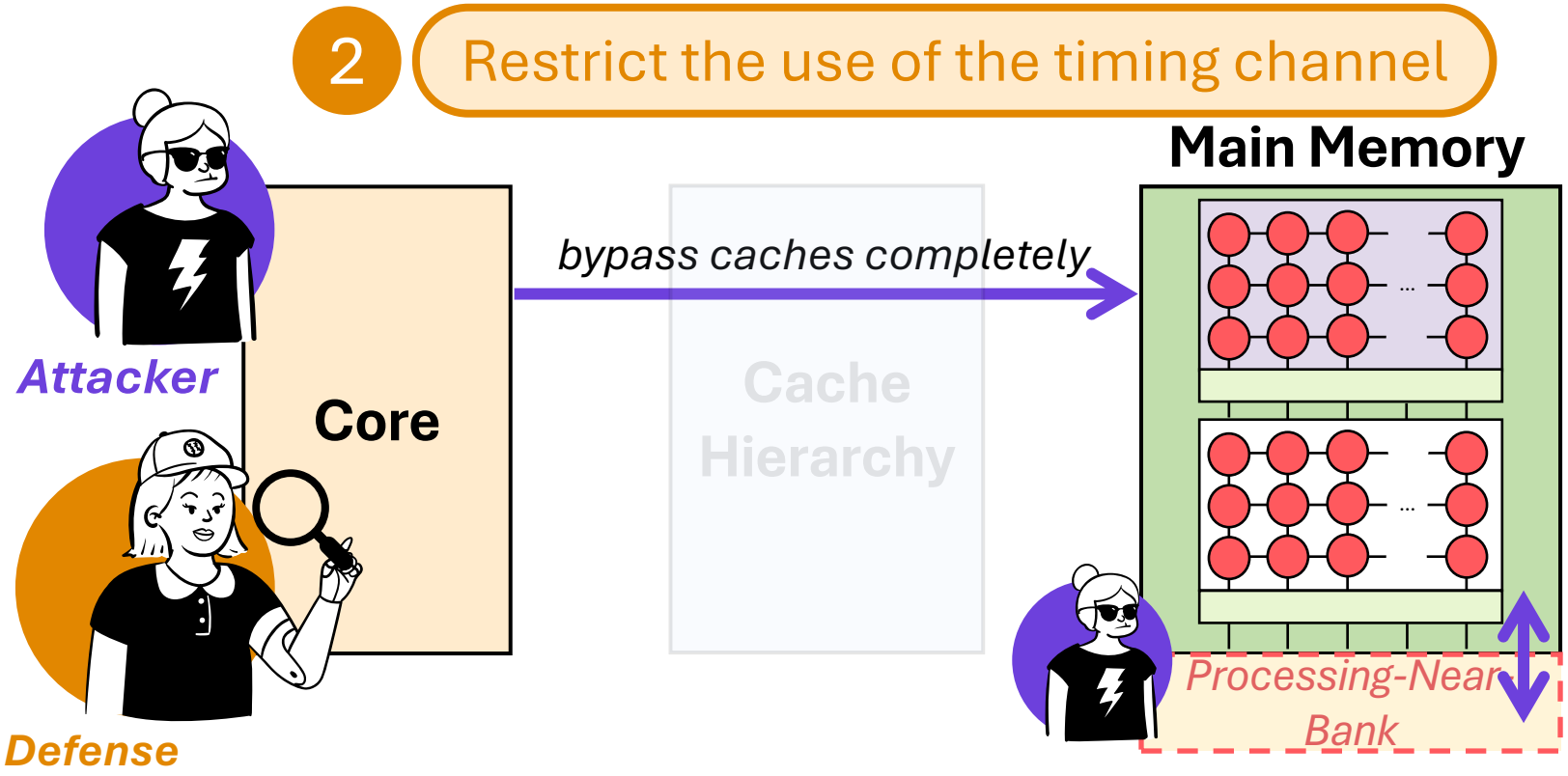
- Defenses have two options:



e.g., restrict cache flush instructions,
detect attacks based on cache statistics

2. Hard-to-Mitigate with Practical Defenses

- Defenses have two options:



Defenses that restrict the use of the timing channel with cache management methods are **inapplicable to PIM architectures**

Outline

Motivation and Problem

Key Observation

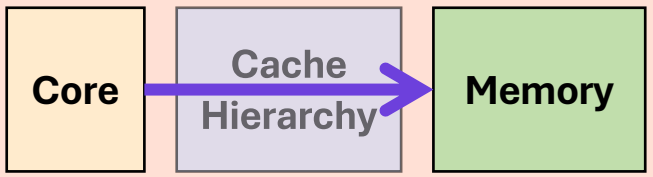
IMPACT

Mitigating IMPACT

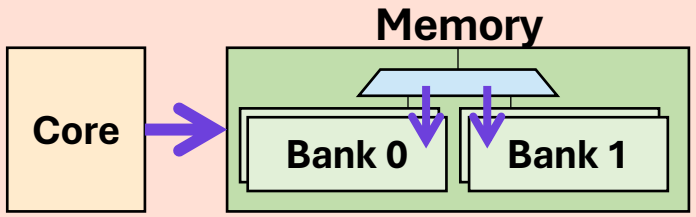
Conclusion

IMPACT

a set of high-throughput In-Memory Processing-based timing Attacks that leverage direct and fast main memory accesses enabled by PIM architectures



eliminate expensive cache bypassing steps



leverage intrinsic parallelism of PIM operations

IMPACT Case Studies

1. IMPACT-PNM Covert Channel

2. IMPACT-PUM Covert Channel

3. PNM-Based Genomic Privacy Attack

Evaluation Methodology

- **Environment:** System simulation using Sniper [Carlson+, SC'11]
  - **PNM architecture:** PIM-Enabled Instructions [Ahn+,ISCA'15]
  - **PUM architecture:** RowClone [Seshadri+,MICRO'13]

- **System Configuration:**

Processor	Out-of-order, 2.6GHz clock frequency
DRAM	DDR4, 1 channel, 1 rank/channel, 4 bank groups, 4 banks/bank group, 128K rows/bank
Mem Ctrl.	Open Row Policy, Row Timeout = 100 ns
MMU	L1 DTLB (4KB): 64-entry, 4-way, 1-cycle, L1 DTLB (2MB): 32-entry, 4-way, 1-cycle, L2 TLB: 1536-entry, 12-way, 12-cycle
LLC	2 MB/core, 16-way, 50-cycle access latency

- **Comparison Points:**
  - DRAMA-clflush, DRAMA-eviction, DMA-Engine

https://github.com/CMU-SAFARI/IMPACT

IMPACT Case Studies

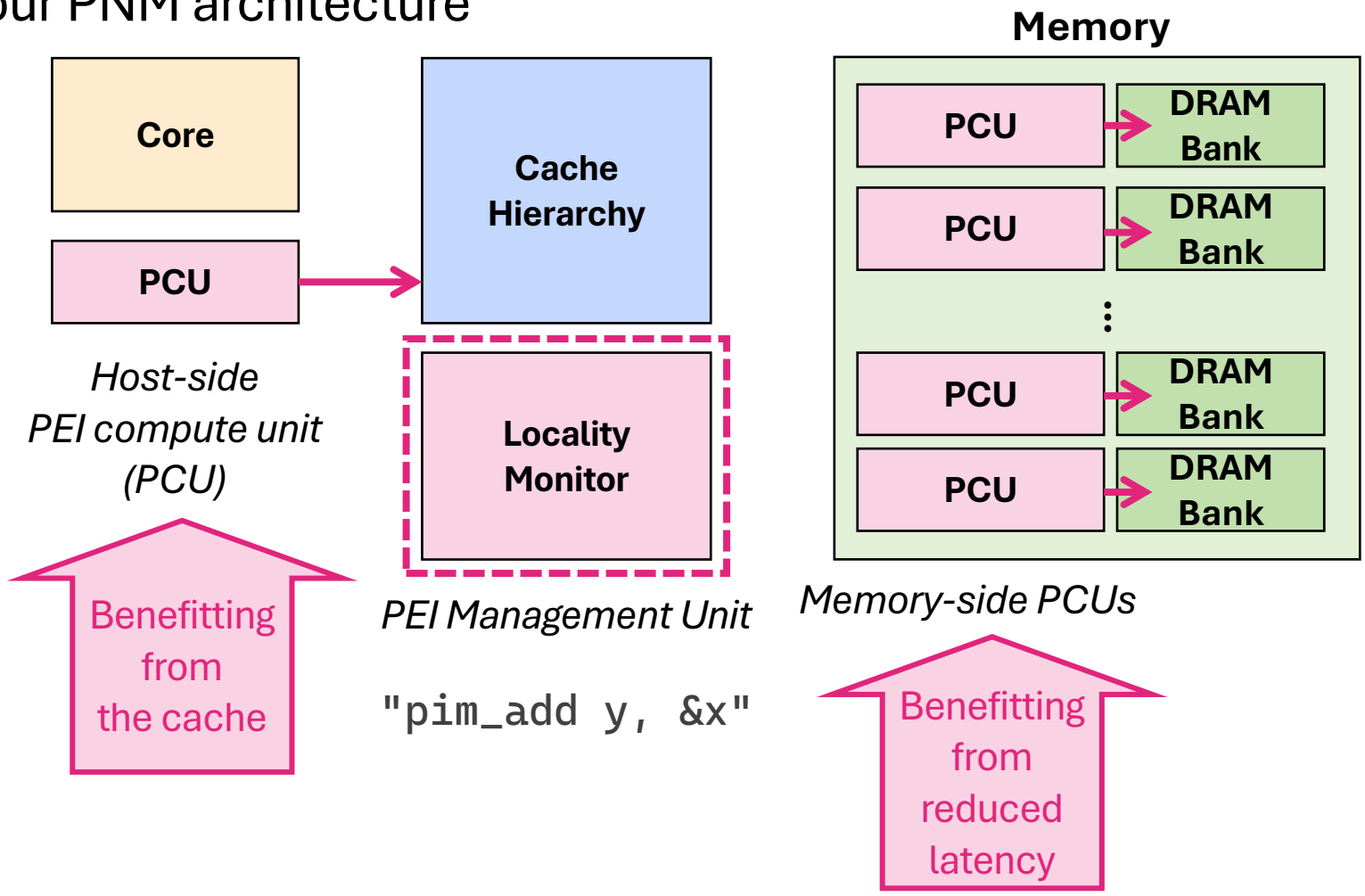
1. IMPACT-PNM Covert Channel

2. IMPACT-PUM Covert Channel

3. PNM-Based Genomic Privacy Attack

PNM Architecture: PIM-Enabled Instructions (PEI)

