Slice+Slice Baby:
Generating Last-Level Cache Eviction Sets in the Blink of an Eye

Bradley Morgan[1,2], Gal Horowitz[3], Sioli O'Connell[1], Stephan van Schaik[4], Chitchanok Chuengsatiansup[5], Daniel Genkin[6]
Olaf Maennel[1], Paul Montague[2], Eyal Ronen[3] and Yuval Yarom[7]

[1]The University of Adelaide [2]Defence Science and Technology Group [3]Tel-Aviv University [4]University of Michigan [5]The University of Klagenfurt [6]Georgia Tech [7]Ruhr University Bochum

Why Should I Care?

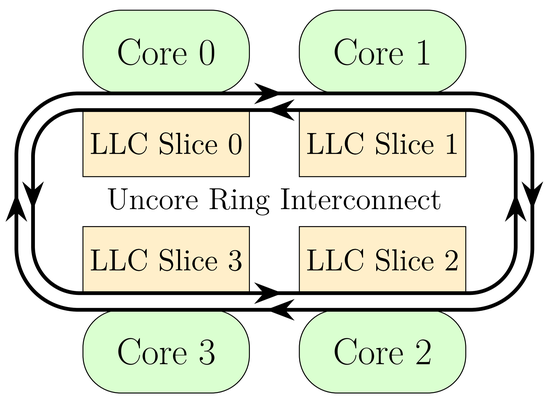
We create LLC eviction sets in 0.1 to 1.6 seconds.

- **We propose a weird gate to *compare* racing LLC memory accesses, identifying their LLC slice.**
- **We infer slice indices within memory pages to achieve speed-ups.**

Defences against cross-core LLC attacks must be feasible within shorter timeframes.

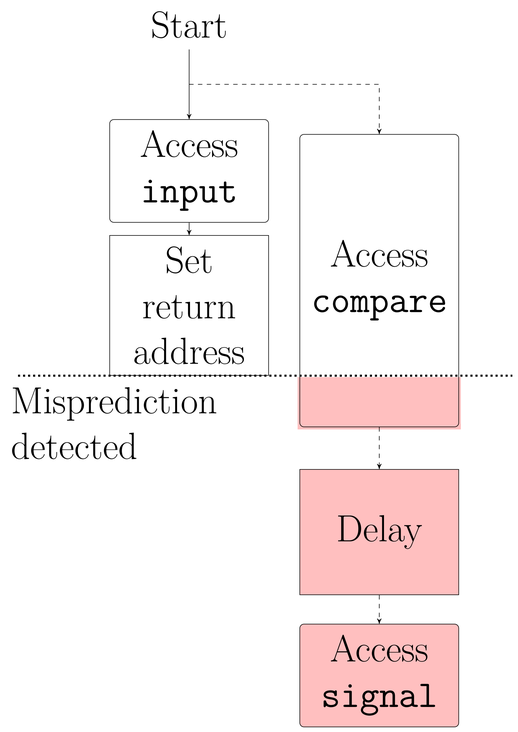
LLC Attacks are Hard

Intel uses a proprietary function to distribute memory across LLC 'slices'. An attacker is unaware where their memory resides.

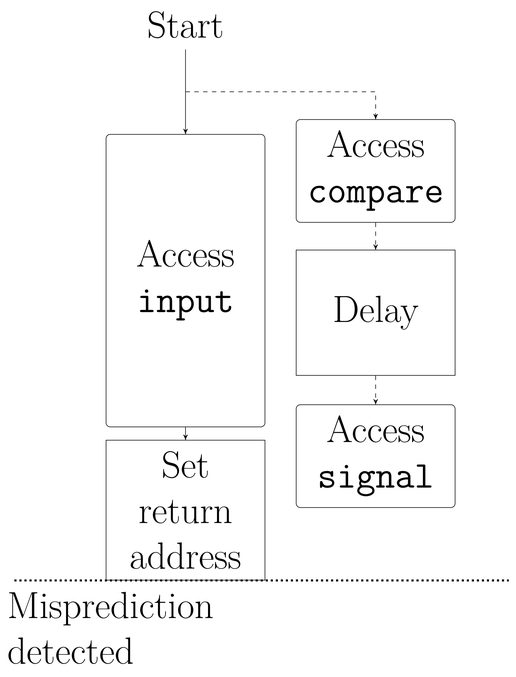


Classifying Memory by Slice

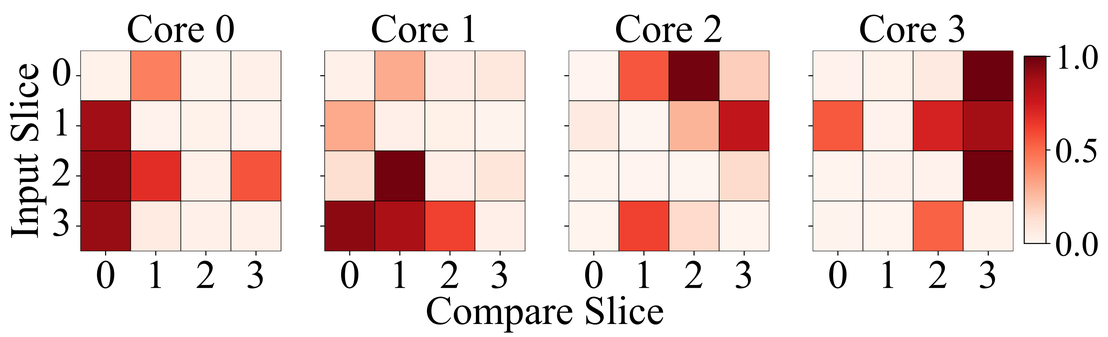
Our noise-resilient memory access *comparator* weird gate [1, 2]:



- `input` latency quicker than `compare`.



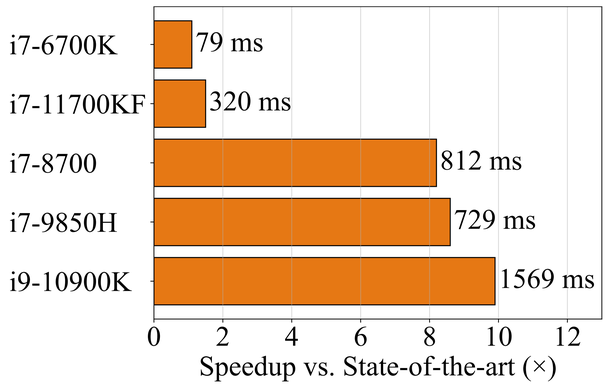
- `input` latency slower than `compare`.



Dark red is a positive `signal`, indicating a faster `compare` slice access.

Slice-Aware Eviction Sets

We can quickly build LLC eviction sets with known slice mappings.



References

[1] D. Katzman, W. Kosasih, C. Chuengsatiansup, E. Ronen, and Y. Yarom, "The gates of time: Improving cache attacks with transient execution," in *USENIX Sec.*, 2023.

[2] D. A. Kaplan, "Optimization and amplification of cache side channel signals." arXiv 2303.00122, 2023.

Links

bradley.morgan@adelaide.edu.au
github.com/0xADE1A1DE/Slice-Slice-Baby