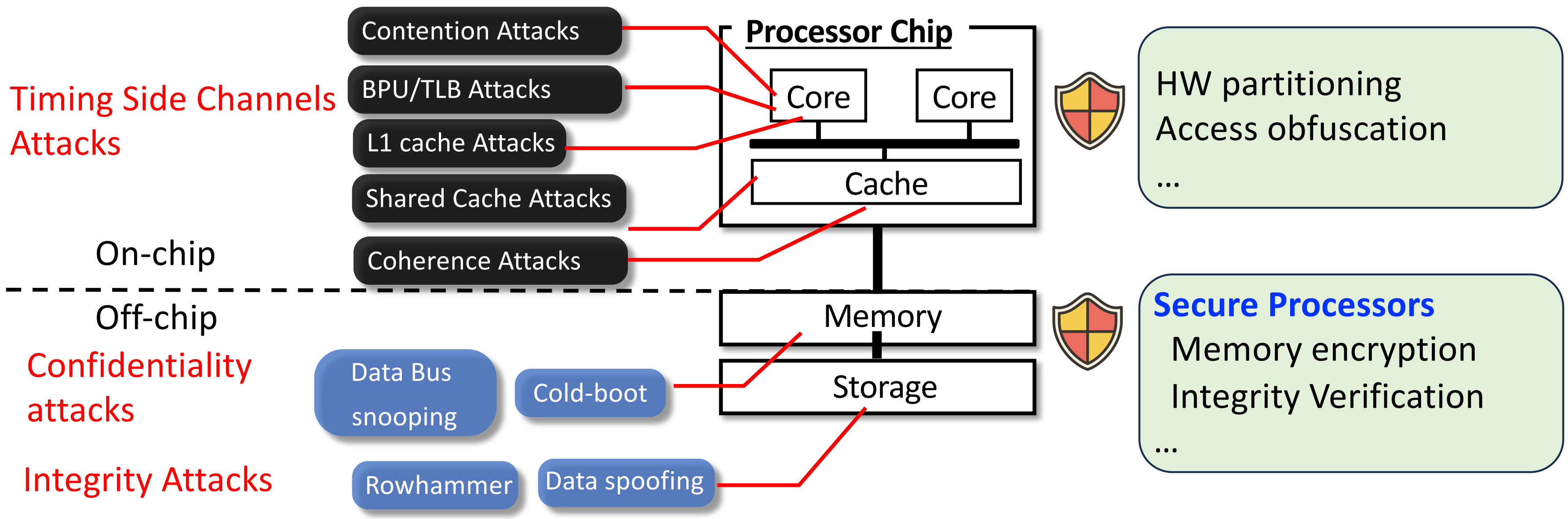
IvLeague: Side Channel-resistant Secure Architectures Using Isolated Domains of Dynamic Integrity Trees