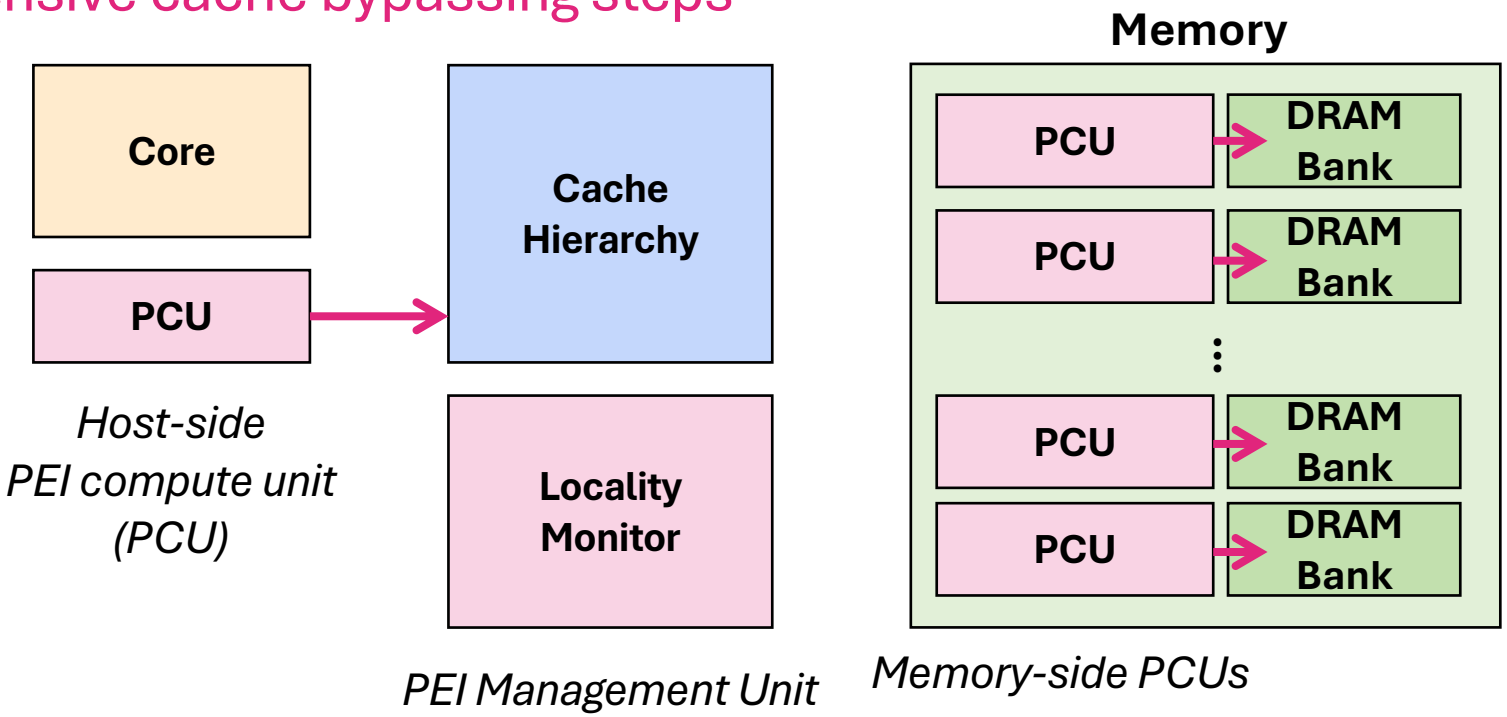
- We assume PIM-Enabled Instructions [Ahn+, ISCA'15] as our PNM architecture



[1] J. Ahn, S. Yoo, O. Mutlu, and K. Choi, "PIM-enabled Instructions: A Low-overhead, Locality-aware Processing-in-memory Architecture," in ISCA, 2015.

IMPACT-PNM: Key Mechanism

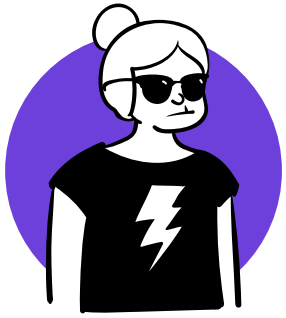
- Offloads attack instructions to a memory-side PCU to eliminate expensive cache bypassing steps




```
"pim_add y, &x"
```

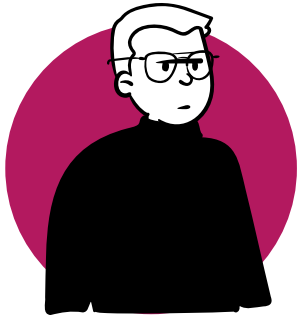

[1] J. Ahn, S. Yoo, O. Mutlu, and K. Choi, "PIM-enabled Instructions: A Low-overhead, Locality-aware Processing-in-memory Architecture," in ISCA, 2015.

