I. Why Do We Need Security Evaluation?

The Endless Cat-and-Mouse Game

- Spectre
- InvisiSpec
- Delay-on-Miss...
- Spectre rewind
- Speculative interference attack

II. Challenge: Bridge the Gap

Defenses:
- SafeSpec
- Delay-on-Miss
- invisSpec
- Muontrap
- GhostStores
- GhostMinion

Evaluation Tools:
- JasperGold
- Rosette
- CVC5
- Boolector
- z3
- Coq
- Isabelle/HOL
- ACL2

We Propose: A modeling approach aligned with how architects design defense.

III. Pensieve Microarchitectural Modeling

- Decouple timing and functionality using the hand-shaking interface
- Represent a space of timing behavior with uninterpreted function

What is Uninterpreted Function?

- A UF represents space of functions with same input/output types
  - Example: Bool UF(Bool, Bool)
- UF helps us
  - state “what” affects the output,
  - abstract away the details on “how” the input affects the output
- Use UF to represent a space of timing behavior
  - Memory_latency = UF(historyOf(in_valid, in_addr))

IV. Pensieve Security Evaluation Framework

- Security Property (Speculative Non-interference)

V. Checking Time and Scalability

- Microarchitecture Setup
  - 5 types of instructions
  - 4-entry register file
  - 4-entry data memory
  - 16-entry instruction memory
  - 8-entry ROB
  - GhostMinion defense
- Problem: Checking time increases exponentially as the number of simulated cycles increases
- Future work: Combine Pensieve with more powerful formal verification backend

VI. New Attack Variant on GhostMinion

V-1. Spectre is MITIGATED by InvisiSpec

if (false)
  ld sec // transmitter

V-2. Speculative Interference Attack BREAKS InvisiSpec

(a) pipeline view

V-3. Speculative Interference Attack is MITIGATED by GhostMinion

V-4. New Attack Variant BREAKS GhostMinion

V-5. Summary

No program order between “ld sec” and “ld y”