Mitigating Cache Side-Channel Leakage in Speculative Execution Pipelines via Hardware-Enforced Memory Access Partitioning

Robin Phillips
Professor
Technische Universität München
Arcisstraße 21, 80333 Munich, Bavaria, Germany

Chris Hall
PhD
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

Avery Rodriguez
Associate Professor
University of Waterloo
200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada

Sam Lopez
D.Sc
Delft University of Technology (TU Delft)
Mekelweg 5, 2628 CD Delft, South Holland, Netherlands

Abstract. Speculative execution vulnerabilities, exemplified by Spectre-class and Meltdown-class exploits, continue to undermine the security guarantees of modern out-of-order processors despite numerous microcode patches and compiler-level mitigations. This paper formalizes the threat model for transient execution attacks targeting shared last-level cache (LLC) structures and proposes a hardware-enforced memory access partitioning (HMAP) mechanism that isolates speculative memory traces at the microarchitectural level. HMAP integrates a tagged partition table within the cache hierarchy, enforced by a lightweight finite-state machine embedded in the memory order buffer. Evaluated across SPEC CPU 2017 benchmarks and a suite of synthetic side-channel workloads, HMAP achieves a 94.7% reduction in exploitable cache timing covert channels while incurring a mean runtime overhead of 2.3%, demonstrating practical deployability for security-critical computing environments.

Keywords: speculative execution security, cache side-channel attacks, transient execution vulnerabilities, microarchitectural isolation, last-level cache partitioning, hardware-enforced memory protection, Spectre mitigation, covert channel elimination, out-of-order processor security

Introduction

The proliferation of speculative execution optimizations in contemporary high-performance processors has introduced a class of microarchitectural vulnerabilities that fundamentally challenge the classical separation between hardware performance engineering and system security. Since the public disclosure of Spectre and Meltdown in 2018, the research community has documented an expanding taxonomy of transient execution attacks—including SpectreRSB, Ret2Spec, MDS (Microarchitectural Data Sampling), TAA (TSX Asynchronous Abort), and RIDL (Rogue In-Flight Data Load)—each exploiting subtle timing discrepancies introduced by speculative, out-of-order, or micro-op execution pathways. As of 2024–2026, the deployment of heterogeneous many-core architectures in cloud-native environments, edge inference accelerators, and trusted execution environments (TEEs) such as Intel SGX and AMD SEV has dramatically expanded the attack surface, rendering purely software-based mitigations such as retpolines, Speculative Store Bypass Disable (SSBD), and kernel page-table isolation (KPTI) increasingly insufficient. These countermeasures uniformly impose non-trivial performance penalties—often exceeding 15–30% throughput degradation on I/O-intensive workloads—without eliminating the underlying microarchitectural root cause. The core technical challenge lies in the fact that modern processors speculatively load and cache memory operands along mispredicted execution paths, leaving persistent state changes in shared cache structures that are observable by co-located processes or virtual machines through high-resolution timer-based probing techniques such as Flush+Reload, Prime+Probe, and Evict+Time. Existing LLC partitioning schemes, including Intel Cache Allocation Technology (CAT) and ARM MPAM (Memory Partitioning and Monitoring), operate at coarse-grained software-visible granularity and do not intercept speculative memory references at the microarchitectural level where the leakage actually originates. This fundamental mismatch between the granularity of existing defenses and the precision of the attack surface motivates the need for a hardware-native, speculative-path-aware memory access control mechanism. This paper addresses this gap by introducing Hardware-Enforced Memory Access Partitioning (HMAP), a microarchitectural mechanism that augments the cache hierarchy with a tagged partition table (TPT) and a lightweight finite-state machine (FSM) embedded within the memory order buffer (MOB). HMAP intercepts speculative memory references prior to LLC eviction, enforces per-security-domain access policies at cache-line granularity, and squashes non-compliant speculative loads without stalling the commit pipeline. The remainder of this paper is structured as follows: Section 2 reviews the related work on transient execution attack taxonomies and existing hardware and software mitigations; Section 3 formalizes the threat model and security definitions; Section 4 presents the HMAP architecture and its integration with the ROB-MOB interface; Section 5 details the VLSI-level implementation considerations and FSM design; Section 6 reports experimental evaluation using SPEC CPU 2017 and synthetic covert-channel benchmarks; Section 7 discusses limitations, deployment considerations for TEE environments, and directions for future work; and Section 8 concludes the paper.

This is a preliminary version. To read the full version of the article, please purchase a subscription.

References

1. Semeniuk, V. V. (2025). OPTIMIZATION OF LOCAL DEVELOPMENT PROCESS USING DOCKER PHP IMAGE THAT COMES WITH A FULL SET OF TOOLS OUT OF THE BOX–PERFORMANCE AND OPTIMIZATION EXTENSIONS. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БАГАТОІНДЕКСНОЇ ТРАНСПОРТНОЇ ЗАДАЧІ З НЕЧІТКИМИ ІНТЕРВАЛАМИ.