IMPACT-PNM Overview



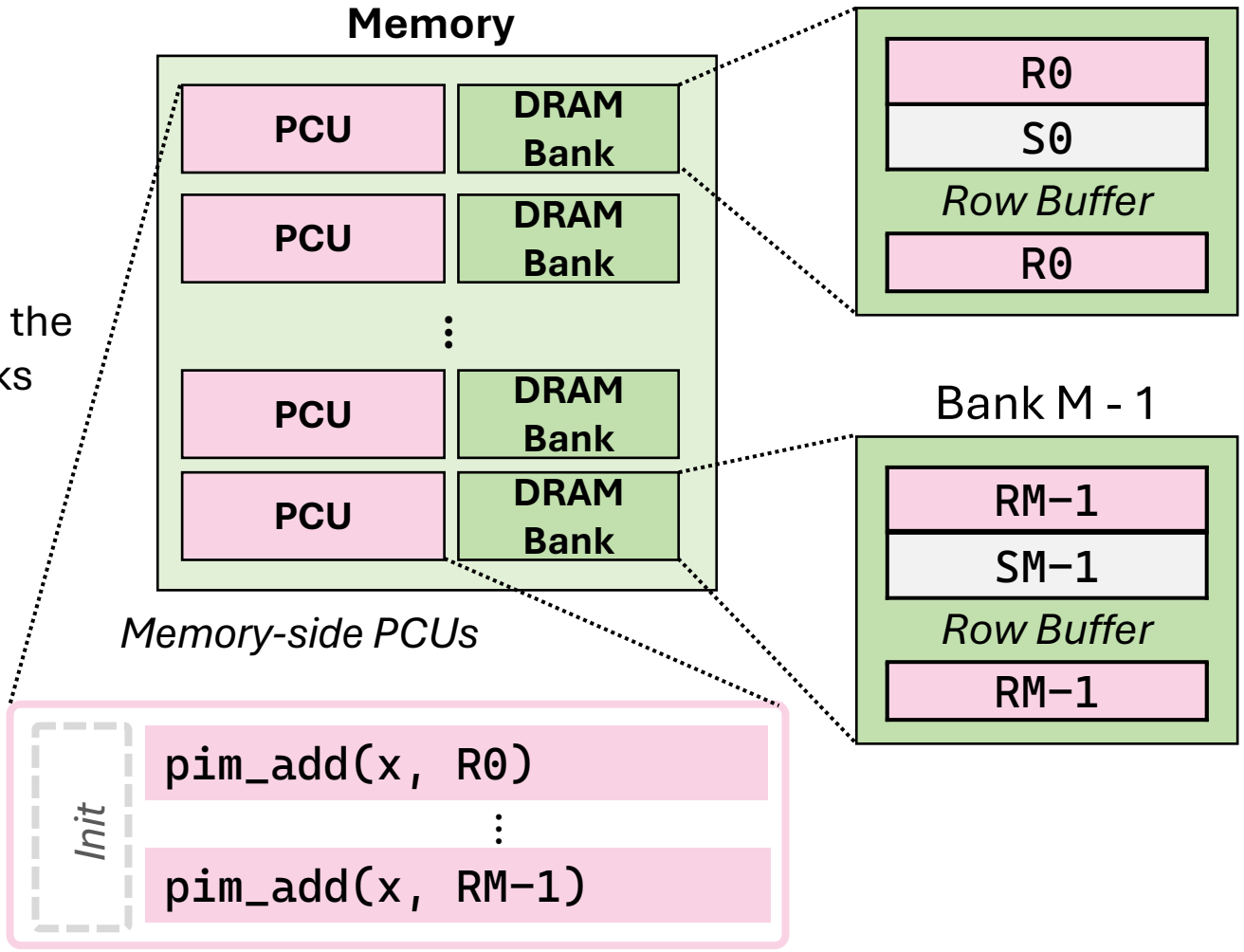
Sender

0 Colocate data in the same set of banks

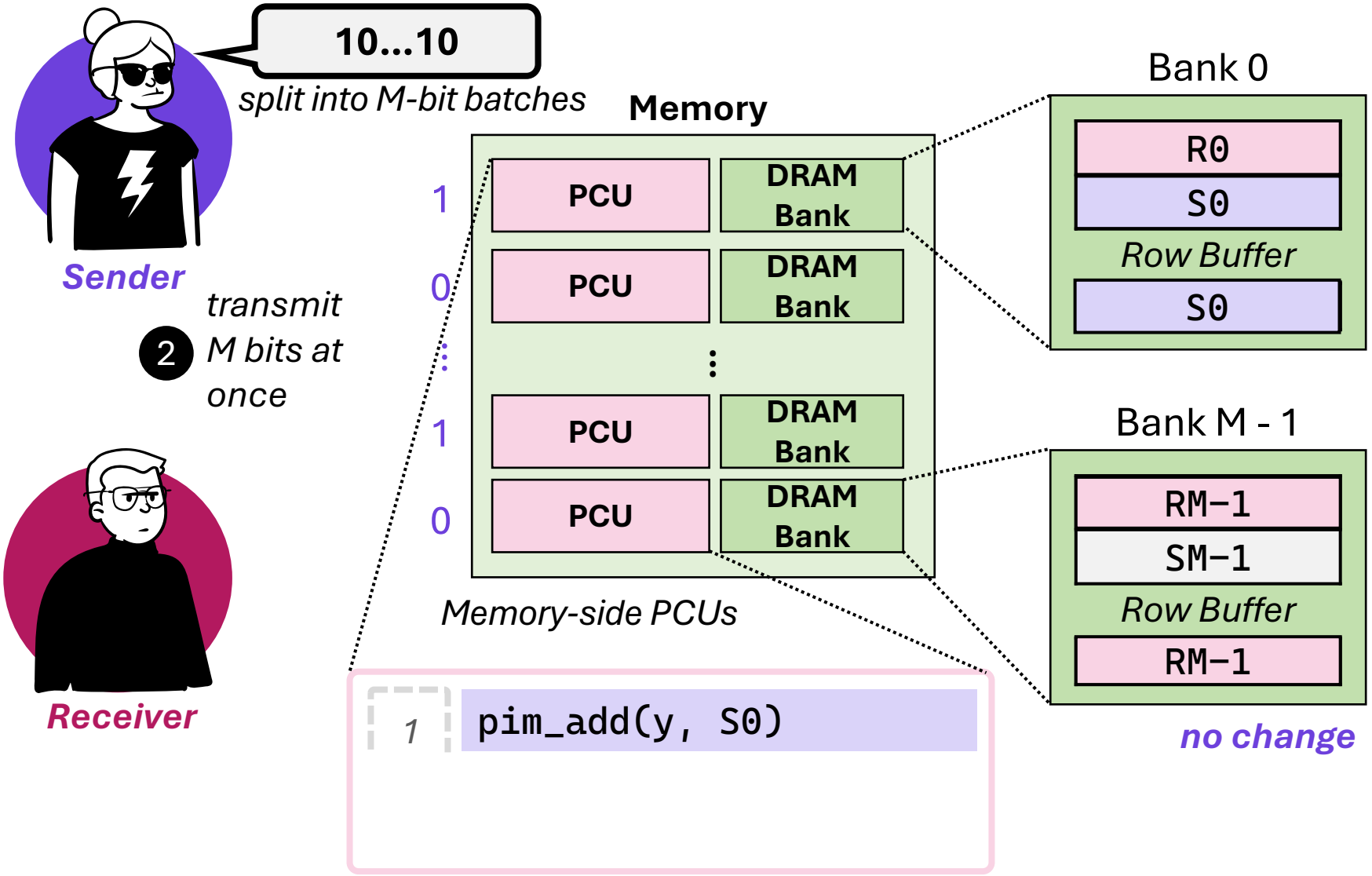


Receiver

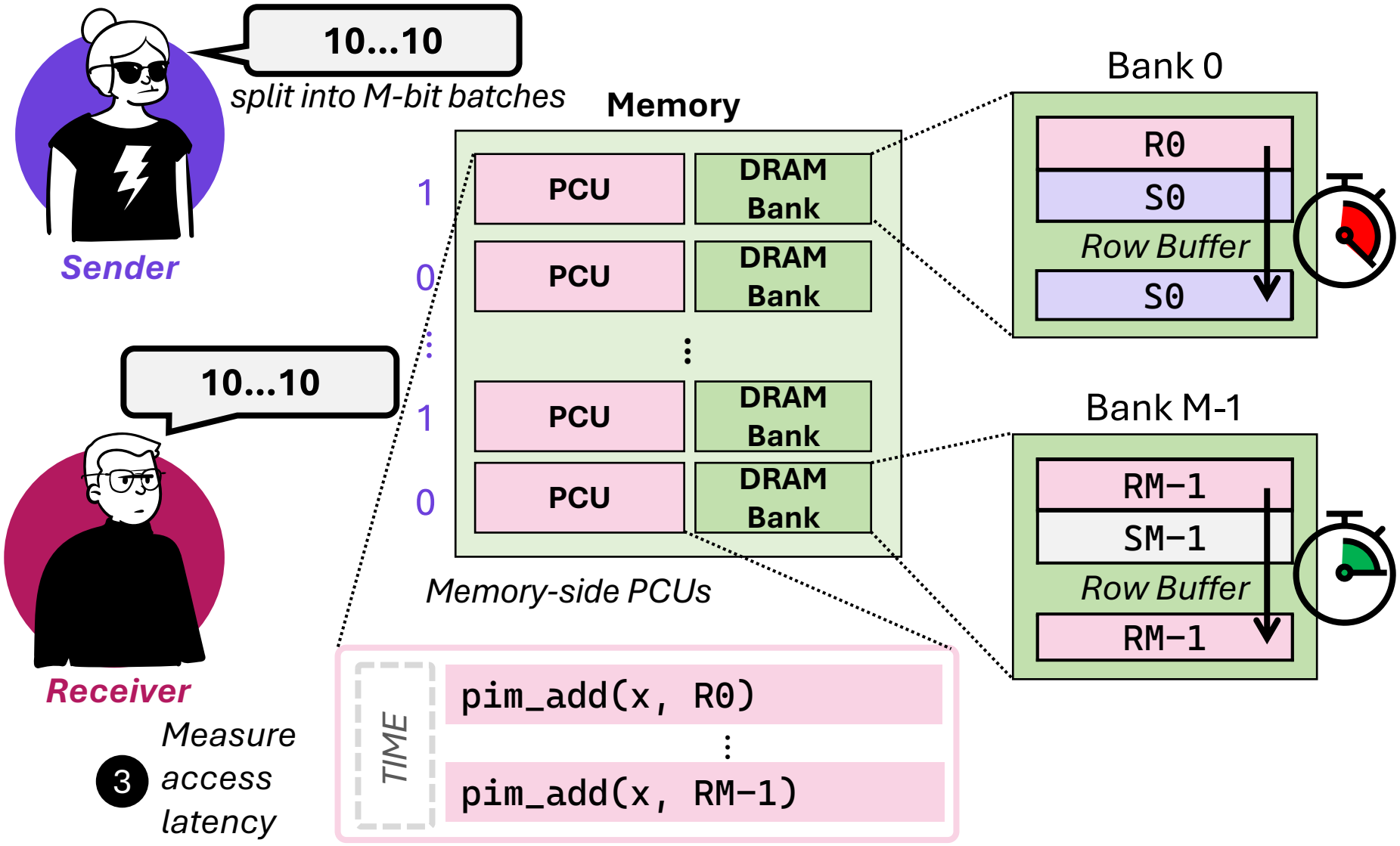
1 *initialize row buffer states*



IMPACT-PNM Overview

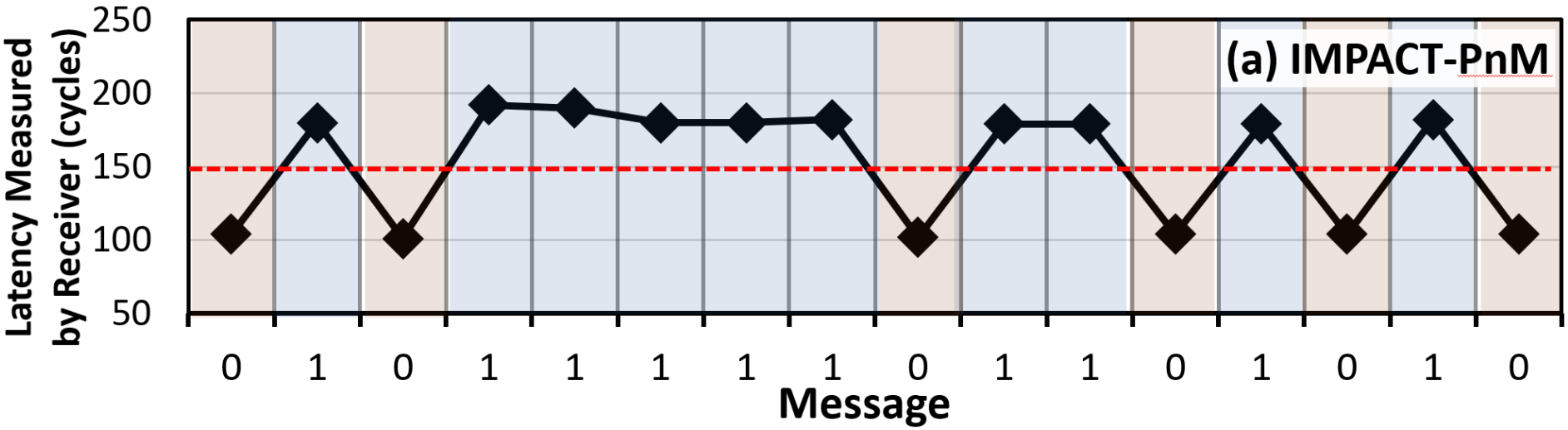


IMPACT-PNM Overview



IMPACT-PNM: Proof-of-Concept Validation

- Sending a 16-bit message using 16 DRAM banks



The receiver successfully determines the transmitted bit values by detecting the row buffer conflicts

IMPACT-PNM: Communication Throughput

- One sender and one receiver process using 16 DRAM banks across increasing LLC sizes:

IMPACT-PNM provides
a high communication throughput of 8.2 Mbps
by eliminating expensive cache bypassing steps
(irrespective of the LLC size)