Md Hafizul Islam Chowdhury and Fan Yao

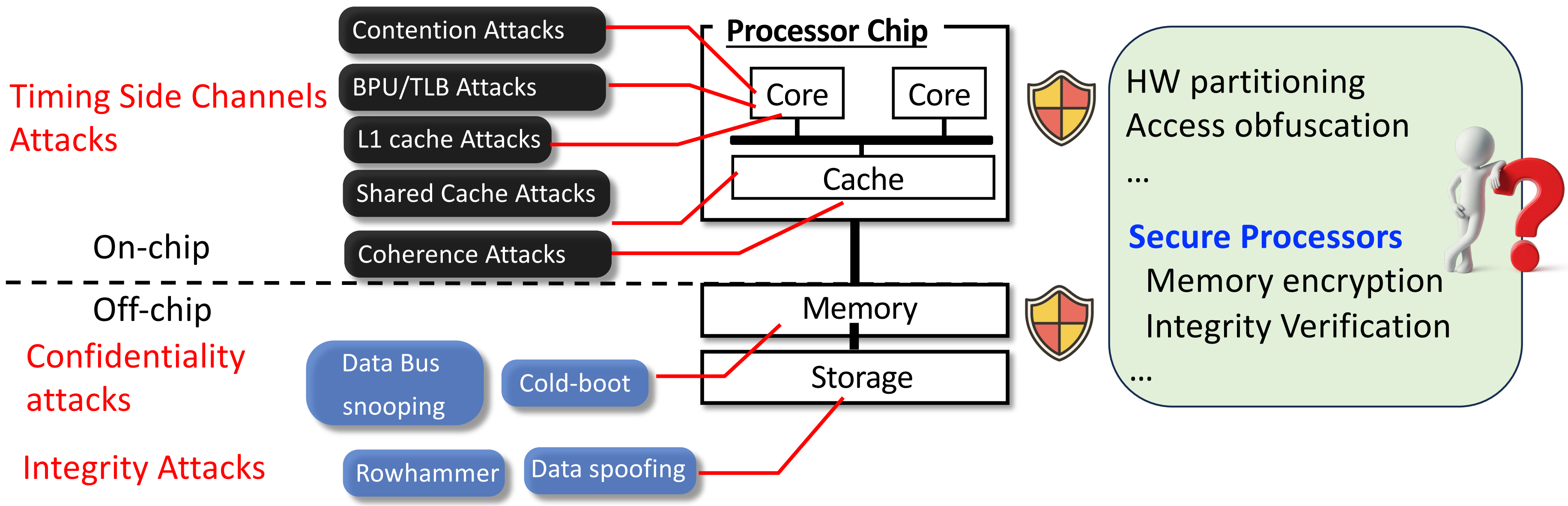
University of Central Florida

New England Hardware Security Day
April 18, 2025

High-level Overview of uArch Security

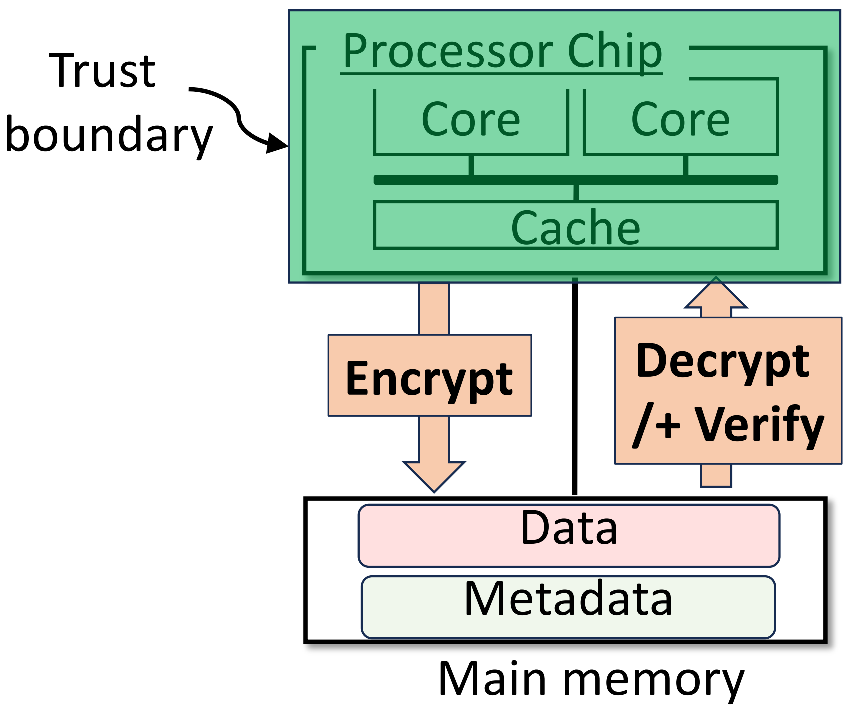


High-level Overview of uArch Security

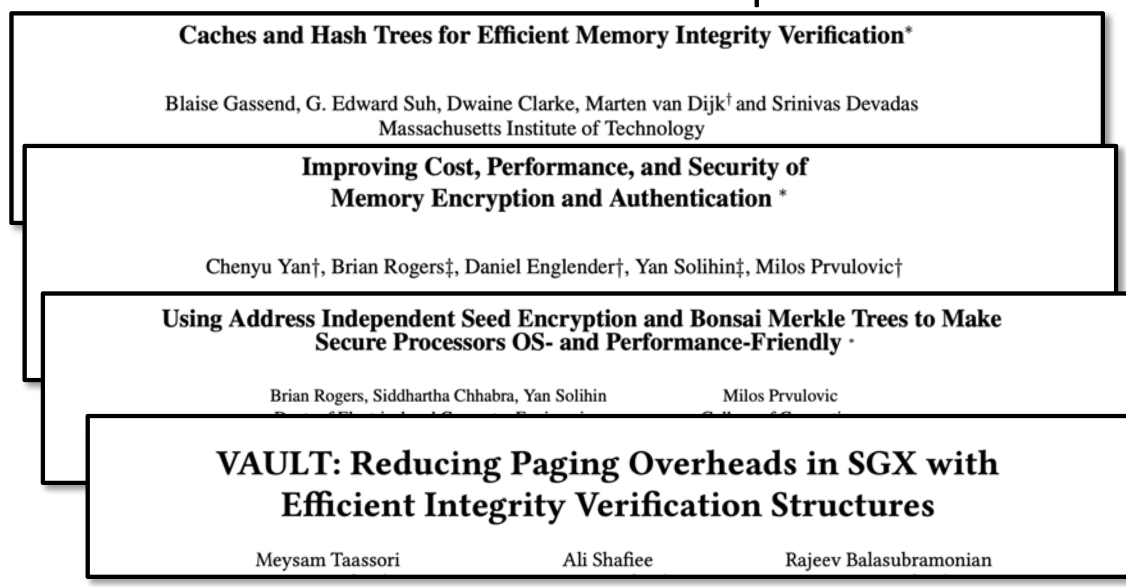


Do these security mechanisms compose well within computing systems?

Secure Processor Architectures in a Nutshell



- Decades of research on secure processors



Caches and Hash Trees for Efficient Memory Integrity Verification*

Blaise Gassend, G. Edward Suh, Dwaine Clarke, Marten van Dijk† and Srinivas Devadas
Massachusetts Institute of Technology

