Safe Speculation for CHERI

Franz A. Fuchs, Jonathan Woodruff, Peter Rugg, Alexandre Joannou, and Simon W. Moore
Department of Computer Science and Technology, University of Cambridge
franz.fuchs@cl.cam.ac.uk
Project URL: cheri-cpu.org

UNIVERSITY OF CAMBRIDGE
Computer Science & Technology

The computer security world has been plagued by **transient-execution attacks** for more than five years. These attacks manage to leak secrets by leveraging **microarchitectural state** in modern processors. We propose **architectural contracts for CHERI** to argue about transient-execution attacks as well as mitigating them. We show that our contract can be implemented efficiently in CHERI-Toooba, which is an **out-of-order superscalar RISC-V CHERI implementation**, previously developed at the University of Cambridge.

CHERI

CHERI is a hardware capability architecture that extends conventional ISAs to enable **memory-safe software**. CHERI is currently in the process of being ratified as an extension to RISC-V. At its core, CHERI adds **capabilities**, memory references with bounds and permissions. With CHERI, **fine-grained compartmentalisation** models can be built.

Capability Speculation Contract (CSC)

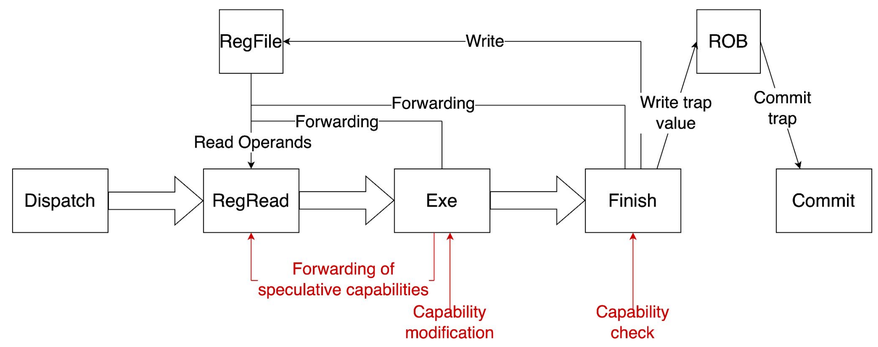
All instruction and data-memory accesses issued in speculation must be authorised by capabilities either:

1. In the committed register file;
2. In memory transitively reachable through 1.

Testing Results

We found two classes of violations against CSC in our CHERI-Toooba implementation:

- Meltdown-CF (Capability Forgery).
- Access to powerful code capabilities.



ALU pipeline causing Meltdown-CF.

Meltdown-CF Sample Violation

```
lc   rDest, 0(cDest)    // delay following insts
cbld cDest, rInv, cInv  // transiently build
                        // a capability
lc   rDest, 0(cDest)    // load the secret
```