IMPACT Case Studies

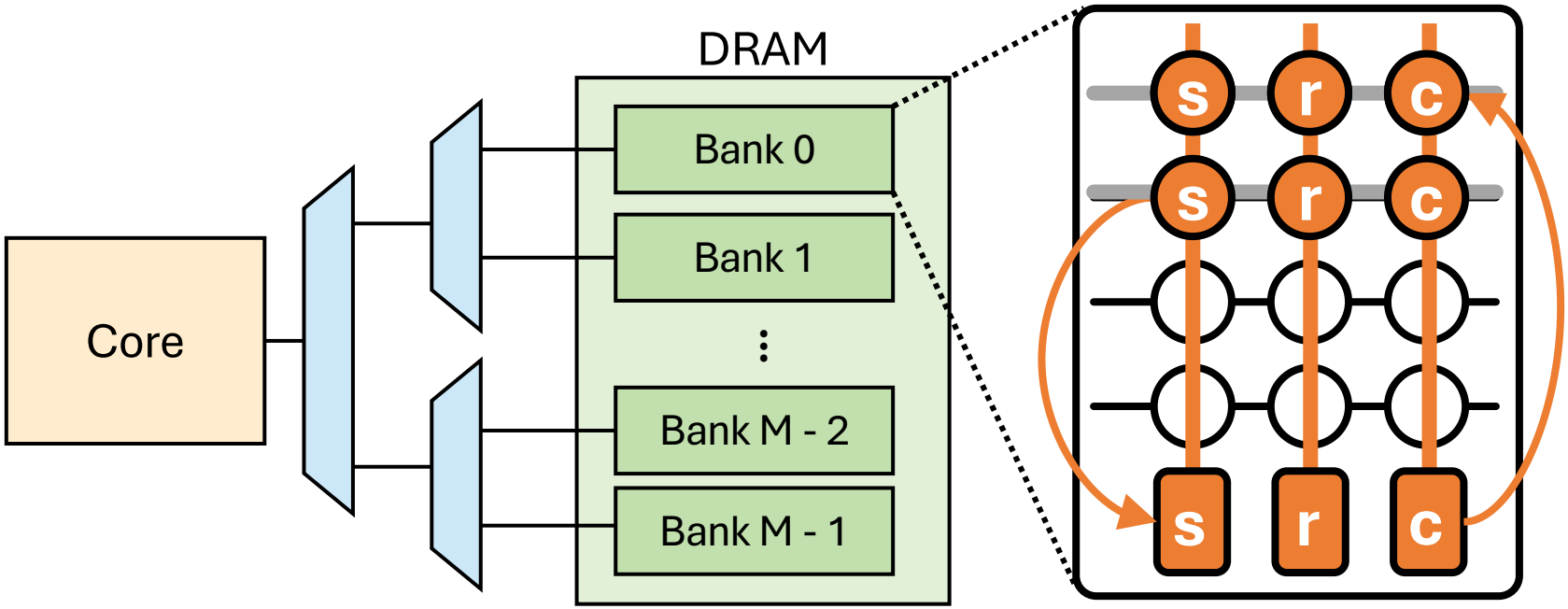
1. IMPACT-PNM Covert Channel

2. IMPACT-PUM Covert Channel

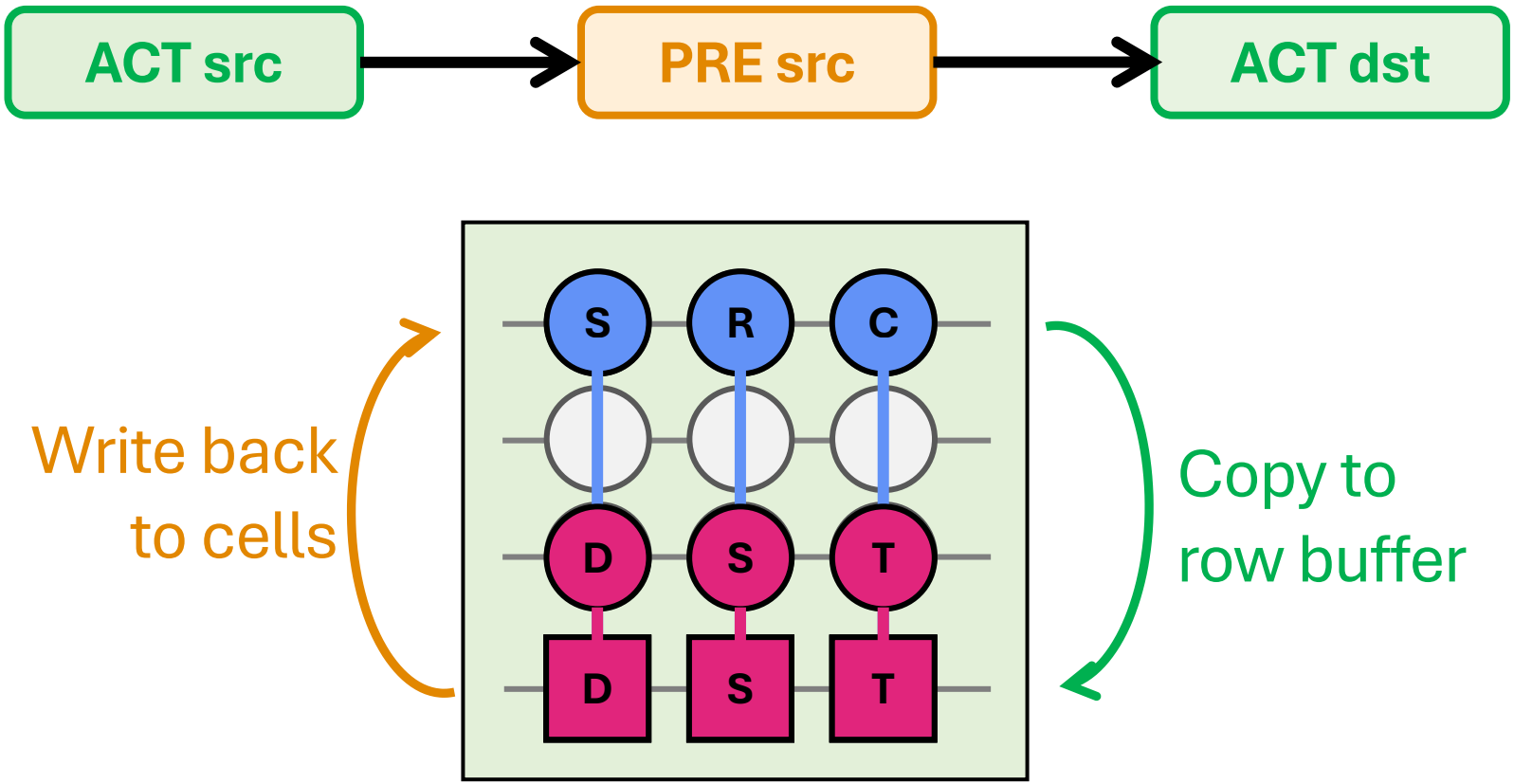
3. PNM-Based Genomic Privacy Attack

PUM Architecture: RowClone

- We assume a PUM architecture that provides userspace applications with RowClone [Seshadri+, MICRO'13]:
  - enables bulk data copy and initialization operations

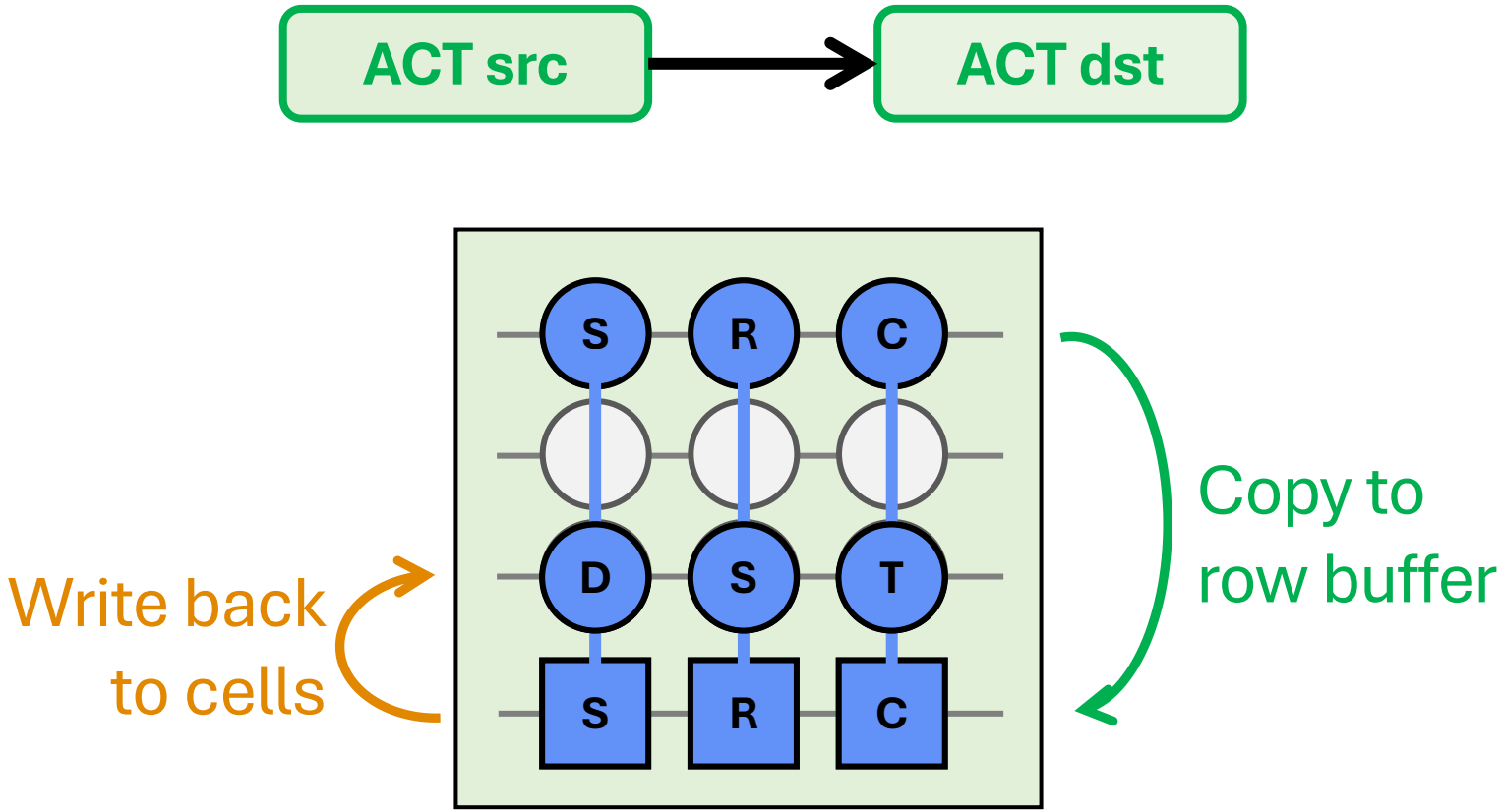


Recall: DRAM operation



[Seshadri+ MICRO'13]

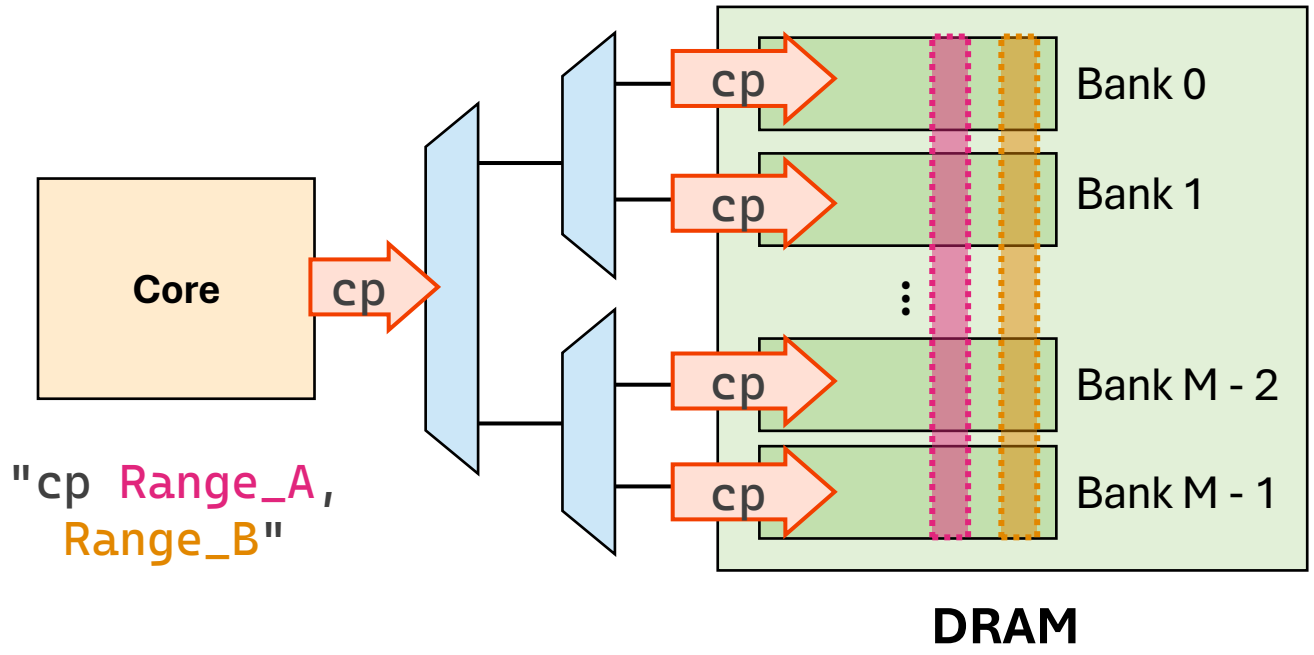
In-DRAM Row-Copy (RowClone)



