ShieldCXL: A Practical Obliviousness Support with Sealed CXL Memory

Kwanghoon Choi, Igjae Kim, Sunho Lee, Jaehyuk Huh

ACM TACO

School of Computing, KAIST

KAIST

CASYS | KAIST Computer Architecture & System Lab

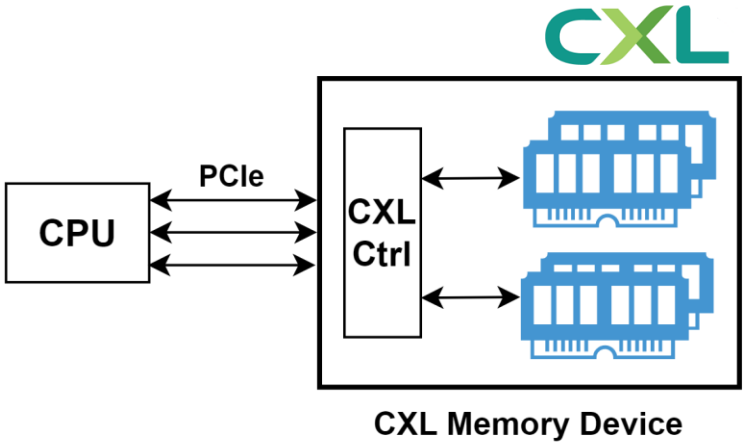
Contents

- Background
- Motivation
- Design
- Evaluation
- Conclusion

CXL – Promising Memory Interface

- Compute Express Link (CXL)
  - Bandwidth and capacity expansion by CXL memory
  - High access latency
  - CXL memory in heterogeneous memory system
- **Memory security is a critical requirement!**

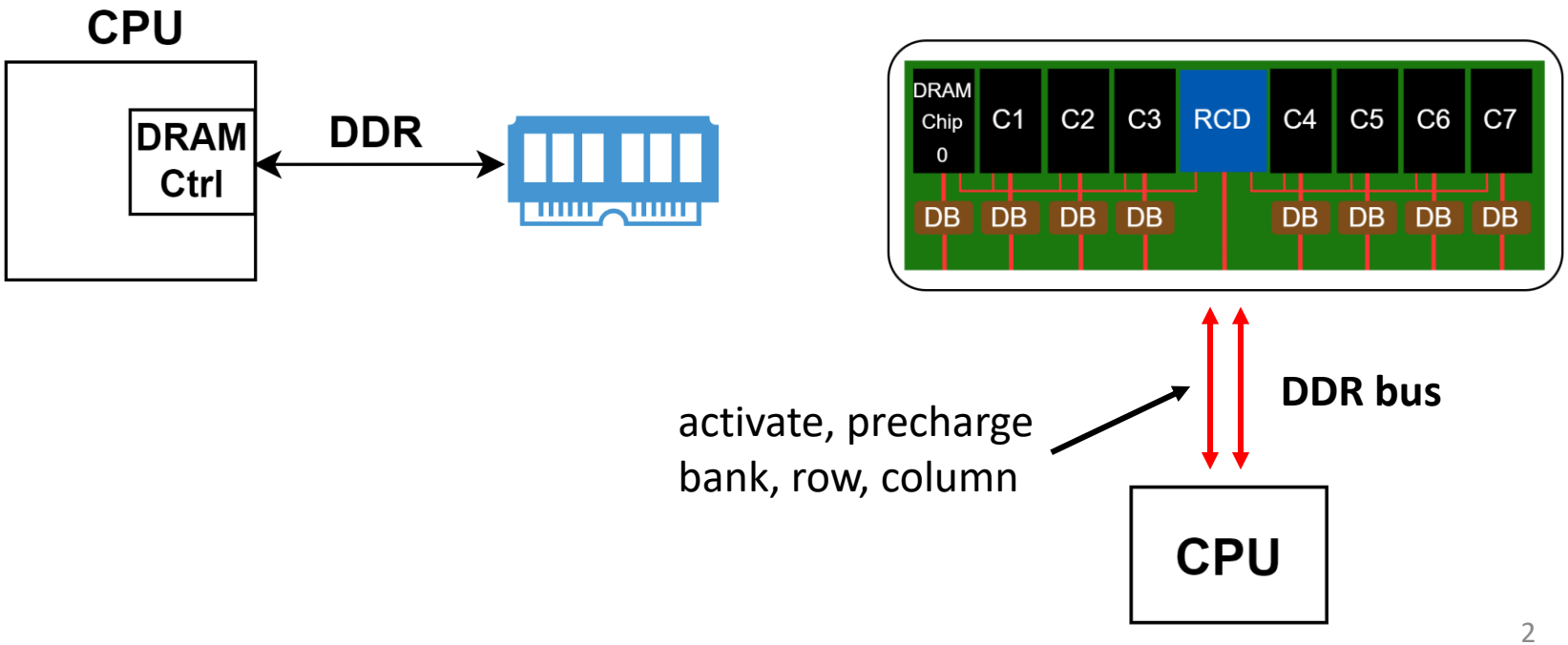
DDR-attached Memory



Memory expansion with CXL

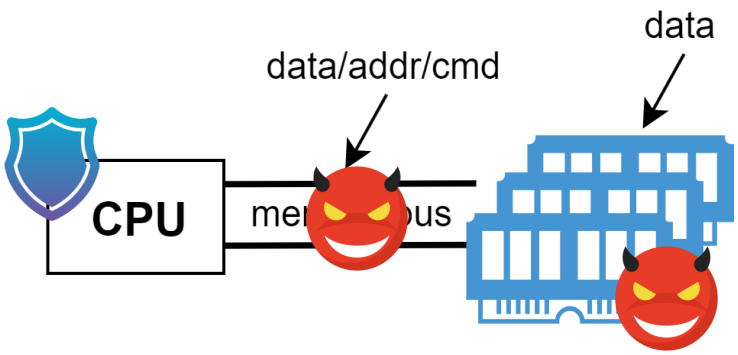
DDR Memory Interface

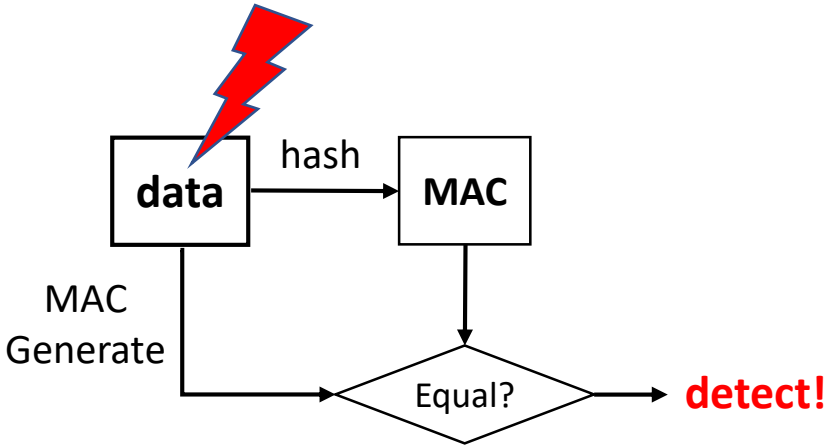
- DDR protocol: low level command
  - Command, control, address, clock
- Attack surface: DDR bus + on-DIMM interconnect



HW-based Memory Protection

- Confidentiality – steal the data
  - Attacks: Bus snooping, row-hammer*
  - Solution: encryption
- Integrity – modify the data
  - Attacks: Bus tampering, row-hammer
  - Solution: MAC (Message Authentication Code)