Improving Cost, Performance, and Security of Memory Encryption and Authentication *

Chenyu Yan†, Brian Rogers‡, Daniel Englender†, Yan Solihin‡, Milos Prvulovic†

Using Address Independent Seed Encryption and Bonsai Merkle Trees to Make Secure Processors OS- and Performance-Friendly *

Brian Rogers, Siddhartha Chhabra, Yan Solihin   Milos Prvulovic

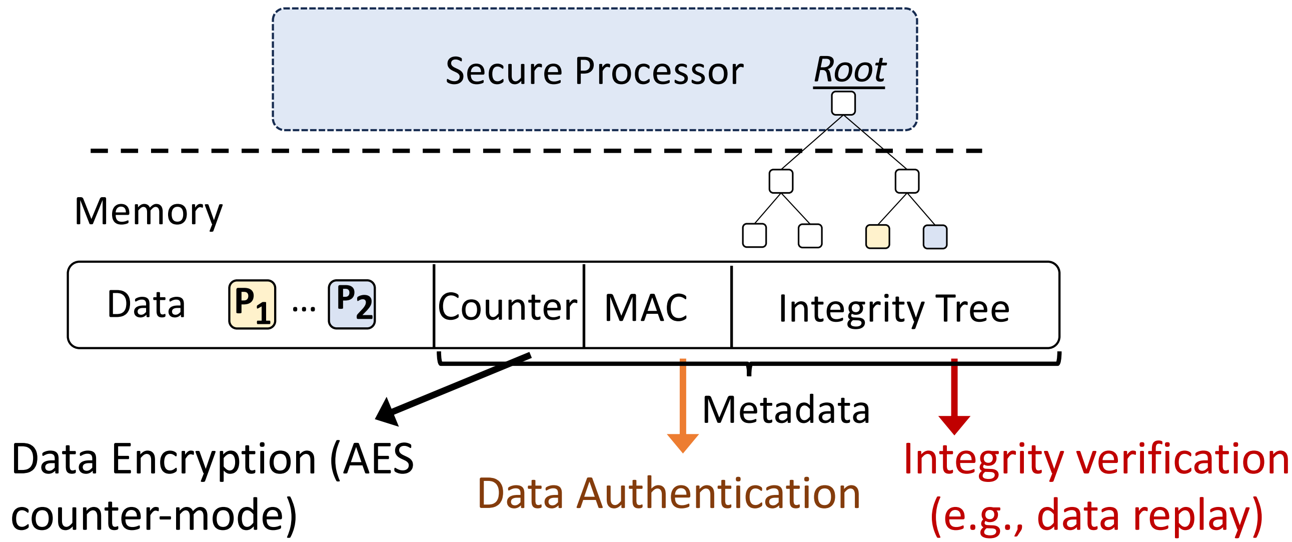
VAULT: Reducing Paging Overheads in SGX with Efficient Integrity Verification Structures

Meysam Taassori   Ali Shafiee   Rajeev Balasubramonian

- Foundations of today’s TEE-enabled processors
- Industry solutions: **Intel SGX**, **Apple Secure Enclave** etc.

