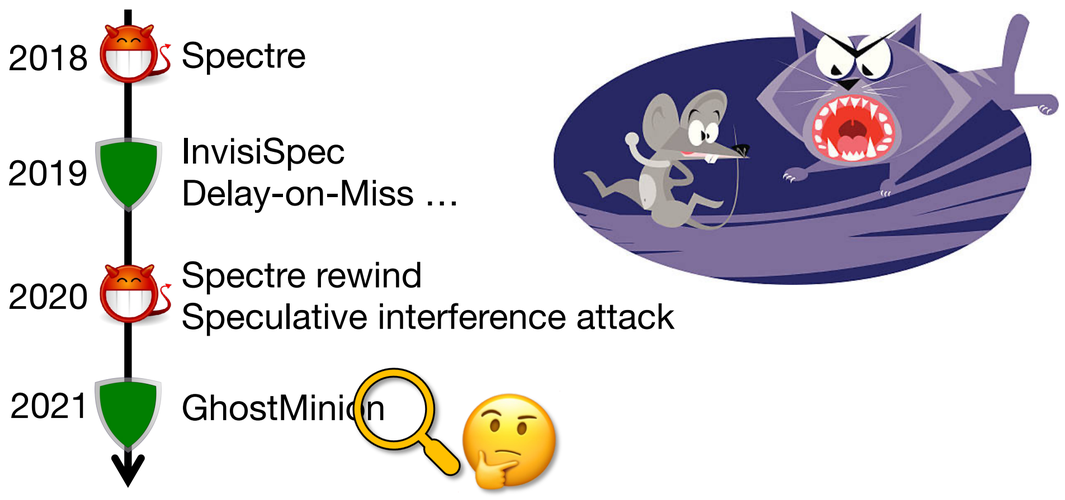
Pensieve: Microarchitectural Modeling for Security Evaluation

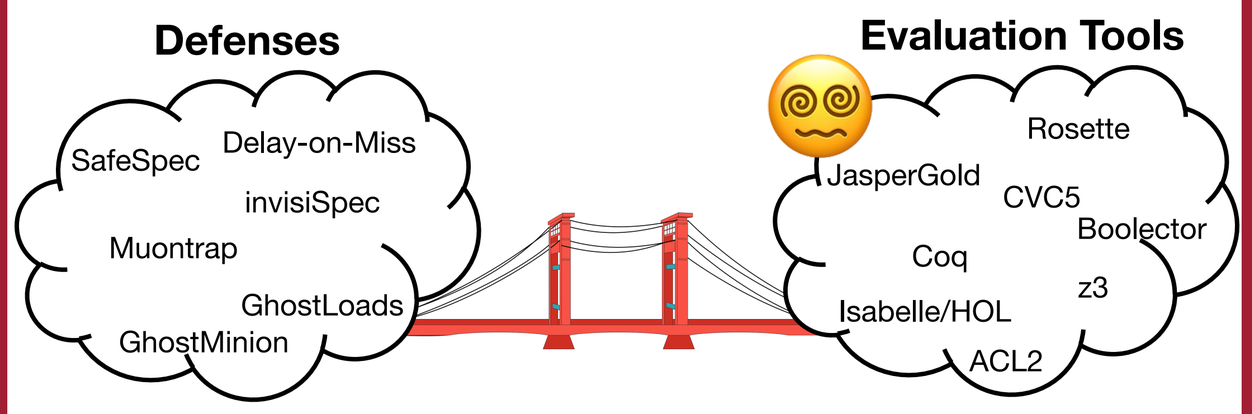
Yuheng Yang, Thomas Bourgeat, Stella Lau, Mengjia Yan