- Copies the source (src) row's content to the destination (dst) row in-DRAM

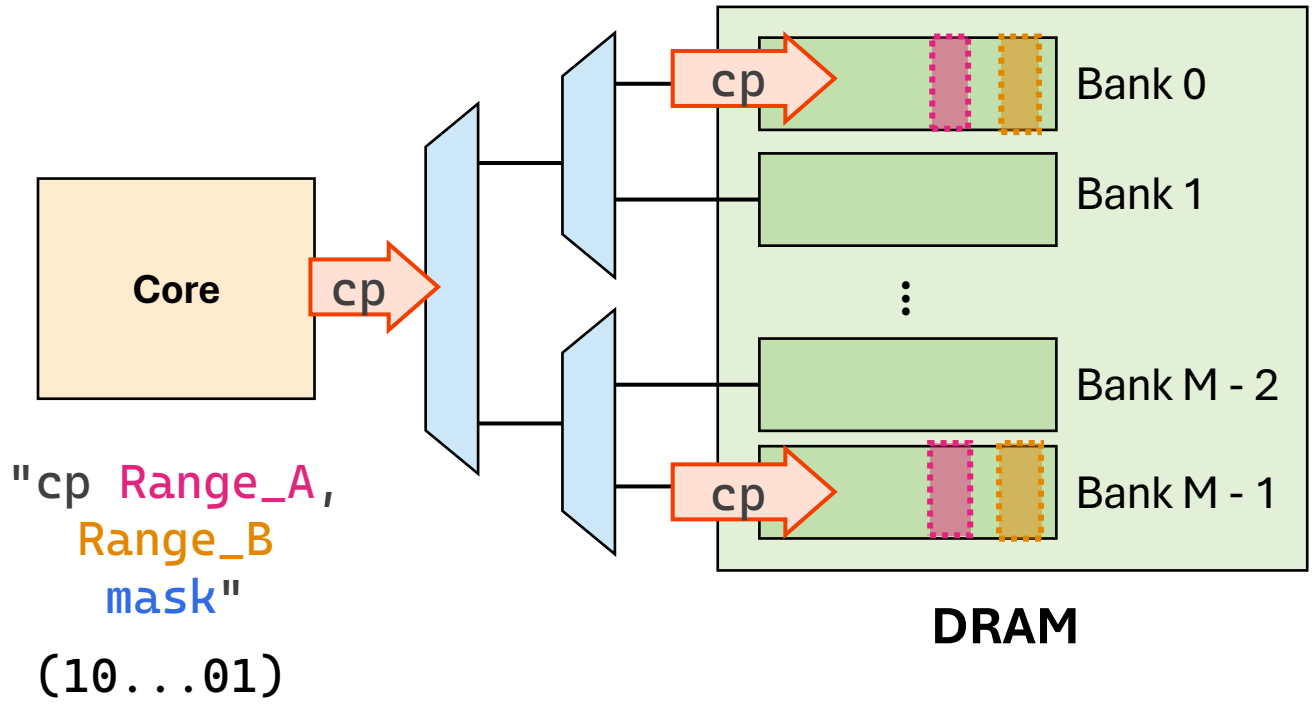
Supporting RowClone

- We assume two copy operations that exploit parallelism:
  - copying ranges across all banks

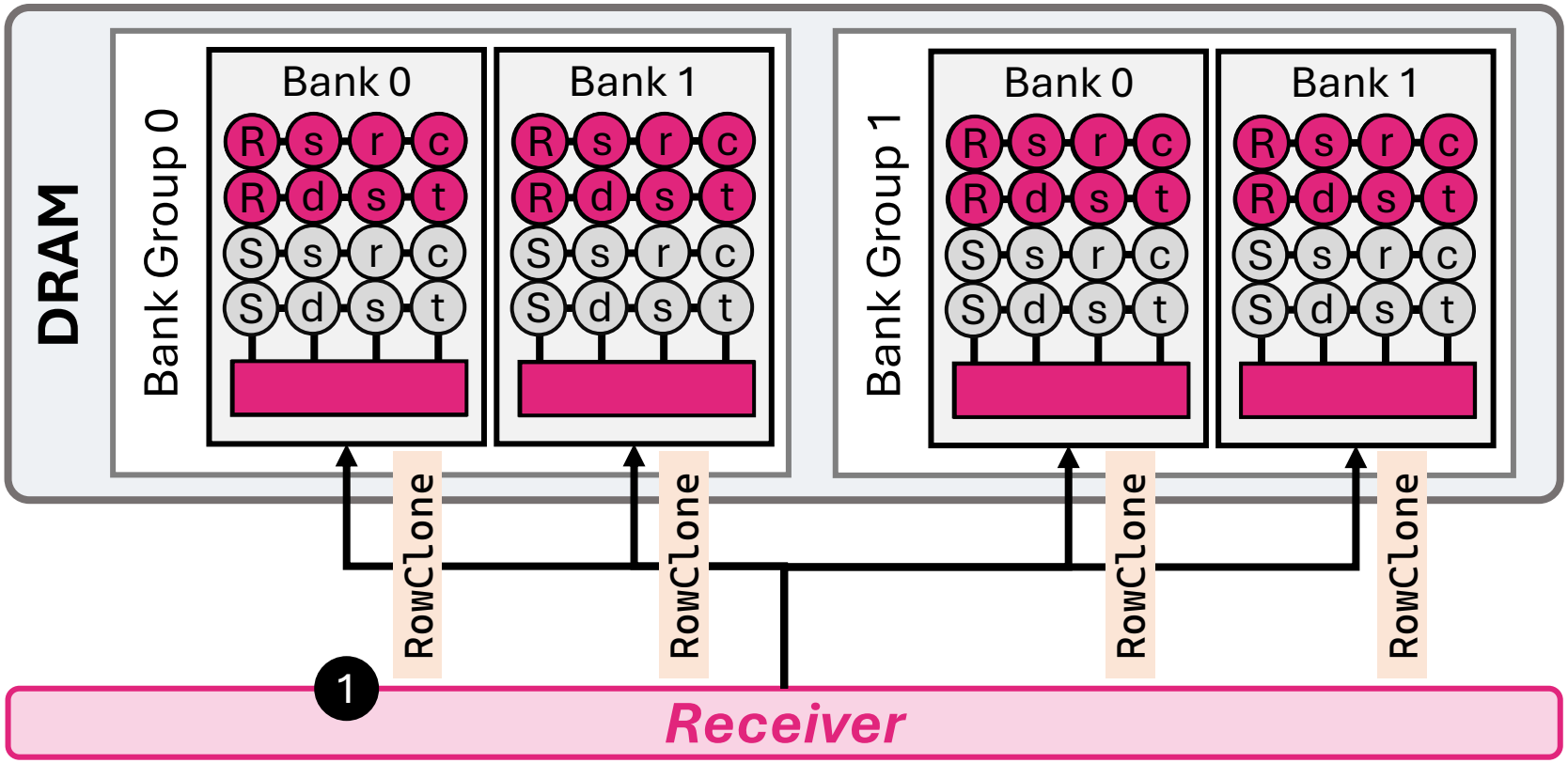


Supporting RowClone

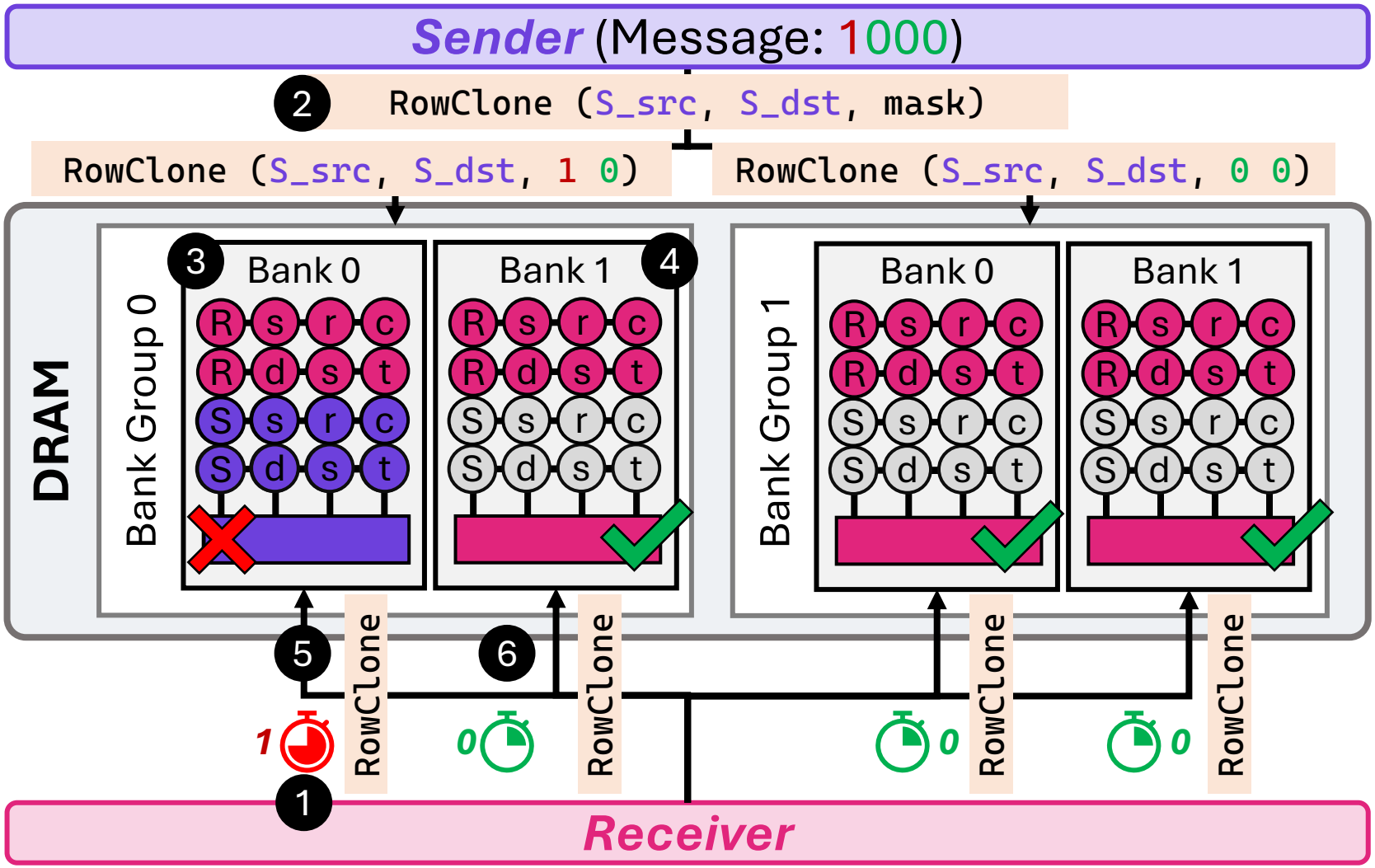
- We assume two copy operations that exploit parallelism:
  - copying ranges across all banks
  - selective copying with a mask operation