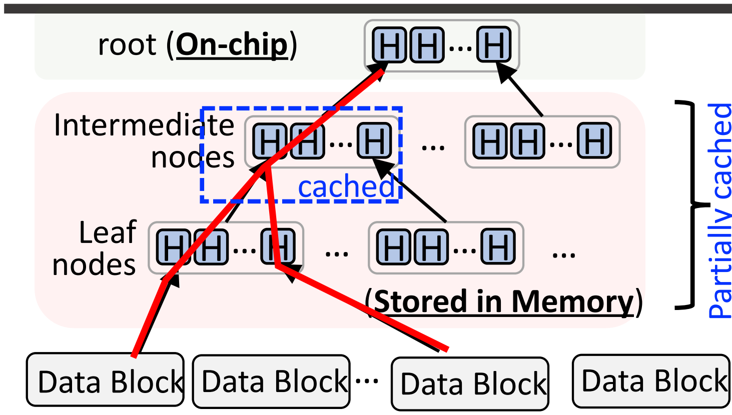
UCF

Secure Processor Designs Worsen uArch Security



Key observation: metadata sharing across security domains

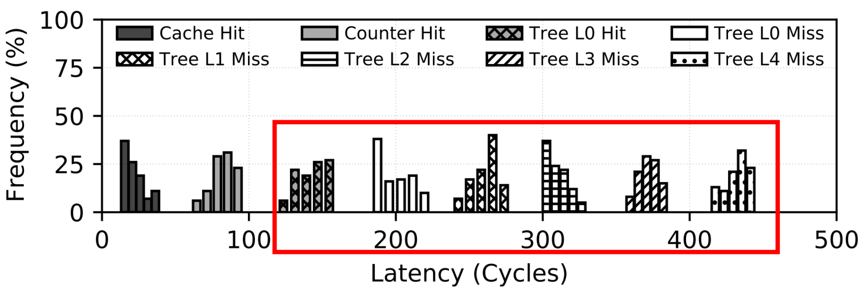
Side Channels Exploiting Security Metadata[1]



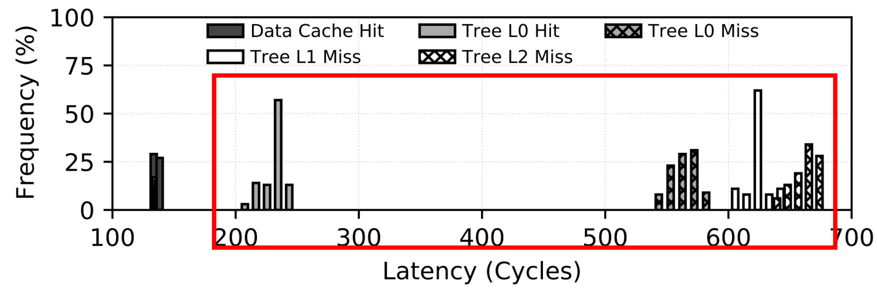
Integrity tree globally shared

Metadata-based timing modulation:
Victim data load -> Integrity Tree Traversal -> Cached shared node -> Attacker's faster integrity verification of data loads

Latency distribution due to integrity tree traversal



BMT Hash Tree



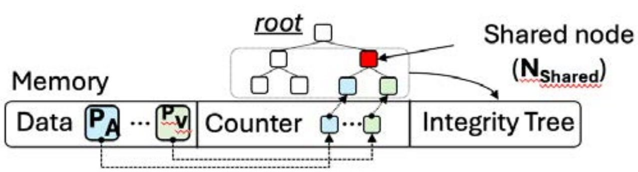
Intel SGX Tree

[1]**MetaLeak**: Uncovering Side Channels in Secure Processor Architectures Exploiting Metadata, *Md Hafizul Islam Chowdhuryy, Hao Zheng and Fan Yao, ISCA'2024*