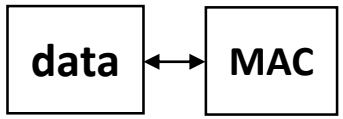




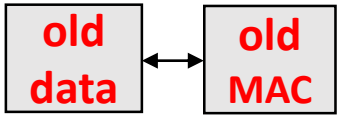
*Andrew Kwong, et al. "RAMBleed: Reading Bits in Memory Without Accessing Them", S&P'20

Freshness for DDR Interface

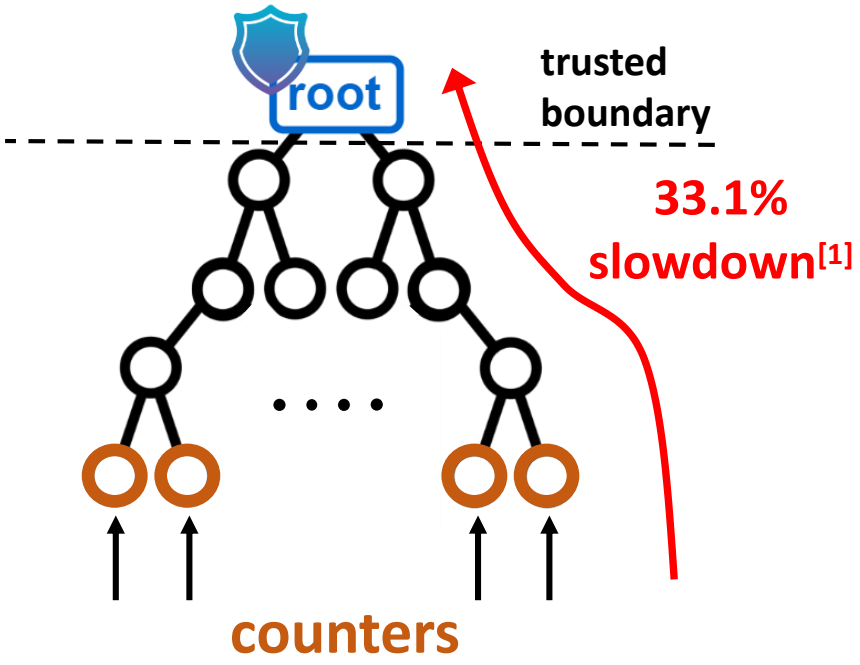
- Freshness – replay attack
  - Replay the old data
  - Solution: data encryption with authenticated counter
  - Counter authentication: integrity tree



MAC authentication succeeds



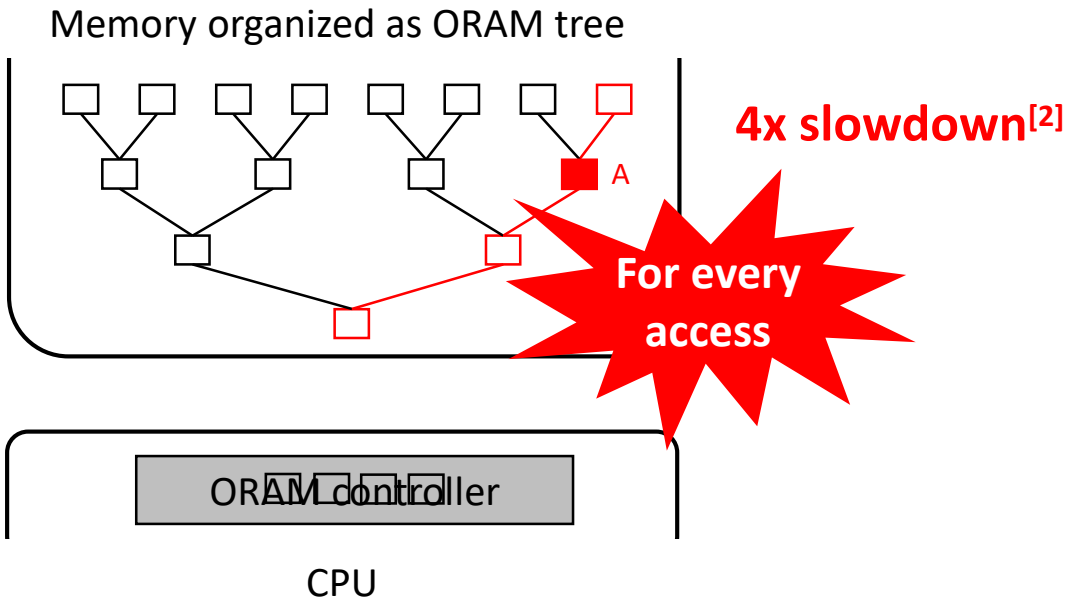
MAC authentication succeeds as well!



[1] M. Taassori, et al. "VAULT: Reducing Paging Overheads in SGX with Efficient Integrity Verification Structures", ASPLOS'18

Access Obfuscation for DDR Interface

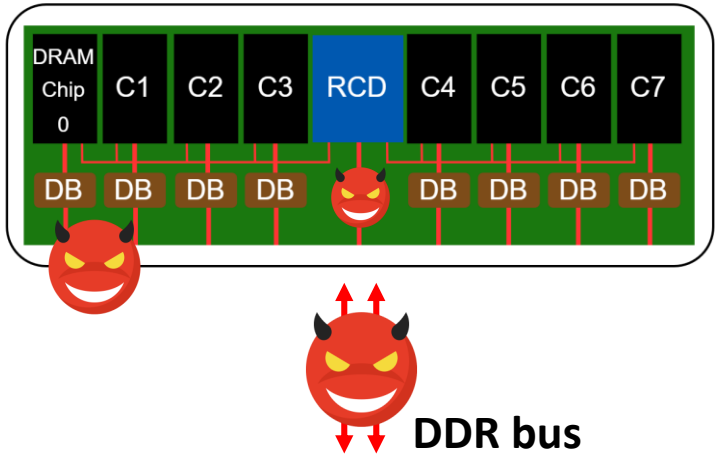
- Access obfuscation – access pattern leakage
  - When and which blocks were read/written
- ORAM[1]: memory shuffling and data re-encryption



[1] Emil Stefanov, et al. “Path ORAM: An Extremely Simple Oblivious RAM Protocol”, CCS’13
[2] Ling Ren, et al. “Constants Count: Practical Improvements to Oblivious RAM”, USENIX Security’15

Access Obfuscation in DDR

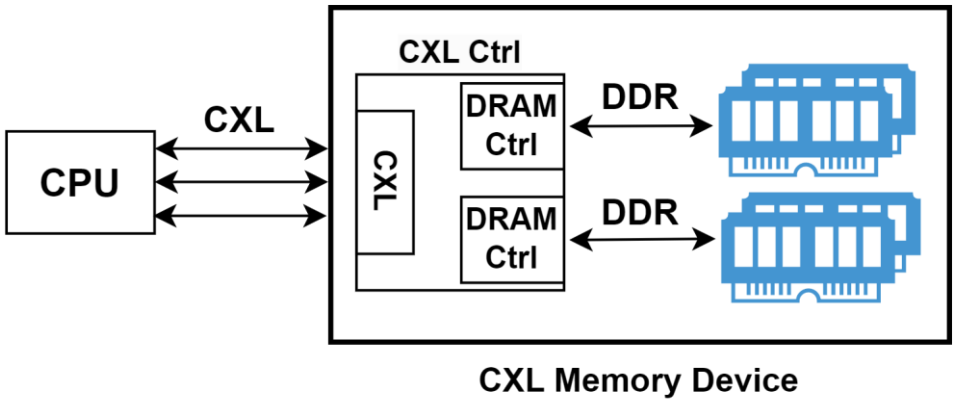
- Access obfuscation – access pattern leakage
  - When and which blocks were read/written
- ORAM[1]: memory shuffling and data re-encryption
- Address/command/data encryption -> **not available in DDR**