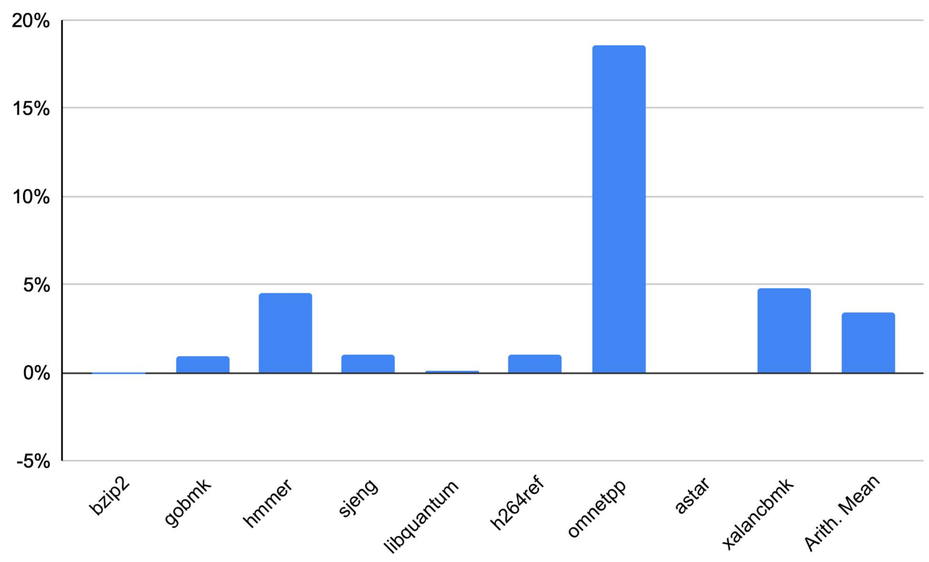
Fixing Meltdown-CF

- Reason: forwarding capabilities to other pipeline stages before finishing exception checks.
- Fix: Clear capability tag on CHERI security violations.
- Effect: CHERI-RISC-V changed most instructions from an exception throwing behaviour to a tag-clearing behaviour.
- Evaluation: No measurable overhead on SPECint2006

Accessing Powerful Code Capabilities

Speculation with code capabilities worked as follows in CHERI-Toooba:

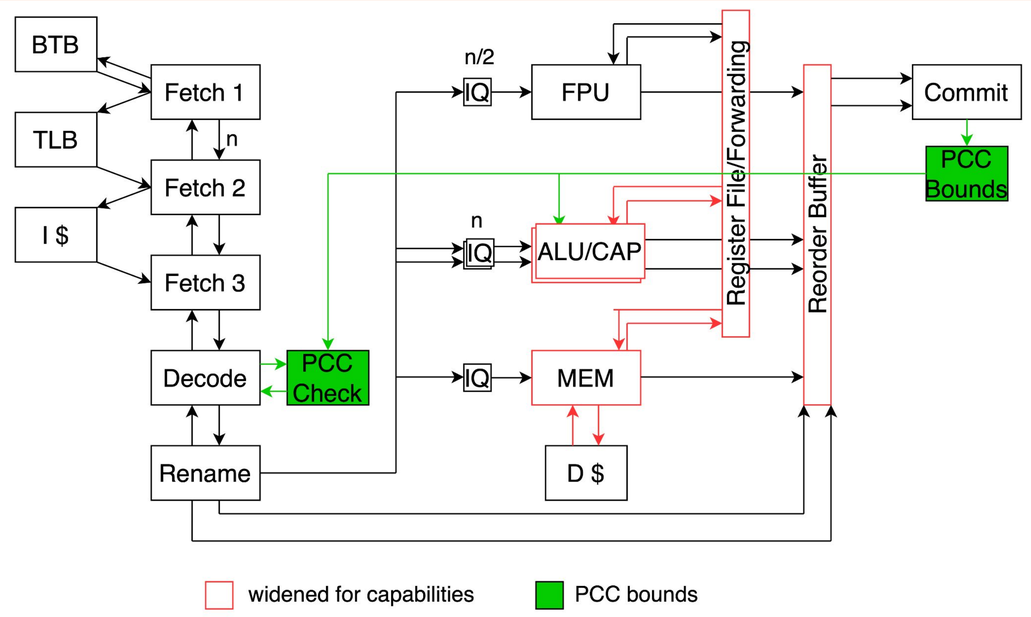
- BTB and RSB stored entire code capabilities leading to speculative code capabilities floating around
- Therefore, control flow can be diverted to code that should have never been accessible



Cycle overhead measurements of SPECint2006 SinglePCC.

SinglePCC

- Mitigation mechanism that only allows a single set of bounds for code capabilities in the processor at a time.
- All control-flow speculation only uses addresses instead of entire capabilities.
- The bounds are updated from the Commit stage on an architectural control-flow edge.



CHERI-Toooba with SinglePCC extension.

Conclusions

In our research, we have proved that the **Capability Speculation Contract (CSC)** is a valuable addition to the CHERI ISA. We developed **automated testing** that is **portable** to other CHERI-RISC-V implementations. Our testing reveals violations of CSC that can lead to **dangerous attacks** against CHERI systems. We developed mitigation mechanisms against all violations and found that our mitigation mechanism called SinglePCC only incurs a **3.43% overhead in SPECint2006** for complete enforcement of CSC.

SRI International

CHERI