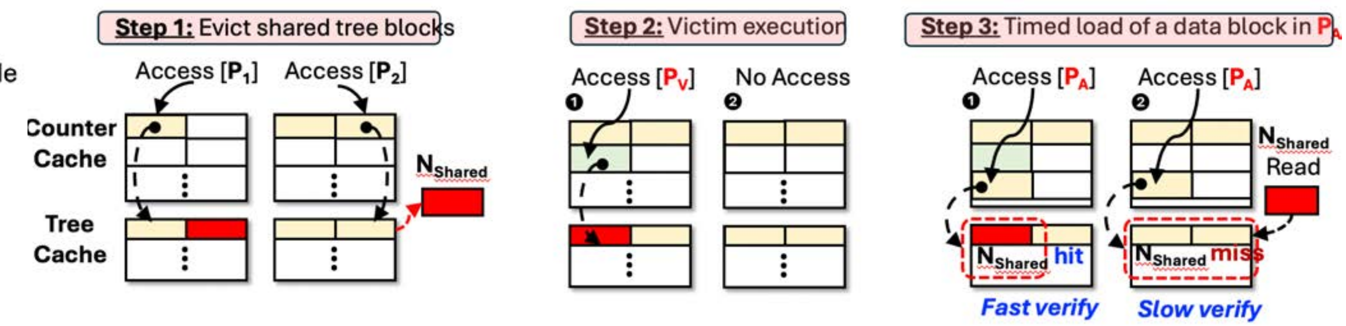
UCF

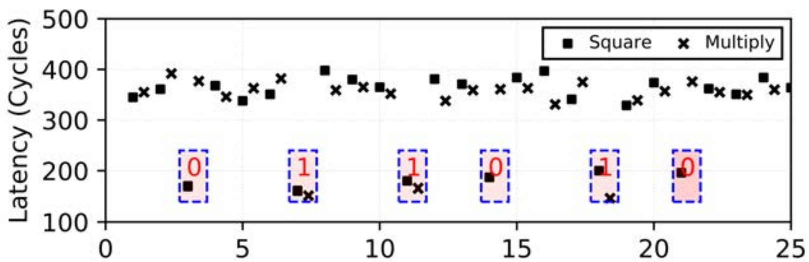
Side Channels Exploiting Security Metadata[1]

Exploitation mechanisms similar to Evict+Reload -> **mEvict+mReload**

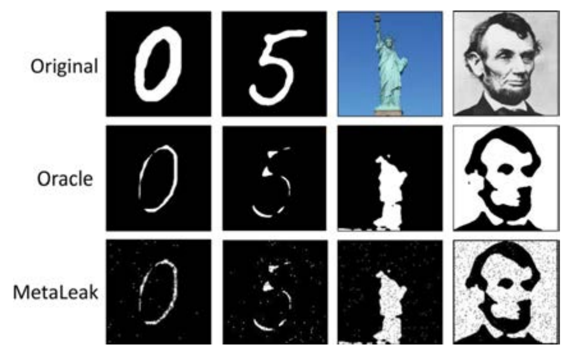


Sharing of integrity tree metadata





RSA private exponent recovery (Intel SGX i7-9700K)



libjpeg image restoration

Metadata Mechanisms Extends uArch Attack Surface

Existing uArch defenses cannot mitigate the metadata-based attacks

Assumptions made by typical microarchitectural defense

For shared-data exploits (typically read-only)

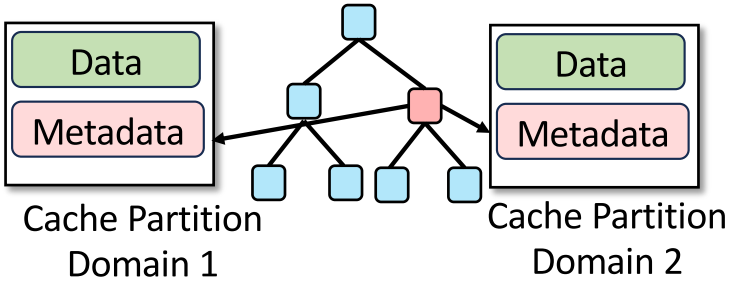
--> Disabling **data/memory sharing** on untrusted domains

For contention-based exploit (no memory sharing)

--> Isolation of shared HW resource for **data access (e.g., cache partitioning)**

We previously ***do not assume*** readable and writable shared data across domains