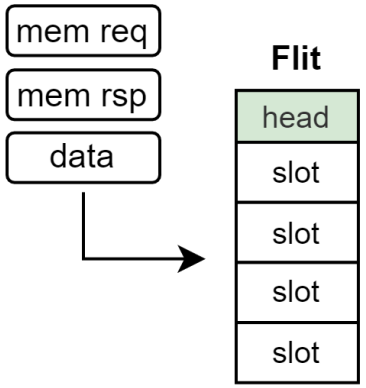


[1] Emil Stefanov, et al. “Path ORAM: An Extremely Simple Oblivious RAM Protocol”, CCS’13
[2] Ling Ren, et al. “Constants Count: Practical Improvements to Oblivious RAM”, USENIX Security’15

CXL Interface

- Interfaces in CXL memory system
  - CPU-CXL: CXL channel based on PCIe
  - Inside CXL: DDR interface
- Flow control unit (flit) – the basic unit of transfer in CXL
  - Fixed size of 68B
  - Multiple packets in a flit <--> DDR low level interface

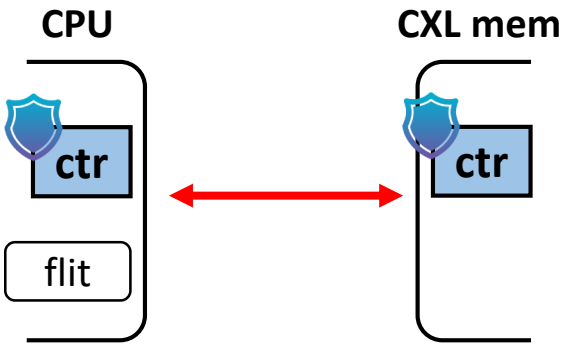




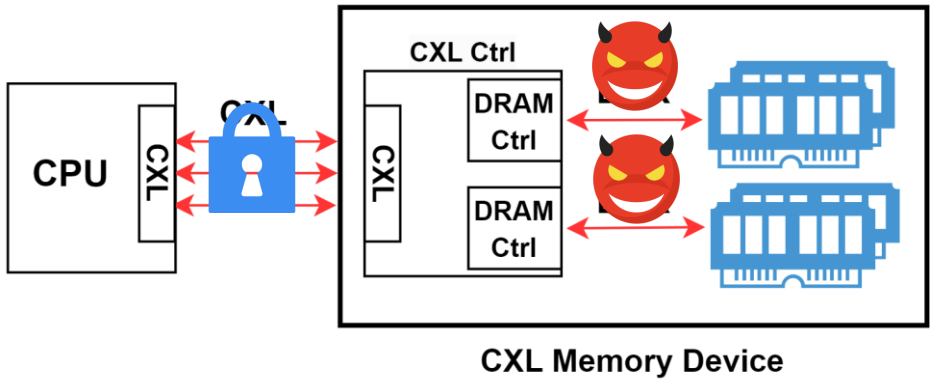
Flit construction

CXL Interface Protection Schemes

- CXL integrity and data encryption (CXL IDE)
  - Flit counter encryption & MAC
  - Freshness without integrity tree, but limited to CXL channel only
- **Protection for DDR interface on CXL board is still needed**



Flit protection in CXL channel

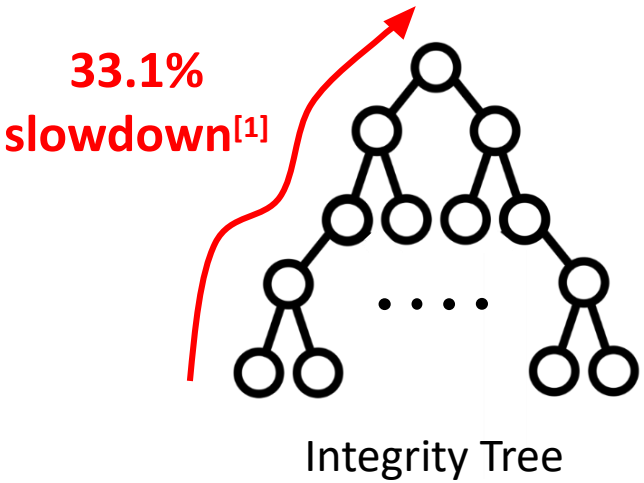


CXL memory system

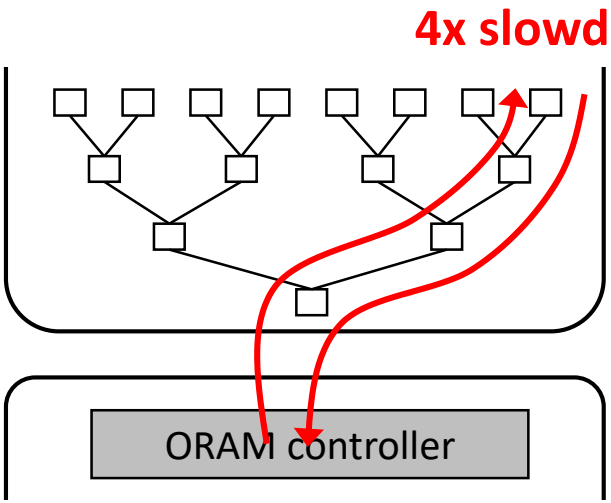
Contents

Overhead of Protection Schemes

- Integrity tree
  - Tree traversal from leaf to root for every memory access
  - Abandoned since Scalable SGX
- ORAM
  - Shuffling and reorganizing data for every memory access
  - Not used in real world



Integrity Tree

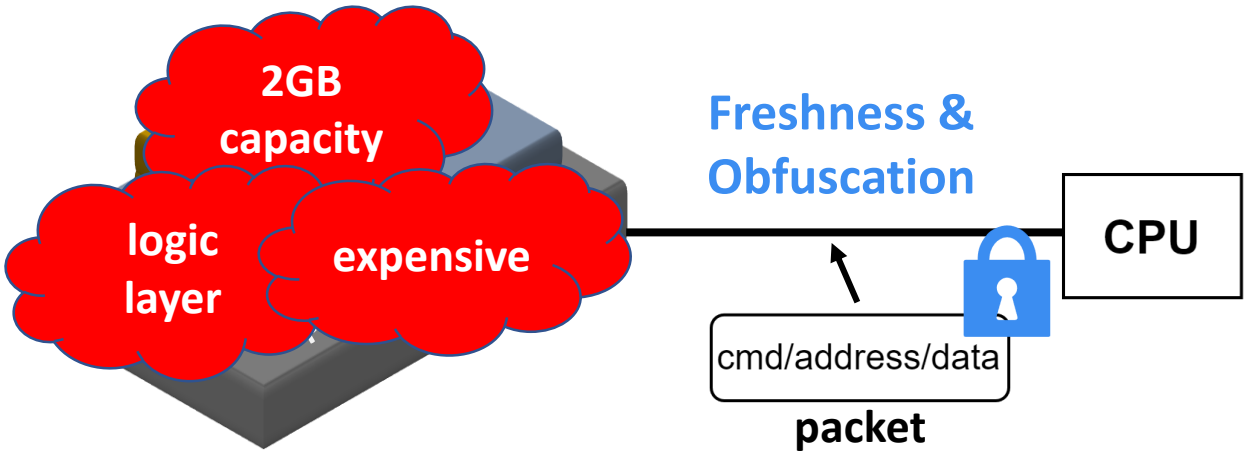


ORAM

[1] M. Taassori, et al. "VAULT: Reducing Paging Overheads in SGX with Efficient Integrity Verification Structures", ASPLOS'18
[2] Ling Ren, et al. "Constants Count: Practical Improvements to Oblivious RAM", USENIX Security'15

Opportunity in CXL Interface

- CXL interface: flit-based communication
  - CXL enables different memory protection techniques
- What if DDR protection scheme is not needed any more?
- Smart memory based approach[1, 2] -> **discontinued**



[1] Shaizeen Aga, et al. “Invisimem: Smart memory defenses for memory bus side channel”, ISCA’17
[2] Amro Awad, et al. “Obfusmem: A low-overhead access obfuscation for trusted memories”, ISCA’17

Key Insights

Problem

Current protection schemes incurs large overhead to CXL memory

Key insights

Use CXL interface only: enabling a totally different protection scheme

Our Goal

This work proposes
HW-based memory protection
for CXL-only memory system with low overhead

Challenges

C1) How to protect DDR interface on CXL memory board?