I. Why Do We Need Security Evaluation?

The Endless Cat-and-Mouse Game



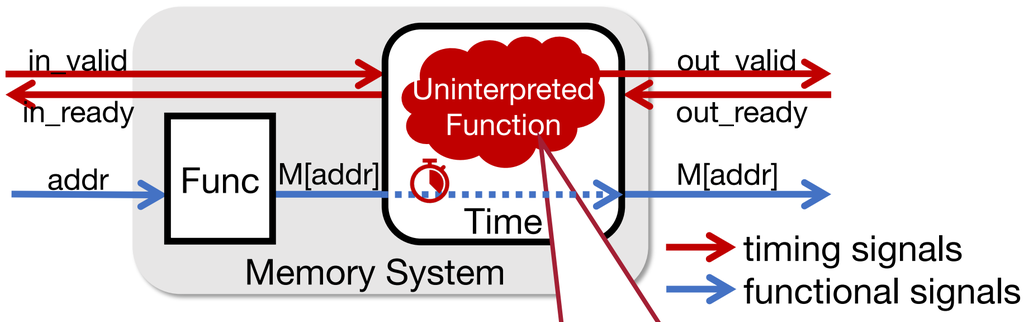
II. Challenge: Bridge the Gap



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

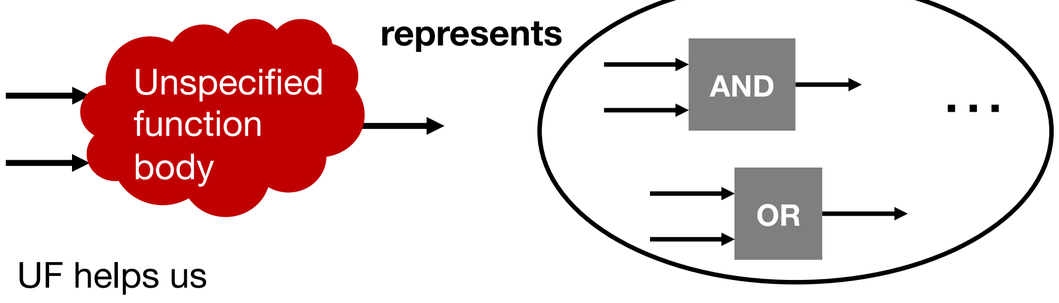
III. Pensieve Microarchitectural Modeling

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



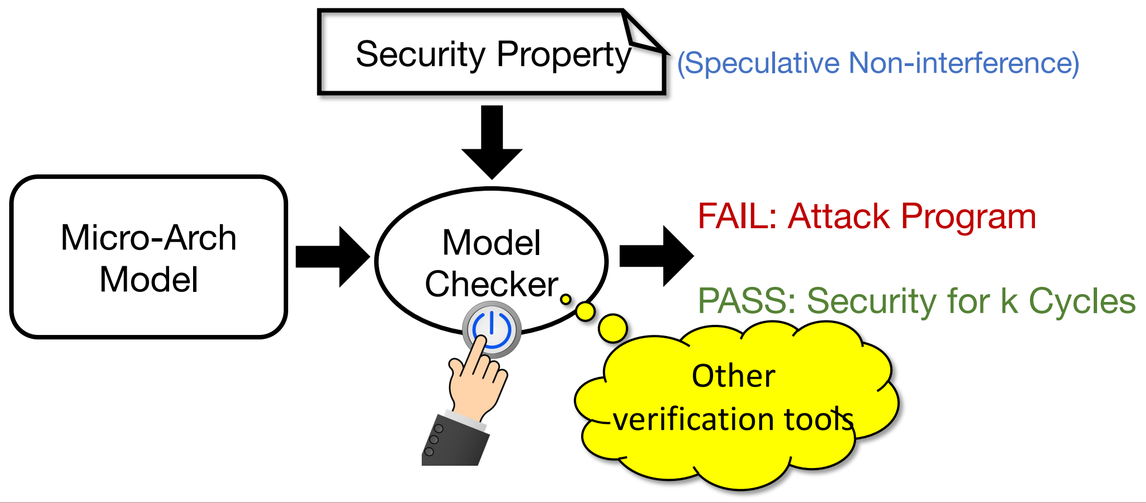
What is Uninterpreted function?

- A UF represents space of functions with same input/output types
  - Example: Bool UF(Bool, Bool)

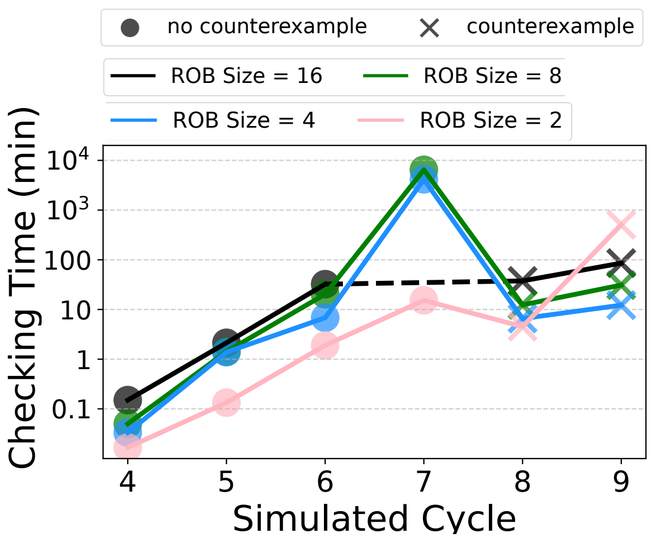


- UF helps us
  - state "**what**" affects the output,
  - abstract away the details on "**how**" the input affects the output
- Use UF to represent a space of timing behavior
  - `Memory_latency = UF(historyOf(in_valid, in_addr))`

IV. Pensieve Security Evaluation Framework



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 Penseive with more powerful formal verification backend