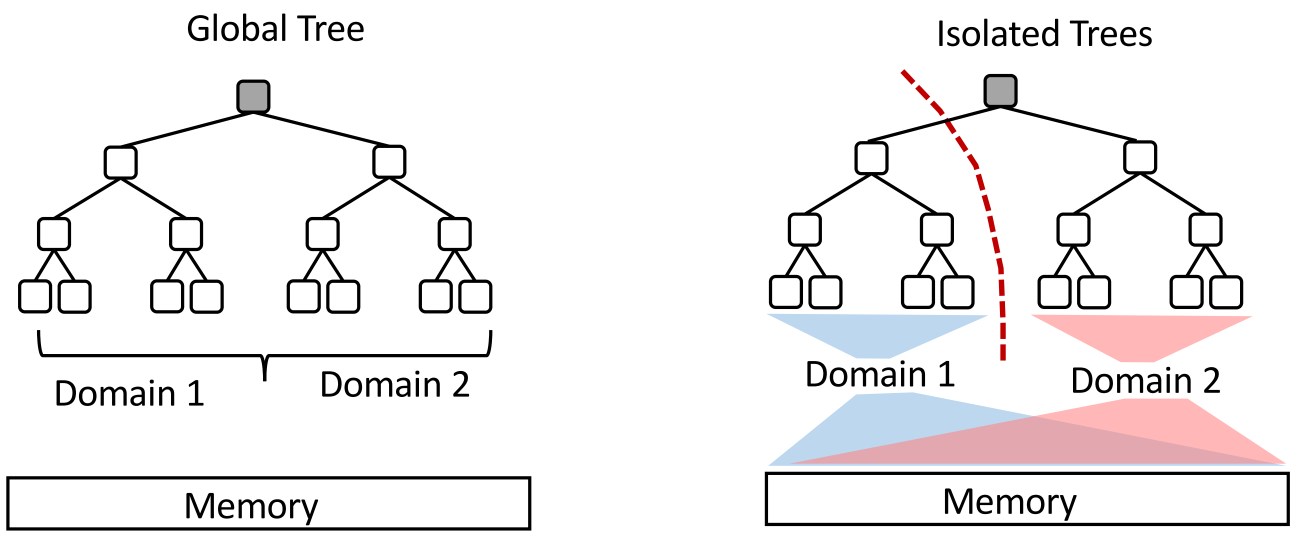
Metadata coherence issues



Need to rethink the secure processor designs for uArch security!

IVLeague: Side-channel Resistant Integrity Metadata Mechanism

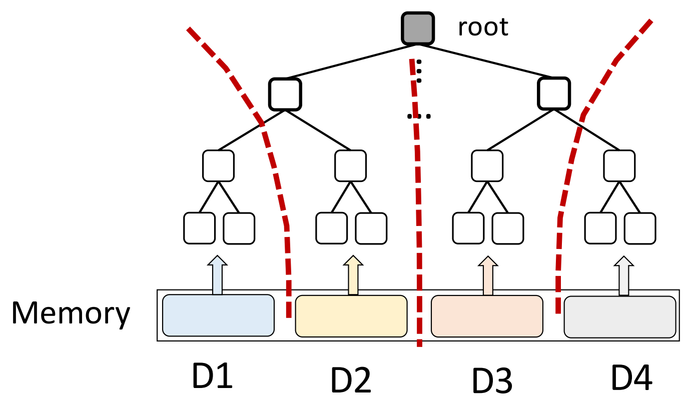
- Extend microarchitectural defense principles for security metadata mechanisms
- Main idea: metadata-level isolation for integrity verification (IV)
  - Ensure no tree node sharing in memory between domains



2. IvLeague: Side Channel-resistant Secure Architectures Using Isolated Domains of Dynamic Integrity Trees Md Hafizul Islam Chowdhuryy and Fan Yao, MICRO 2024

UCF

Statically Partitioning the Integrity Tree?



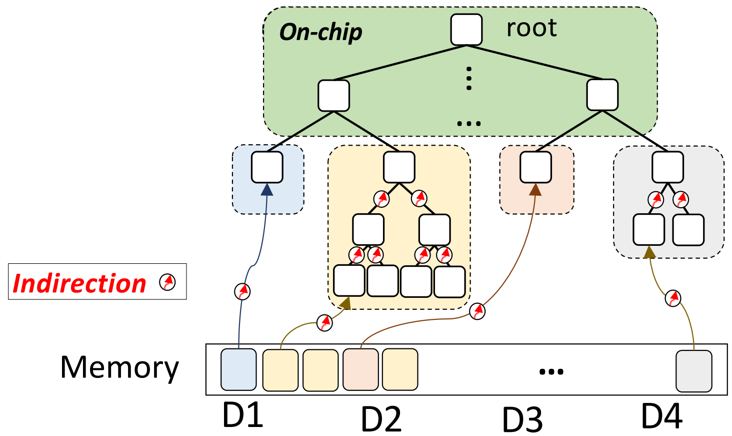
- Low domain management overhead (similar to global tree)
- Fixed number of supported domains, fixed coverage per domain

➢ 1. Does not scale well according to runtime domains (e.g., enclaves)

➢ 2. Could not support applications with larger dynamic memory footprint

➢ 3. Rely on the OS (untrusted) to map pages from fixed region to domains

Fully Dynamic Isolated Integrity Trees?