C2) How to protect CXL channel from physical attack?

C3) How to safeguard data stored in memory?

C4) How to mitigate increased latency to CXL-only memory?

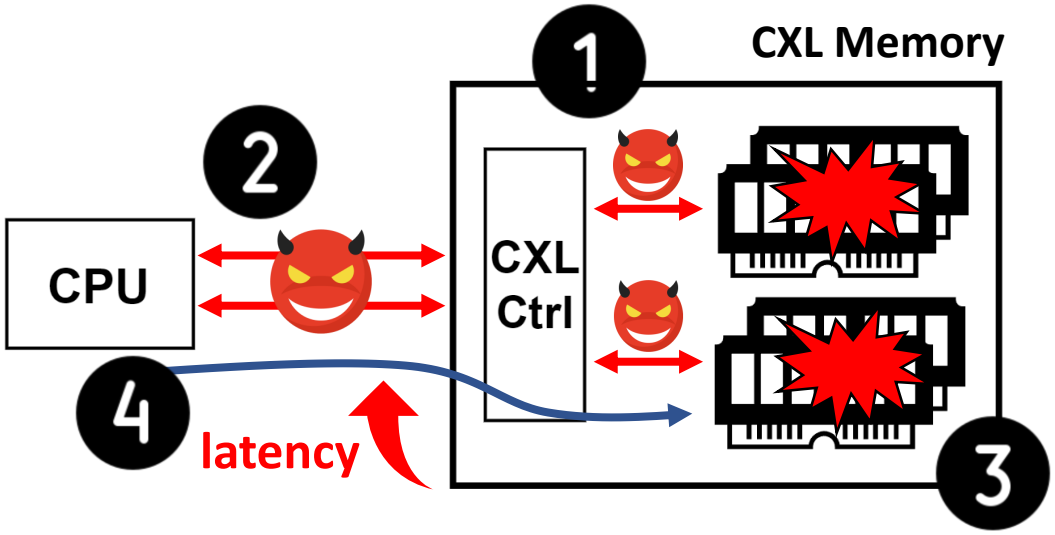
Challenges

C1) How to protect DDR interface on CXL memory board?

C2) How to protect CXL channel from physical attack?

C3) How to safeguard data stored in memory?

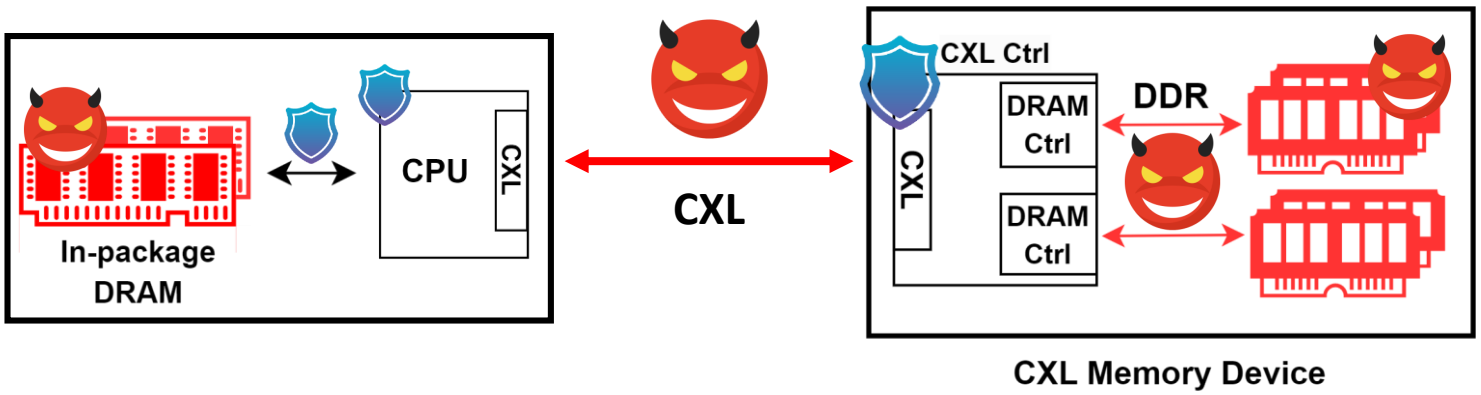
C4) How to mitigate increased latency to CXL-only memory?



Contents

Threat Model

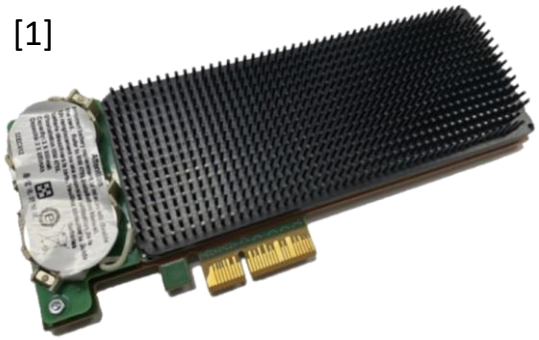
- Trusted Computing Base (TCB)
  - On-chip component of CPU & CXL
- Threat Model
  - Snooping/tampering on DDR interface and CXL channel
  - Untrusted DRAM die
  - In-package DRAM cache side-channel attacks



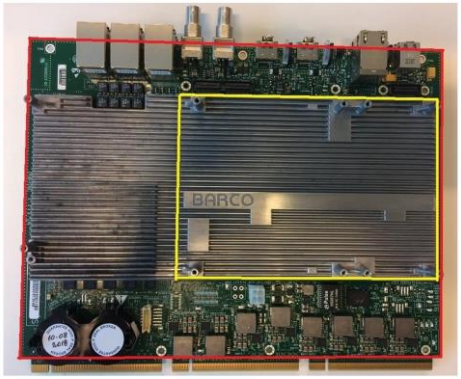
Tamper-responding Sealing

- DDR interface on CXL board is still exposed to attackers
- **Intrusion-detectable sealing which hinders physical access**
- Widely employed on cryptographic modules
  - Applied on critical section for security
  - PCIe, media card, USB, etc.

[1]



[2]



[3]

[1] CEX7S / 4769 PCIe Cryptographic Coprocessor, IBM
[2] Barco n.v. Integrated Cinema Media Processor, Barco ICMP
[3] SecureUSB KP, SecureData Inc

Applying sealing to CXL memory

PCIe Cryptographic Module

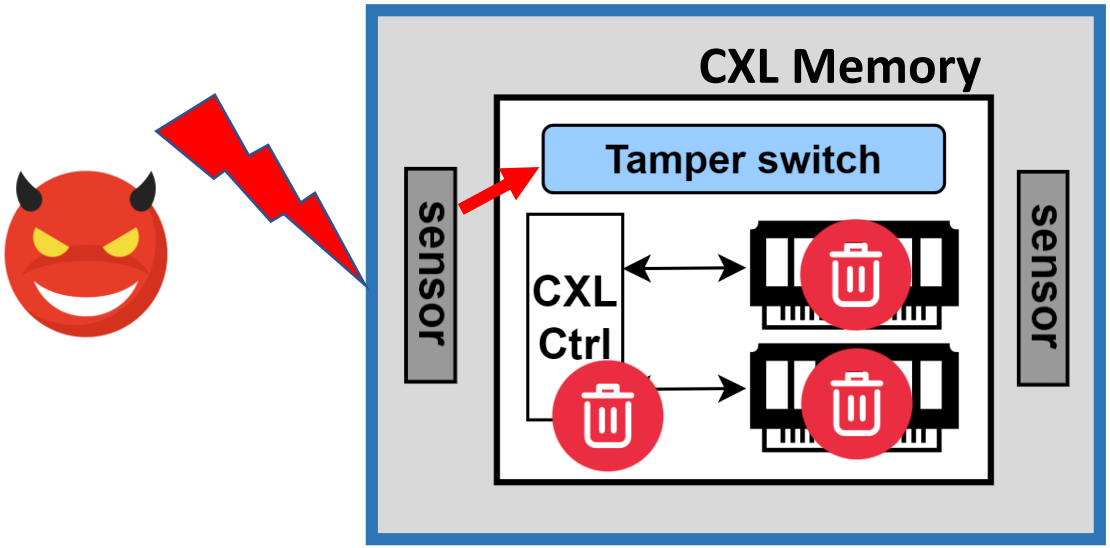
CXL Memory Module

Sealing can be applied in a similar manner!