IMPACT-PUM Overview

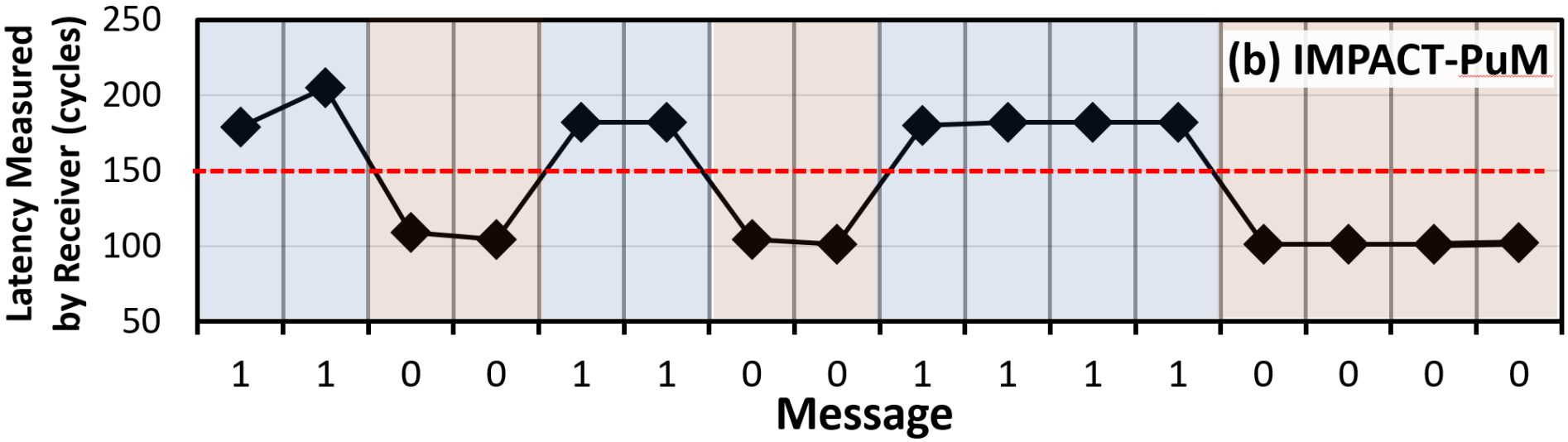


IMPACT-PUM Overview



IMPACT-PUM: Proof-of-Concept Validation

- Sending 16-bit messages using 16 DRAM banks



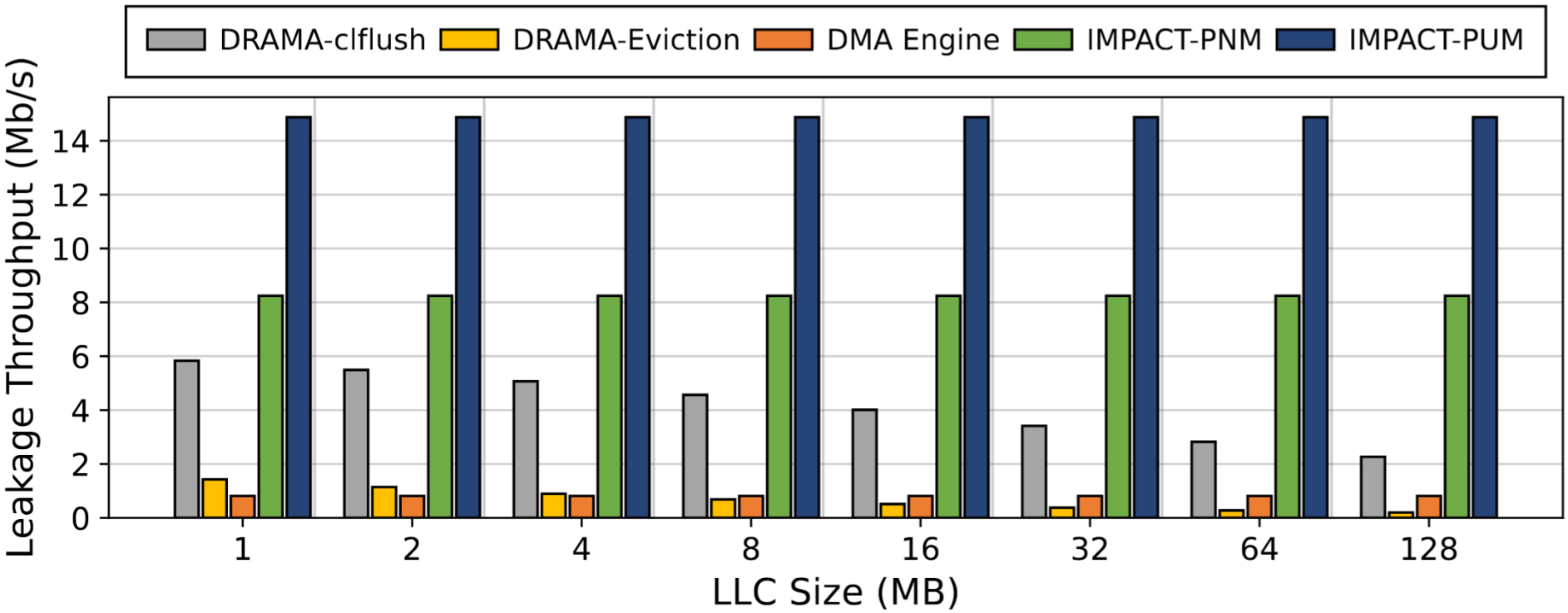
The receiver successfully determines the transmitted bit values by detecting the row buffer conflicts

IMPACT-PUM: Communication Throughput

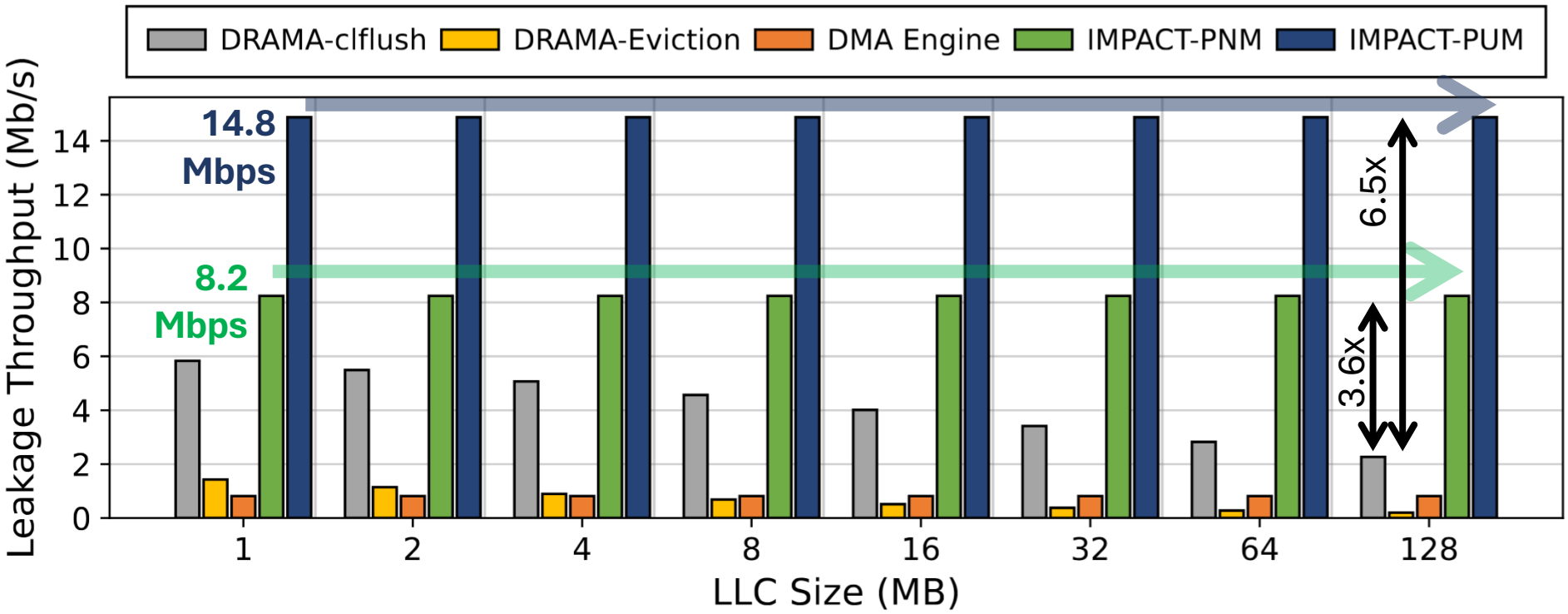
- One sender and one receiver process using 16 DRAM banks across increasing LLC sizes:

IMPACT-PUM provides
a high communication throughput of 14.8 Mbps
by exploiting intrinsic parallelism of processing-using-memory operations
(irrespective of the LLC size)

Covert Channel Throughput Comparison



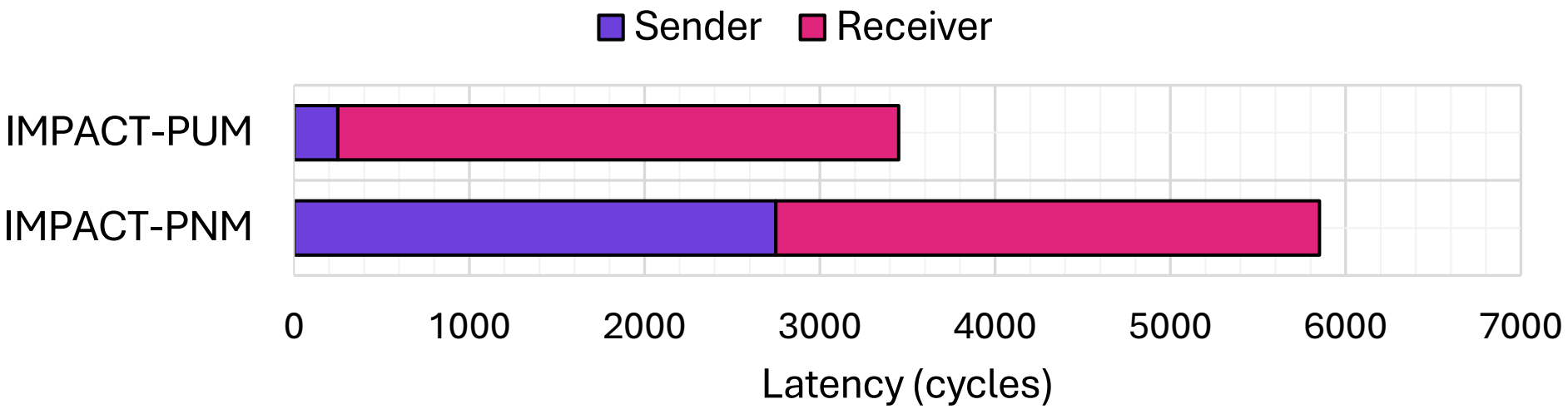
Covert Channel Throughput Comparison



IMPACT-PNM and **IMPACT-PUM** achieve **8.2** and **14.8 Mbps** leakage throughput, respectively, *across all LLC sizes*

IMPACT-PNM and **IMPACT-PUM** outperform **all state-of-the-art main memory-based attacks** across all LLC sizes

IMPACT-PNM and IMPACT-PUM Comparison



IMPACT-PUM's sender routine takes **11.1x less time**
compared to IMPACT-PNM due to exploiting parallelism of PUM operations
and results in higher throughput