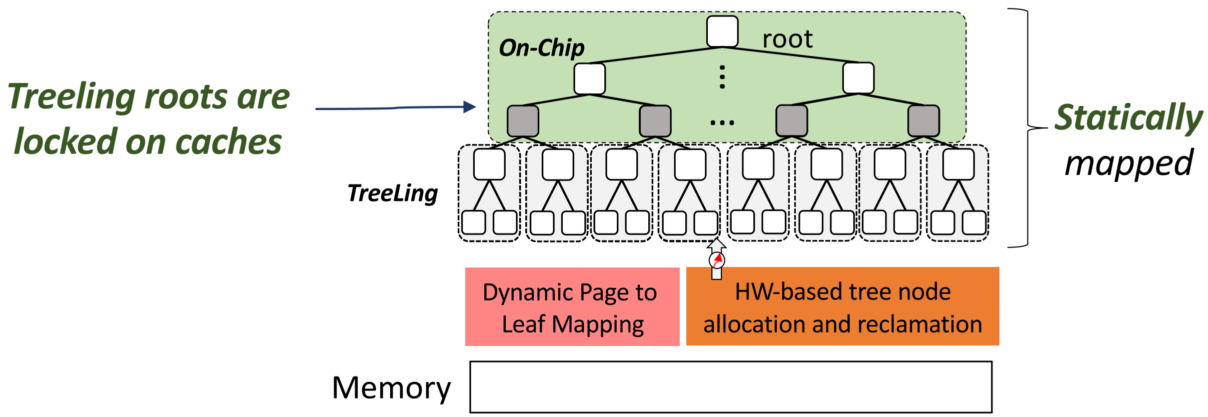
- Build and grow **per-domain trees** at runtime-> flexible memory coverage
- Support dynamic runtime domain scaling

➢ Prohibitive metadata overhead for IV tree traversal (i.e., indirection metadata)

➢ Substantial tree traversal overhead -> long IV latency for reads

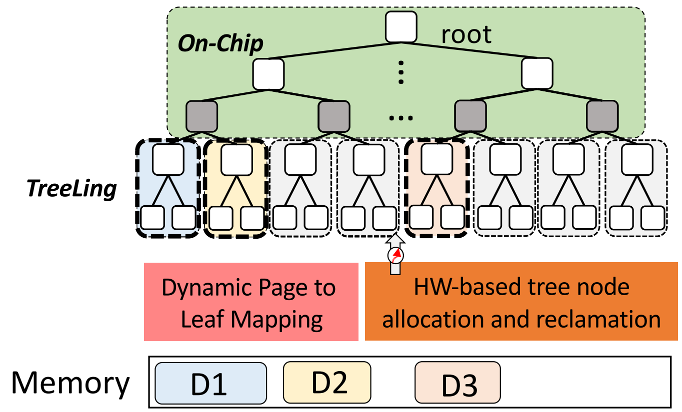
IvLeague: Dynamic Domains of Isolated Tiny Static Trees

Split the integrity tree into many ***small but fixed-sized*** sub trees (**TreeLing**).
Each TreeLing is **statically-mapped**, **isolated** and **allocated to domains on-demand**.

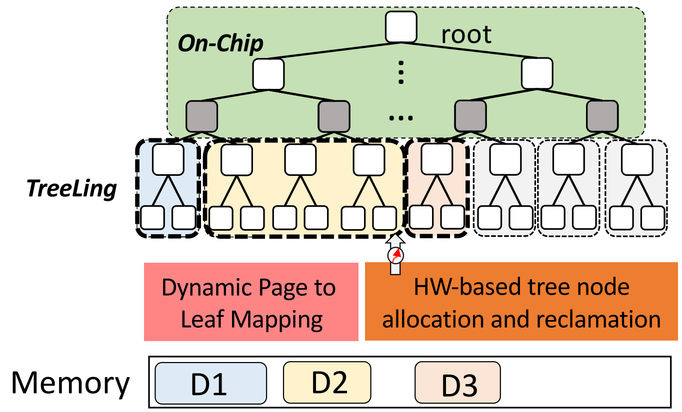


- Each sub-tree (TreeLing) is ***statically mapped***, **no indirection needed for leaf-to-root traversal**
  - Each TreeLing covers ***a small chunk of memory*** (e.g., 8MB to 64MB)
- TreeLings are assigned to domains on-demand, ***resize integrity coverage during runtime***
- Support a large number of secure domains (up to 4K)