Tamper-responding CXL Memory

Propose: protect CXL module with tamper-responding sealing

1. Intrusion-detectable hard enclosure for CXL memory
2. Tampering detection with sensors on the enclosure
3. Tamper switch activation
4. Zeroization of security parameters



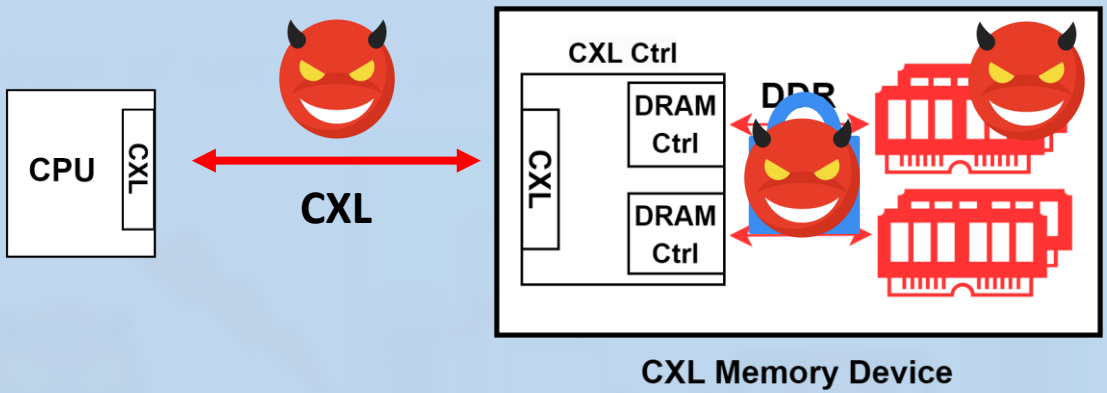
Tamper-responding CXL Memory

Challenges for security

C1) How to protect DDR interface on CXL memory board?

C2) How to protect CXL channel from physical attack?

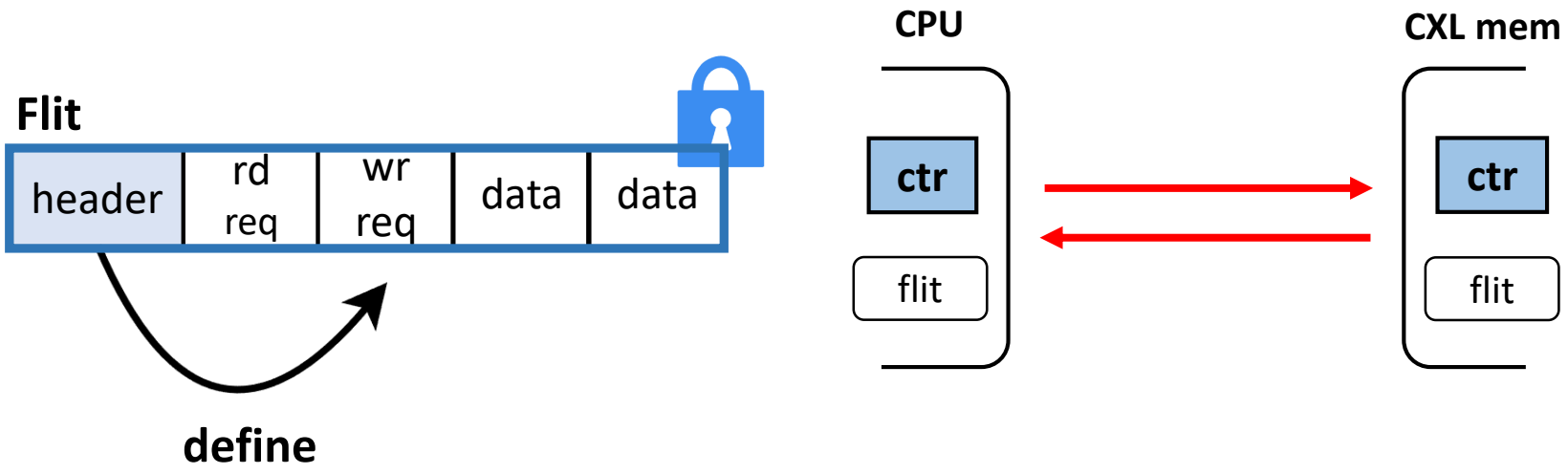
C3) How to safeguard data stored in memory?



Contents

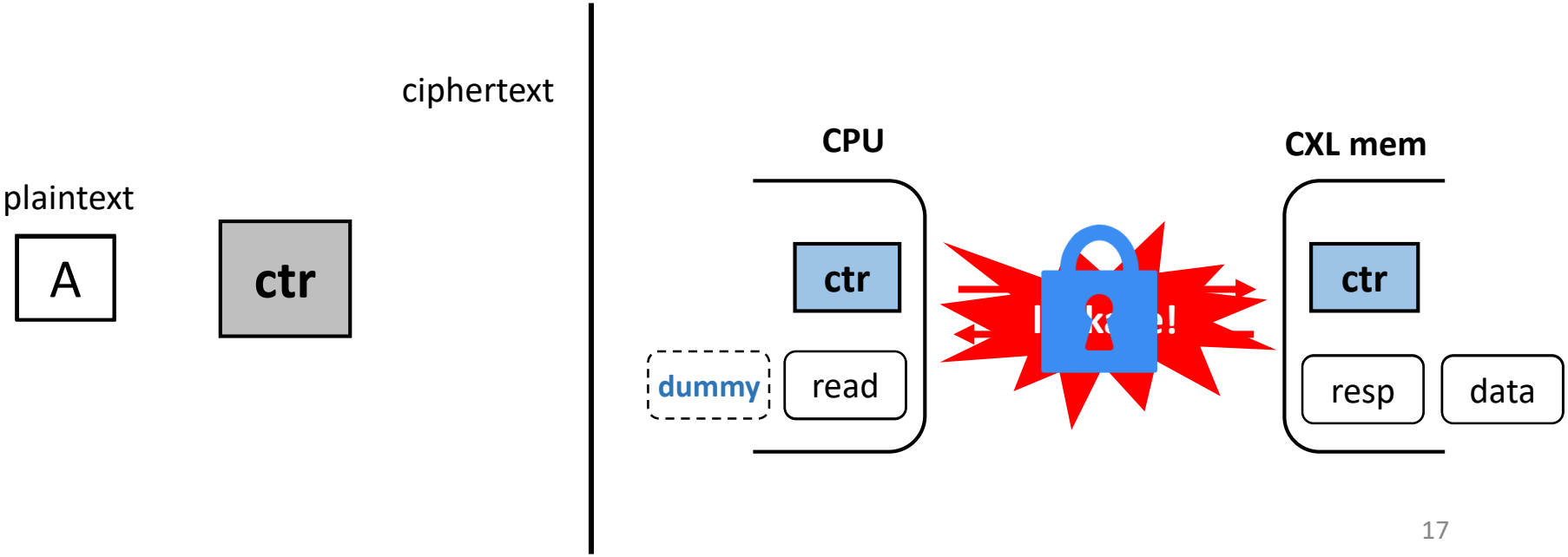
Channel Confidentiality

- Flit-based communication (1 header + 4 slot)
  - Header defines the slots within the flit
  - Each slot contains data or request or response
- **Flit encryption**
  - Encrypt both header and slot
  - Synchronized counter pair in both CXL endpoint