IMPACT Case Studies

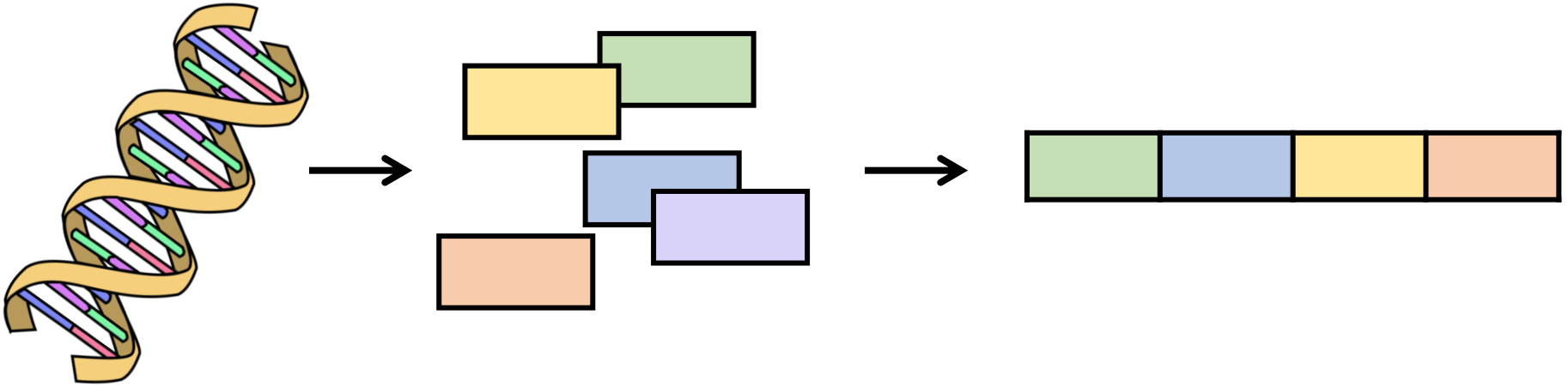
1. IMPACT-PNM Covert Channel

2. IMPACT-PUM Covert Channel

3. PNM-Based Genomic Privacy Attack

PNM-Based Genomics Attack Summary

- DNA read mapping: a fundamental task in genomics

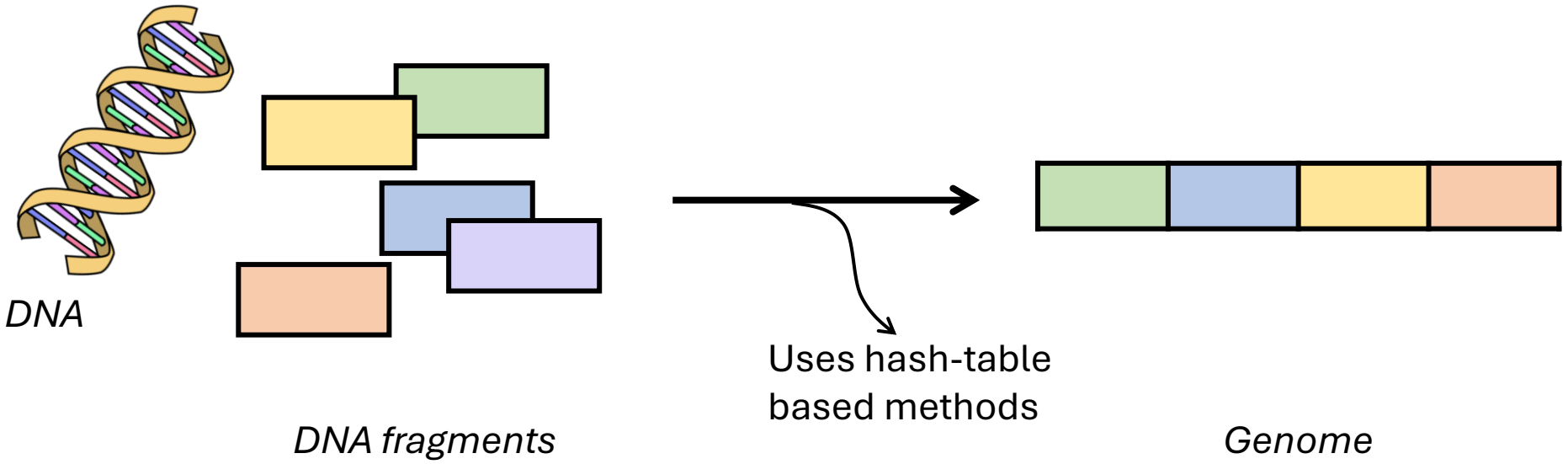


- e.g., Biological research, forensics, diagnostics, drug development, personalized medicine, and agriculture

DNA → Privacy sensitive data

PNM-Based Genomics Attack Summary

- DNA read mapping: a fundamental task in genomics



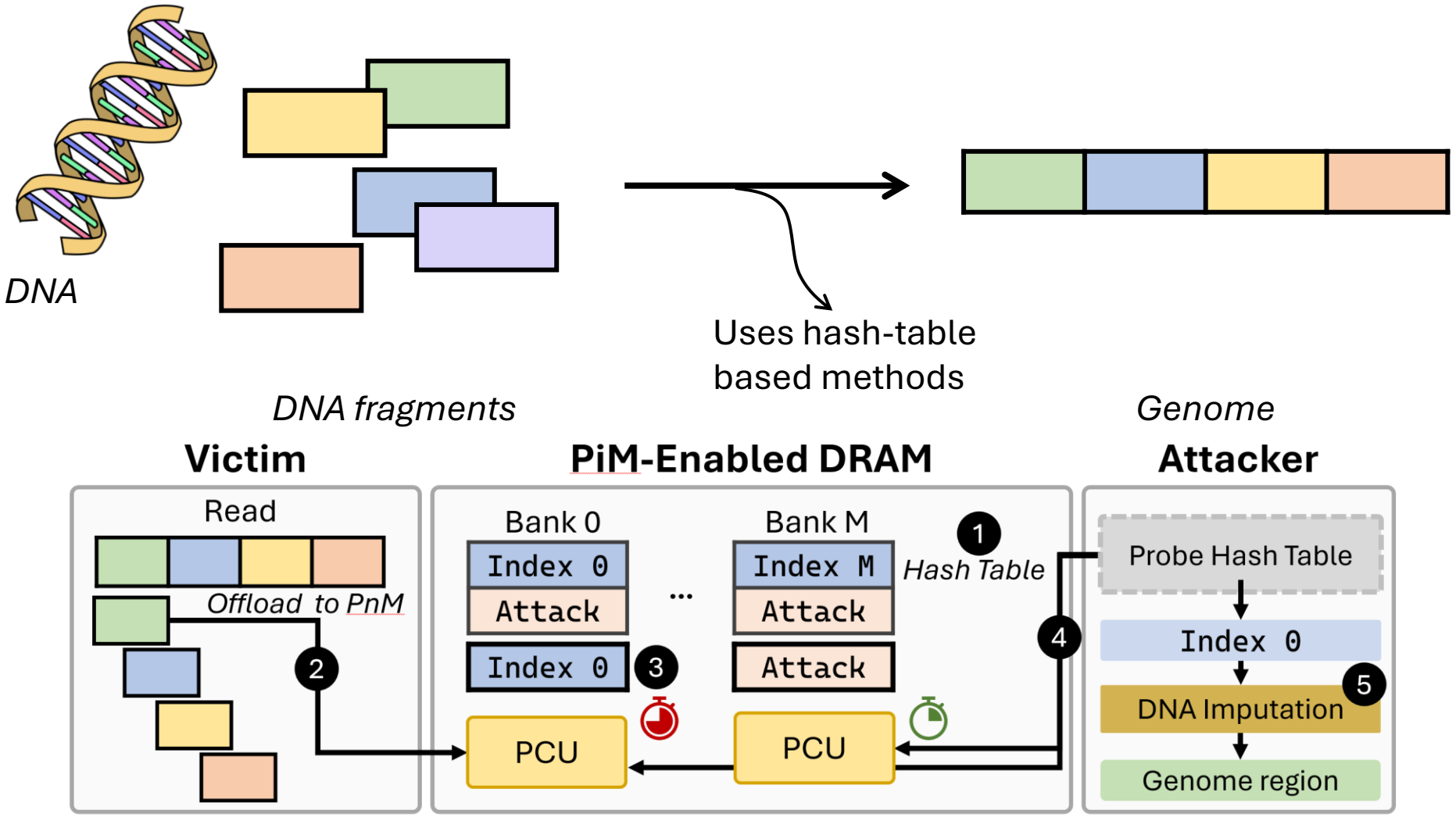
Distributed across DRAM banks

and

accelerated with processing-near-memory architectures

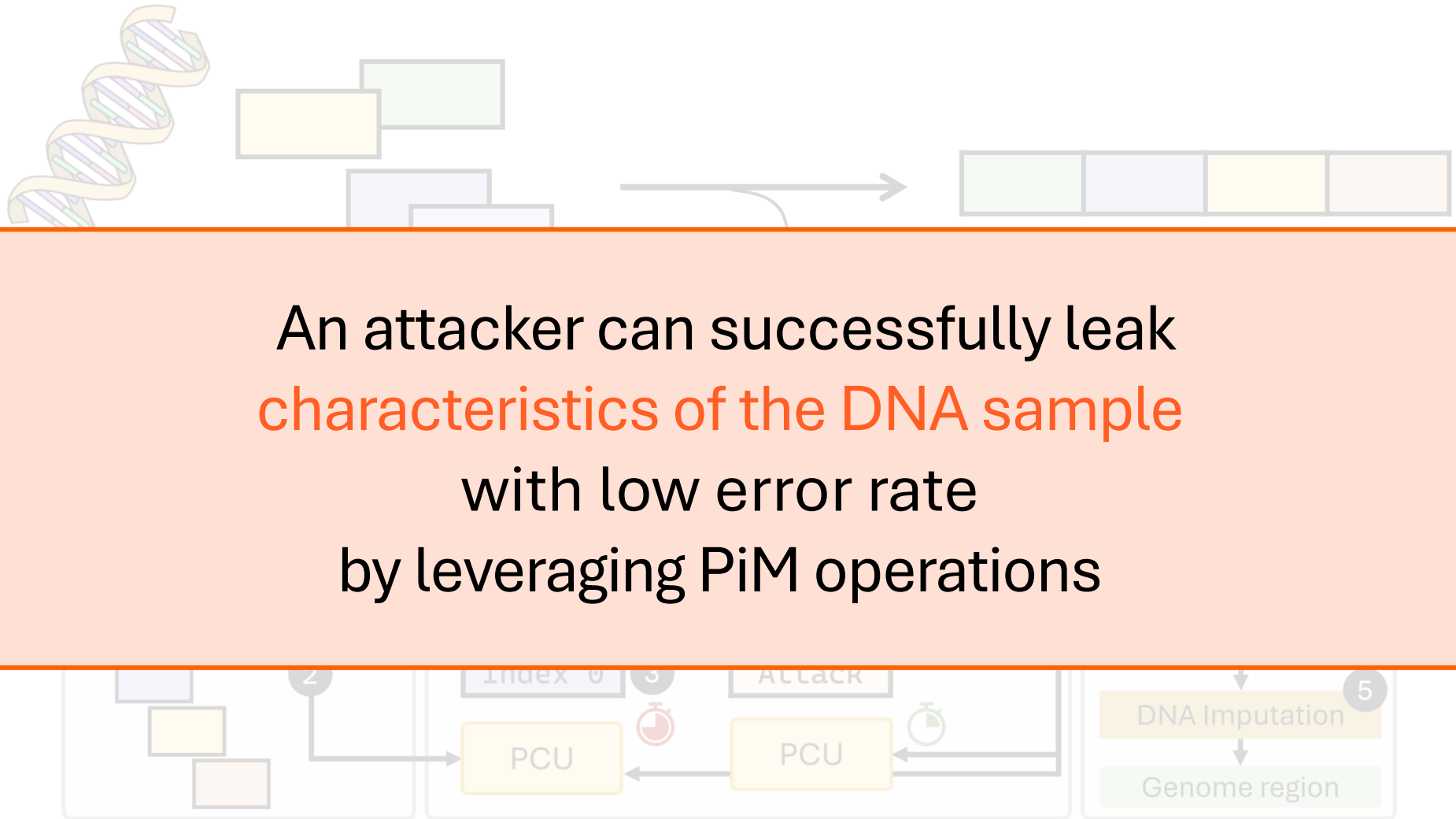
PNM-Based Genomics Attack Summary

- DNA read mapping: a fundamental task in genomics



PNM-Based Genomics Attack Summary

- DNA read mapping: a fundamental task in genomics



An attacker can successfully leak
characteristics of the DNA sample
with low error rate
by leveraging PiM operations