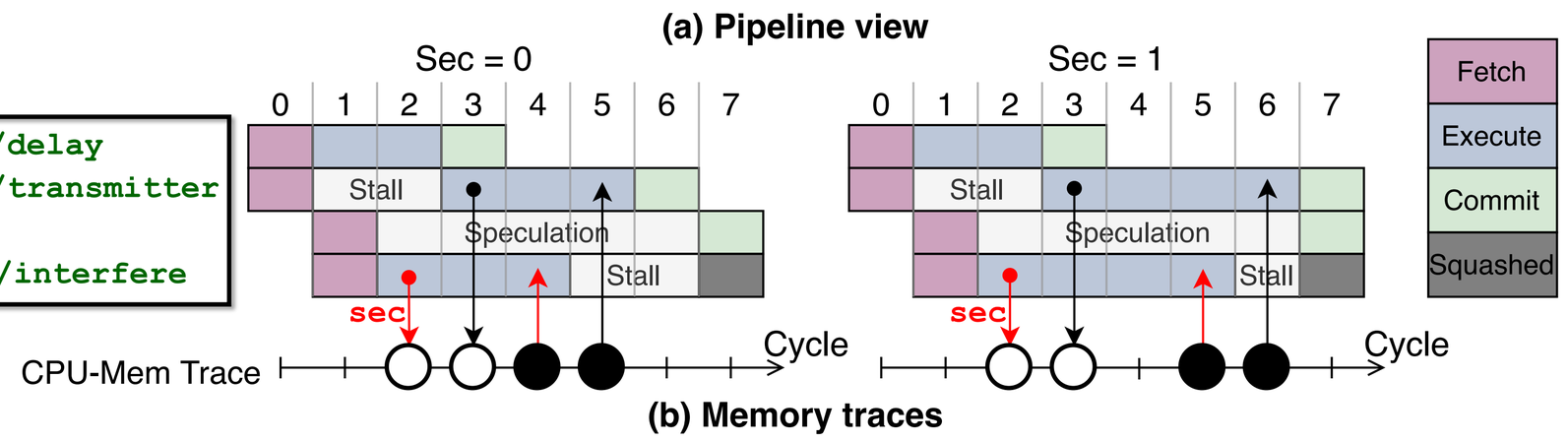
VI. New Attack Variant on GhostMinion

V-1. Spectre is MITIGATED by InvisiSpec

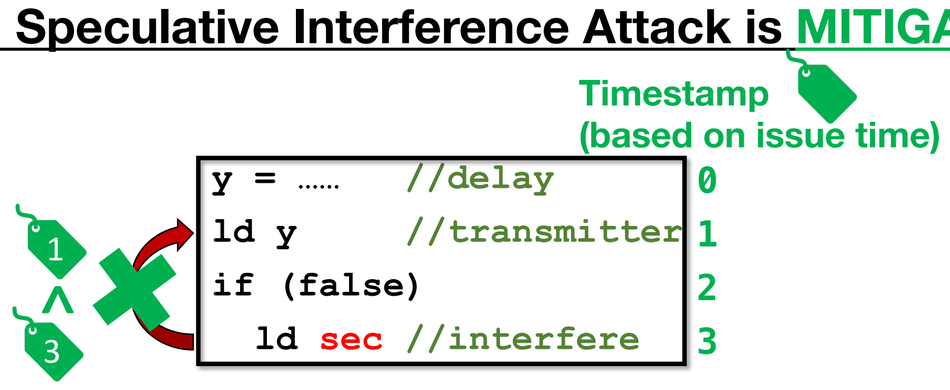
```
if (false)
  ld sec // transmitter
```

```
y = ...   //delay
ld y      //transmitter
if (false)
  ld sec //interfere
```

V-2. Speculative Interference Attack BREAKS InvisiSpec

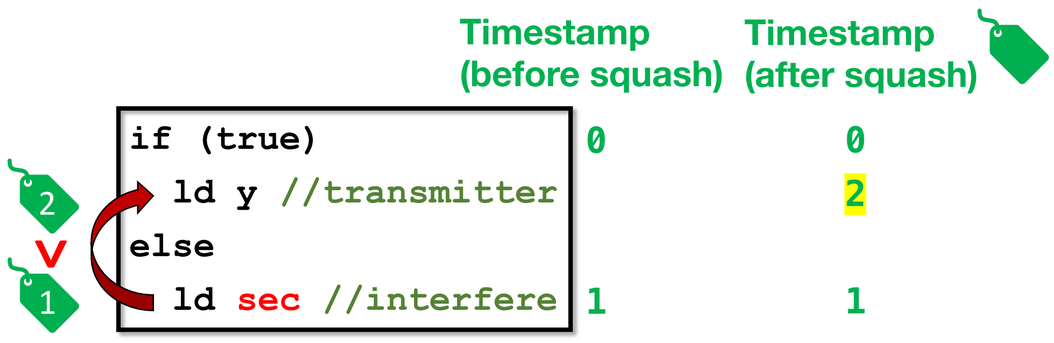


V-3. Speculative Interference Attack is MITIGATED by GhostMinion



"`ld sec`" is after "`ld y`" in program order

V-4. New Attack Variant BREAKS GhostMinion



V-5. Summary

No program order between "`ld sec`" and "`ld y`"