Channel Obfuscation

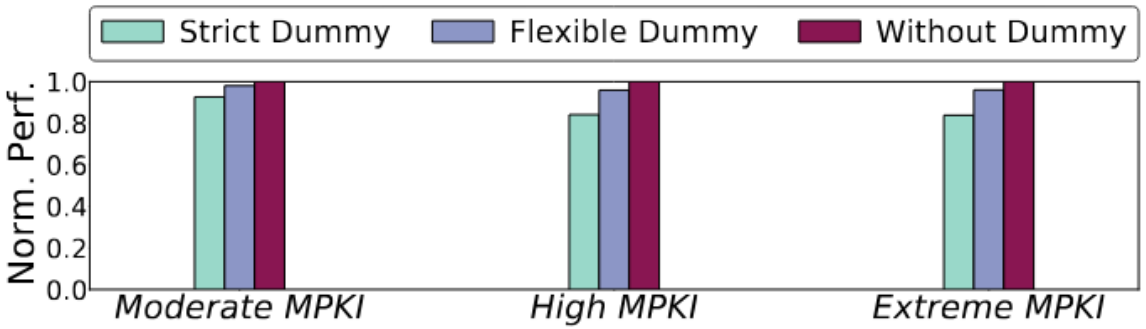
- Obfuscation: hide when and which blocks were read/written
- **Flit counter encryption hides temporal, spatial pattern**
- **Dummy slot hides the type of memory requests**
  - Ensure the same number of flit transmitted in each direction



Flexible Dummy

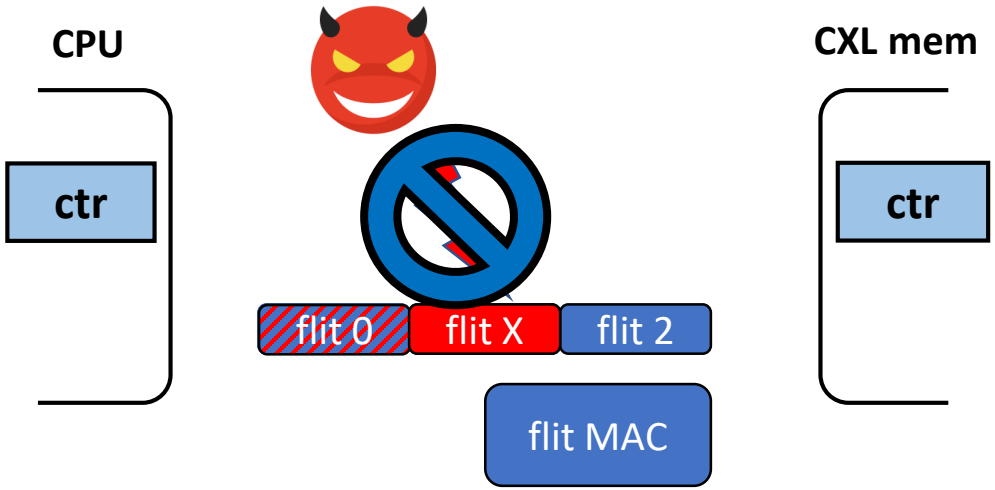
- Prior works assume variable-sized packets
  - **Dummy is required for each packet** to prevent leakage
- CXL flit is of fixed size (68B or 256B)
- It is only necessary to ensure the number of transmitted flits

  -> Dummy is generated **only when the CXL channel is idle**
- Flexible Dummy shows 97.6% performance of No Dummy



Channel Integrity

- Flit injection, drop, and modification
- Replay attack: rollback to old flit
- **Flit counter encryption and MAC**
  - Ensuring integrity and freshness



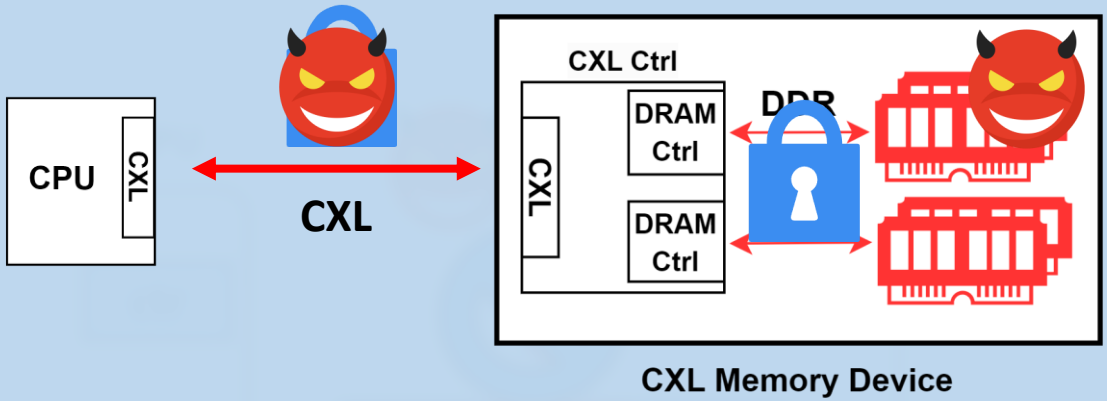
Channel Integrity

Challenges for security

C1) How to protect DDR interface on CXL memory board?

C2) How to protect CXL channel from physical attack?

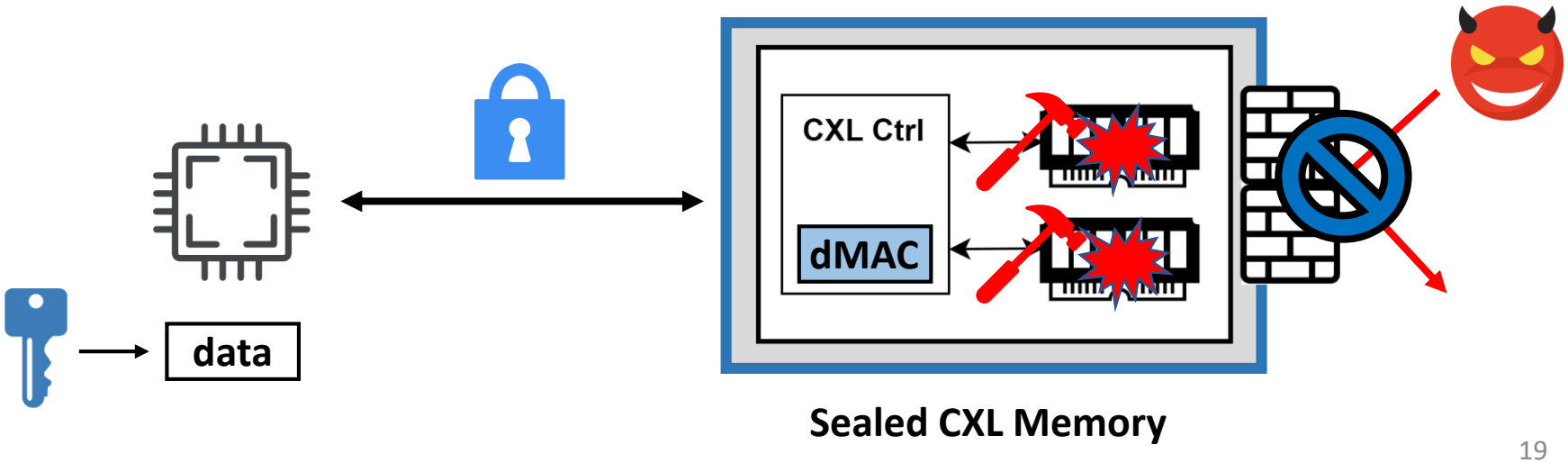
C3) How to safeguard data stored in memory?