More details in the Paper

Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory

F. Nisa Bostancı[†*] Konstantinos Kanellopoulos[†*] Ataberk Olgun[†]
A. Giray Yağlıkçı[†] İsmail Emir Yüksel[†] Nika Mansouri Ghiasi[†]
Zülal Bingöl[†‡] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†]ETH Zürich [‡]Bilkent University

***Abstract*—We introduce IMPACT, a set of high-throughput main memory-based timing attacks that leverage characteristics of *processing-in-memory* (PiM) architectures to establish covert and side channels. IMPACT enables high-throughput communication and private information leakage by exploiting the shared DRAM row buffer. To achieve high throughput, IMPACT (i) eliminates expensive cache bypassing steps required by processor-centric memory-based timing attacks and (ii) leverages the intrinsic parallelism of PiM operations. We showcase two applications of IMPACT. First, we build two covert channels that leverage different PiM approaches (i.e., *processing-near-memory* and *processing-using-memory*) to establish high-throughput covert communication channels. Our covert chan-**

PiM architectures against timing covert- and side-channel attacks.

In this work, we analyze PiM architectures and show that the adoption of PiM architectures creates opportunities for critical main memory-based timing attacks due to two reasons. First, to eliminate data movement, PiM architectures provide direct access to main memory, which is a key building block for high-throughput main memory-based timing attacks. Second, defenses against these attacks either incur high performance overheads or are *not* applicable to PiM architectures.

1. Direct Access to Main Memory. Main memory-based

https://arxiv.org/abs/2404.11284

IMPACT Case Studies

1. IMPACT-PNM Covert Channel

2. IMPACT-PUM Covert Channel

3. PNM-Based Genomic Privacy Attack

Outline

Motivation and Problem

Key Observation

IMPACT

Mitigating IMPACT

Conclusion

Defenses Against IMPACT

- Bank-level partitioning, closed row policy, constant time memory: restrictive and highly costly
- **Adaptive Constant-Time DRAM (ACT):**
  - **Key Idea:** Only employ constant time memory when there is high interference in a DRAM bank

  - ACT reduces channel capacity but *does not* mitigate IMPACT completely
  - Reducing the channel capacity (to be comparable to the state-of-the-art attack) still induces a high performance overhead

Defenses Against IMPACT

- Bank-level partitioning, closed row policy, constant time memory: restrictive and highly costly
- **Adaptive Constant-Time DRAM (ACT):**
  - **Key Idea:** Only employ constant time memory when there is high
  - Reducing the channel capacity (to be comparable to the state-of-the-art attack) still induces a high performance overhead

Mitigating IMPACT
incurs high performance overhead
and more research is needed
to find low-cost solutions

More in the Paper

- PNM-based genomics side channel attack
  - Operational details
  - Throughput and accuracy analysis
- Mitigations for IMPACT
  - Discussing and evaluating 4 countermeasures
- Discussion
  - Other potential PIM-based attack vectors
  - Applicability of IMPACT to complex PIM architectures and future DRAM devices
  - Restricting access to PIM operations

More in the Paper

Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory

F. Nisa Bostancı[†*] Konstantinos Kanellopoulos[†*] Ataberk Olgun[†]
A. Giray Yağlıkçı[†] İsmail Emir Yüksel[†] Nika Mansouri Ghiasi[†]
Zülal Bingöl[†‡] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†]ETH Zürich [‡]Bilkent University

***Abstract*—We introduce IMPACT, a set of high-throughput main memory-based timing attacks that leverage characteristics of *processing-in-memory* (PiM) architectures to establish covert and side channels. IMPACT enables high-throughput communication and private information leakage by exploiting the shared DRAM row buffer. To achieve high throughput, IMPACT (i) eliminates expensive cache bypassing steps required by processor-centric memory-based timing attacks and (ii) leverages the intrinsic parallelism of PiM operations. We showcase two applications of IMPACT. First, we build two covert channels that leverage different PiM approaches (i.e., *processing-near-memory* and *processing-using-memory*) to establish high-throughput covert communication channels. Our covert chan-**

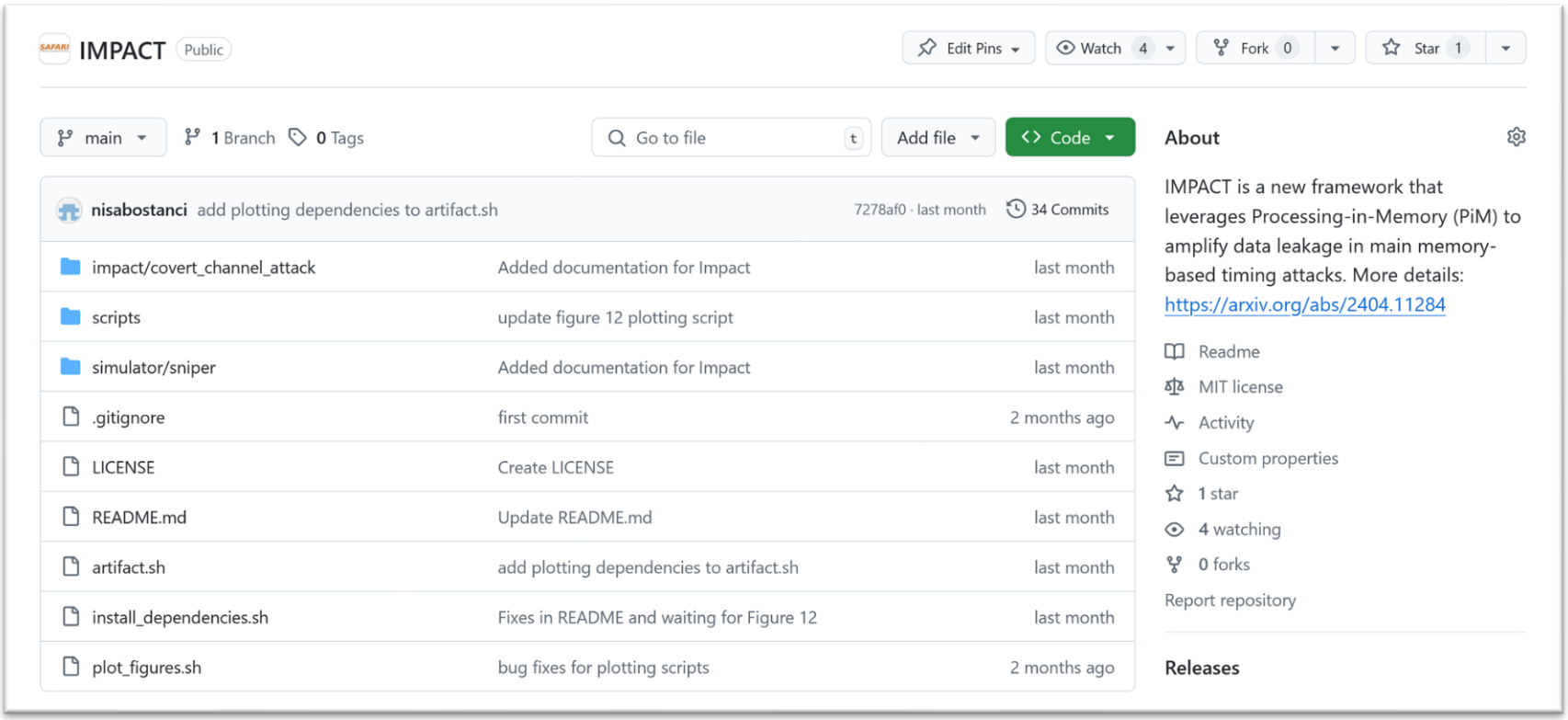
PiM architectures against timing covert- and side-channel attacks.

In this work, we analyze PiM architectures and show that the adoption of PiM architectures creates opportunities for critical main memory-based timing attacks due to two reasons. First, to eliminate data movement, PiM architectures provide direct access to main memory, which is a key building block for high-throughput main memory-based timing attacks. Second, defenses against these attacks either incur high performance overheads or are *not* applicable to PiM architectures.

1. Direct Access to Main Memory. Main memory-based

https://arxiv.org/abs/2404.11284

IMPACT is Open Source and Artifact Evaluated



https://github.com/CMU-SAFARI/IMPACT

Outline

Motivation and Problem

Key Observation

IMPACT

Mitigating IMPACT

Conclusion

Conclusion

Key Observation: PIM architectures create opportunities for critical main memory-based timing attacks due to two reasons :

- PIM architectures provide direct access to main memory, which is a key building block for high-throughput main memory-based timing attacks
- Defenses against these attacks are highly costly or are not applicable to PIM architectures

IMPACT: a set of high-throughput **I**n-**M**emory **P**rocessing-based timing **At**tacks that leverage direct and fast main memory accesses enabled by PiM architectures

IMPACT achieves high throughput by:

- eliminating expensive cache bypassing steps used in main memory-based timing attacks
- leveraging the intrinsic parallelism of PIM operations

Case Studies:

- Demonstrate **two covert channel attacks** leveraging different PIM architectures, achieving high throughput communication
- Showcase **a side-channel attack** that leaks private information of concurrently running victim applications with low error rate

Mitigating IMPACT: We discuss and evaluate **four different countermeasures** against IMPACT, eventually concluding that mitigating IMPACT incurs high performance overheads

Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory

F. Nisa Bostancı K. Kanellopoulos
A. Olgun A. G. Yaglikci I. E. Yuksel
N. Mansouri Ghiasi Z. Bingol M. Sadrosadati O. Mutlu

Paper Link

GitHub Repo

SAFARI

ETH zürich