IvLeague: Dynamic Domains of Isolated Tiny Static Trees



IvLeague: Dynamic Domains of Isolated Tiny Static Trees



IvLeague: Performance Optimization Opportunities

IvLeague's **dynamic intra- and inter-TreeLing** management enables performance optimization opportunities over the default secure processor designs

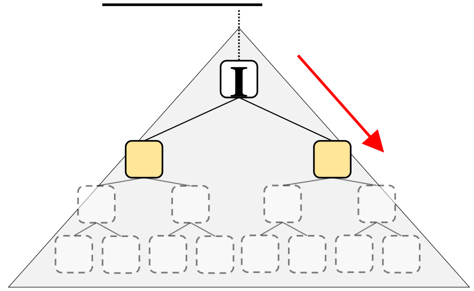
Intermediate tree node

<u>Utilized</u> tree node

<u>Unutilized</u> tree node

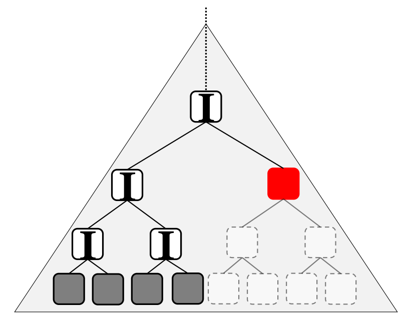
Regular page

Hot page



Intermediate node as leaf
Gradual **intra-tree expansion**
(IvLeague-Invert)

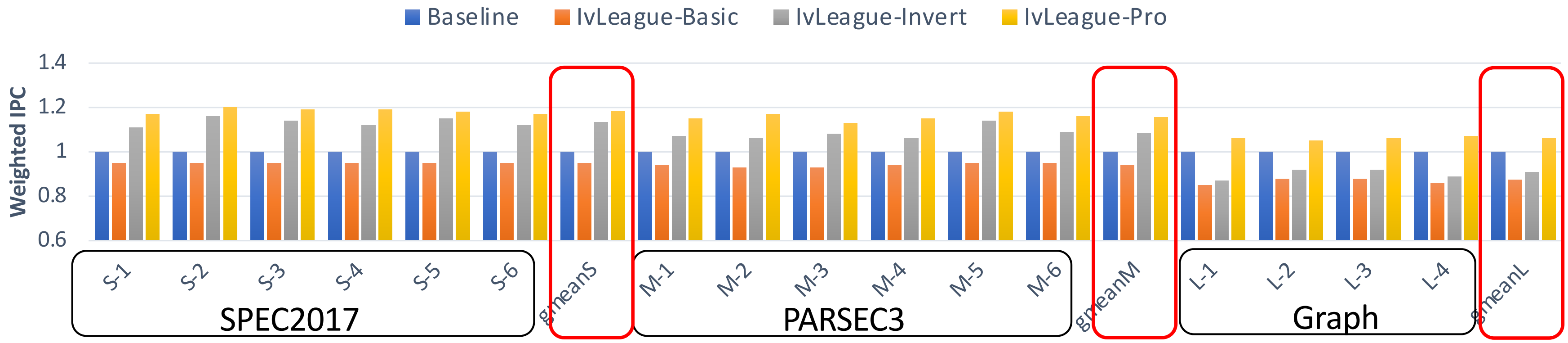
Reduced IV path length



Tracking and swapping of mapping for hot pages
(IvLeague-Pro)

Fast hotpage verification

Performance Comparison



Comparison of performance (i.e., Weighted IPC normalized to Baseline) under different schemes.

	↓2.7%	↓5.5%	↓17.4%
Performance of IvLeague-Basic: Compared to *baseline*	Small	Medium	Large

	↑8.2%/↑13.5%	↑3.4%/↑9.3%	↓13.2%/↑3.4%
Performance of IvLeague-Invert/IvLeague-Pro: Compared to *baseline*	Small	Medium	Large

Side channel-resistant integrity mechanisms can have better performance than the **baseline insecure scheme** with global integrity tree!

UCF

Takeaways and Conclusions

- The need to understand composability of security mechanisms
  - Could a defense for one threat bring a bigger issue for another?
- uArch security *cannot* be considered as a **standalone problem**!
  - Look at uarch security from a broader perspective
- Today's hardware are metadata-rich (despite the deprecation of IV trees)
- Lots of things to explore for cross-threat model uArch security research!

Thank You! Questions?

Fan Yao, Email: fan.yao@ucf.edu
UCF CASR Lab (https://casr.ece.ucf.edu)