Contents

Data Protection in Memory

- Attacks without physical access is feasible
  - Confidentiality: row-hammer*, cold-boot attack
  - Integrity: row-hammer
- **Data encryption at the CPU**
- **Data MAC generation/verification at the CXL controller**



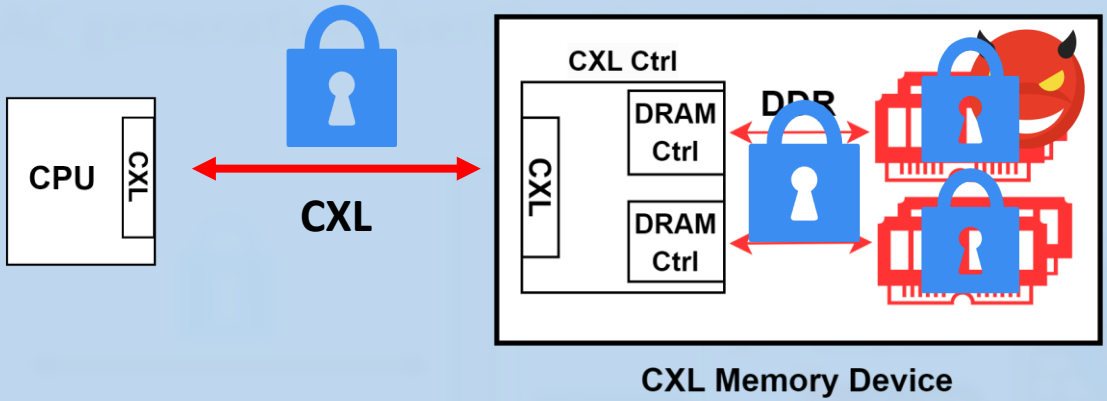
Data Protection in Memory

Challenges for security

C1) How to protect DDR interface on CXL memory board?

C2) How to protect CXL channel from physical attack?

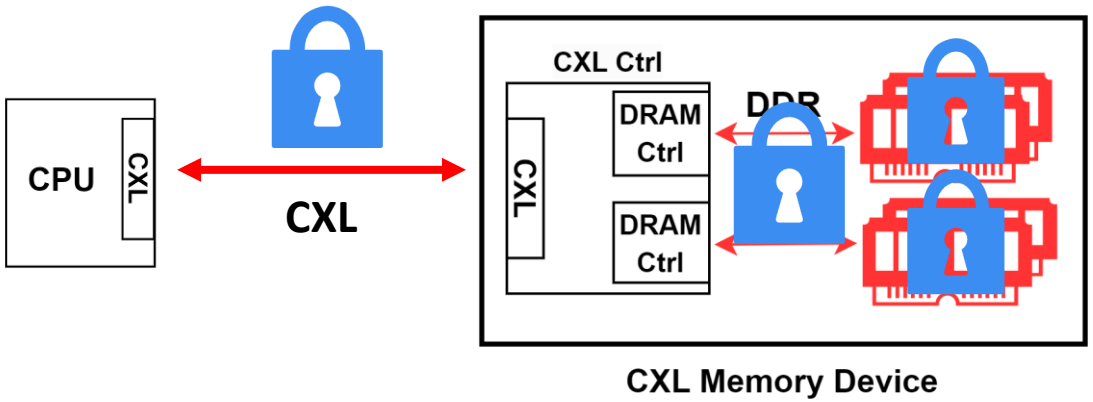
C3) How to safeguard data stored in memory?



Total View on System Security

- CXL DDR interface protection
  - Physical attack unavailable
- CXL Channel physical protection for obfuscation
- Data protection stored in memory

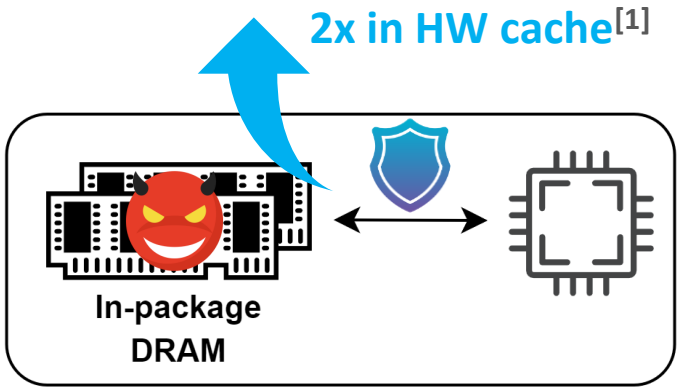
-> **Total confidentiality, integrity, freshness, obfuscation!**

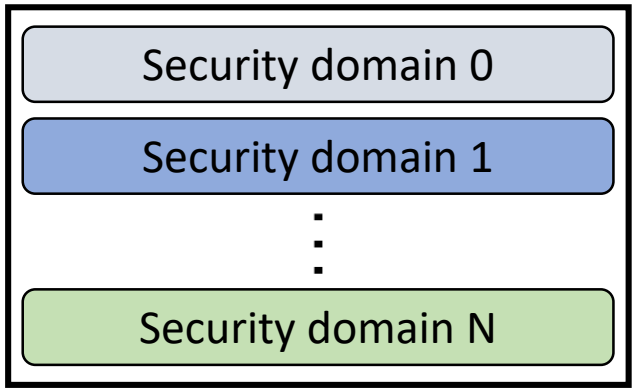


Contents

In-package DRAM Cache

- Integration of CPU and memory in a package
  - DDR interconnect not exposed to attackers
  - Still vulnerable to row-hammer attack
- Extended main memory vs. **hardware cache**
- Side-channel attack via shared cache
  - Strict partition among security domains





In-package DRAM cache

[1] Baptiste Lepers, et al. “Johnny Cache: the End of DRAM Cache Conflicts (in Tiered Main Memory Systems)”, OSDI’23

Contents

- Background
- Motivation
- Design
- Evaluation
- Conclusion

Methodology

- Simulator: ZSim + DRAMSim3

CPU	Core	3.2 GHz, 8 cores, out-of-order x86_64
	L1 Cache	32KB, private, 8-way
	L2 Cache	256KB, private, 8-way
	L3 Cache	8MB, shared, 16-way
CXL	Memory	DDR4-2400, 19.2GB/s, 2 channels
	Bandwidth	PCIe 5.0, 32GB/s per direciton
	Latency	Port delay: 80ns
Memory	DDR	DDR4-2400, 19.2GB/s, 2 channels
	DDR Cache	DDR4-2400, 19.2GB/s, 2 channels
	HBM Cache	HBM, 32GB/s, 8 channels
AES Engine	870MHz, 111.3Gbps, 11 cycles per encryption	
CXL Switch	Switch and port delay: 180ns (used in VI-E)	

Workload (abbr.)	LLC MPKI
namd (namd)	0.45
ep (ep)	1.55
ft (ft)	2.20
gcc (gcc)	3.04
cactusBSSN (bssn)	6.01
cactusADM (adm)	6.54
zeusmp (zeus)	8.99
mg (mg)	15.99
bt (bt)	19.37
mummer (mum)	20.25
xalancbmk (xal)	20.37
graph500-csr (csr)	28.30
sp (sp)	31.60
graph500-list (list)	32.06
libquantum (quant)	42.47
mcf (mcf)	75.88
tiger (tiger)	300.67

Evaluation Schemes

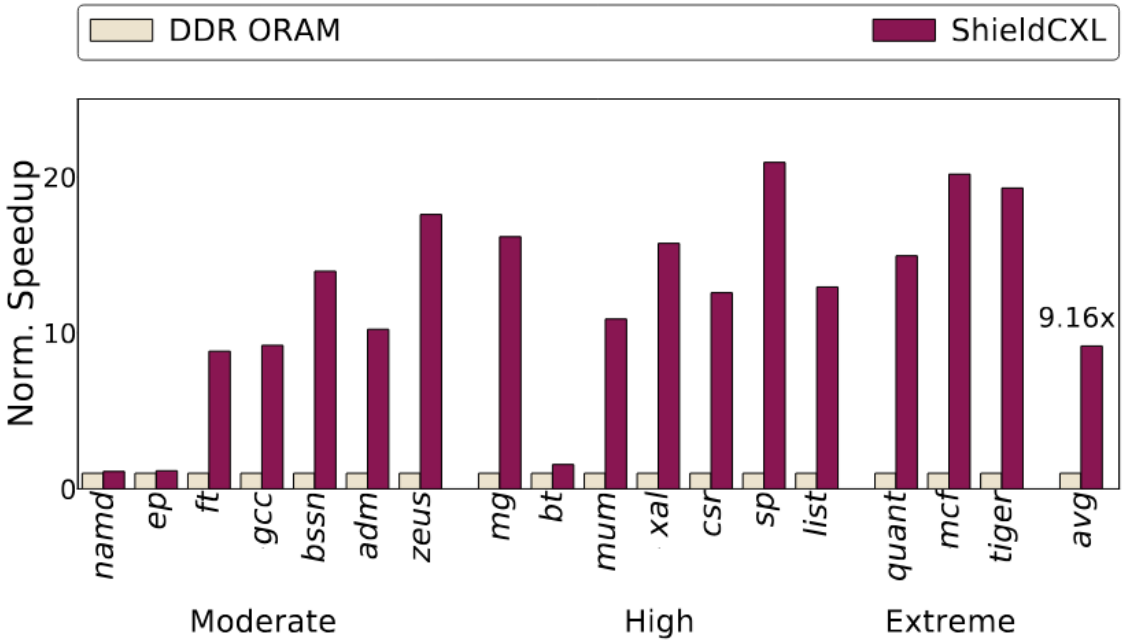
- **Our scheme: ShieldCXL with additional DRAM cache**
- ORAM: PathORAM[1]
- Integrity tree: VAULT[2], variable arity unified tree

Scheme	Access Obfuscation	Confidentiality	Integrity	Freshness	Channel Protection
Unsecure DDR	✗	✗	✗	✗	N.A
Secure DDR	✗	✓	✓	✓	N.A
Unsecure CXL	✗	✗	✗	✗	✗
CXL VAULT	✗	✓	✓	✓	✓
DDR ORAM	✓	✓	✓	✓	N.A
ShieldCXL	✓	✓	✓	✓	✓

[1] Emil Stefanov, et al. “Path ORAM: An Extremely Simple Oblivious RAM Protocol”, CCS’13
[2] Meysam Taassori, et al. “VAULT: Reducing Paging Overheads in SGX with Efficient Integrity Verification Structures”, ASPLOS’18

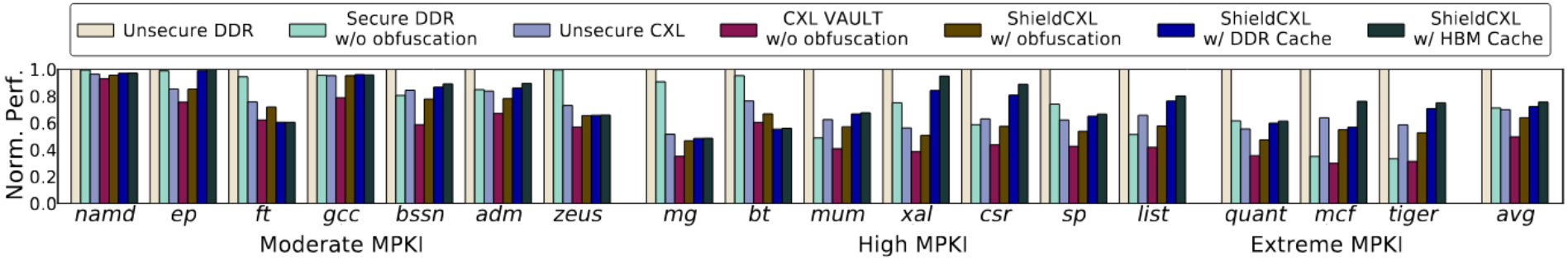
ShieldCXL vs. Path ORAM

- DDR ORAM: Access obfuscation in DDR memory
  - Significant performance degradation with Path ORAM
- ShieldCXL outperforms DDR ORAM by 9.16x



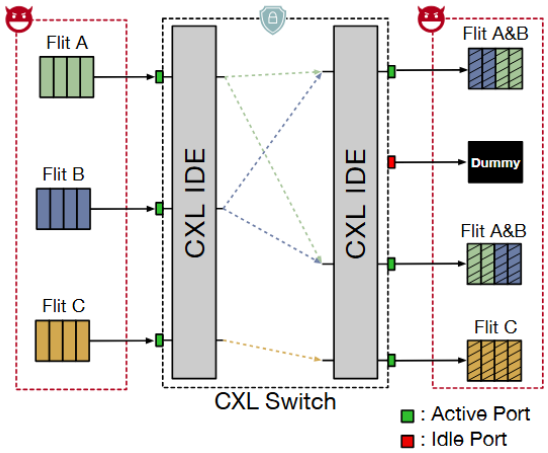
Performance Comparison

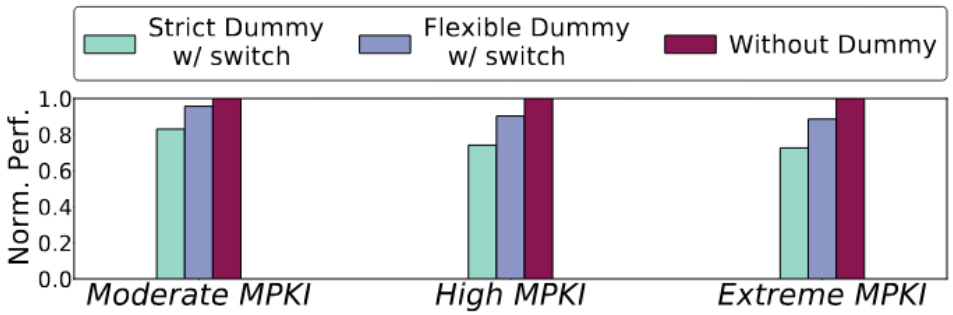
- Performance overhead by protection mechanism
  - secDDR: 21.9% slowdown vs. unsecDDR
  - ShieldCXL: 4.7% slowdown vs. unsecCXL
- Performance improvement by DRAM cache
  - ShieldCXL+HBM shows 17.7% speedup



Obfuscation on CXL Switch

- CXL flit routing by CXL switch
  - Multiple computing & memory nodes on a switch
- Dummy flits to all channels are required for obfuscation
- Flexible Dummy can mitigate performance overheads





Contents

Conclusion

- Existing DDR-based protection scheme incurs too much overhead.
- CXL interface enables an opportunity for efficient protection mechanisms.
- Tamper-responding sealing eliminates complicated DDR-based protection mechanisms.
- Access obfuscation is achieved with low overhead, compared to ORAM.

Q & A