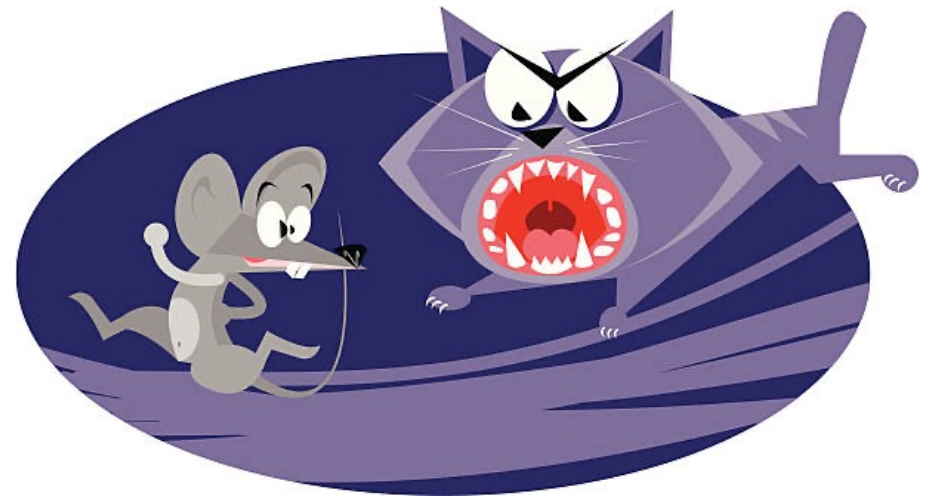
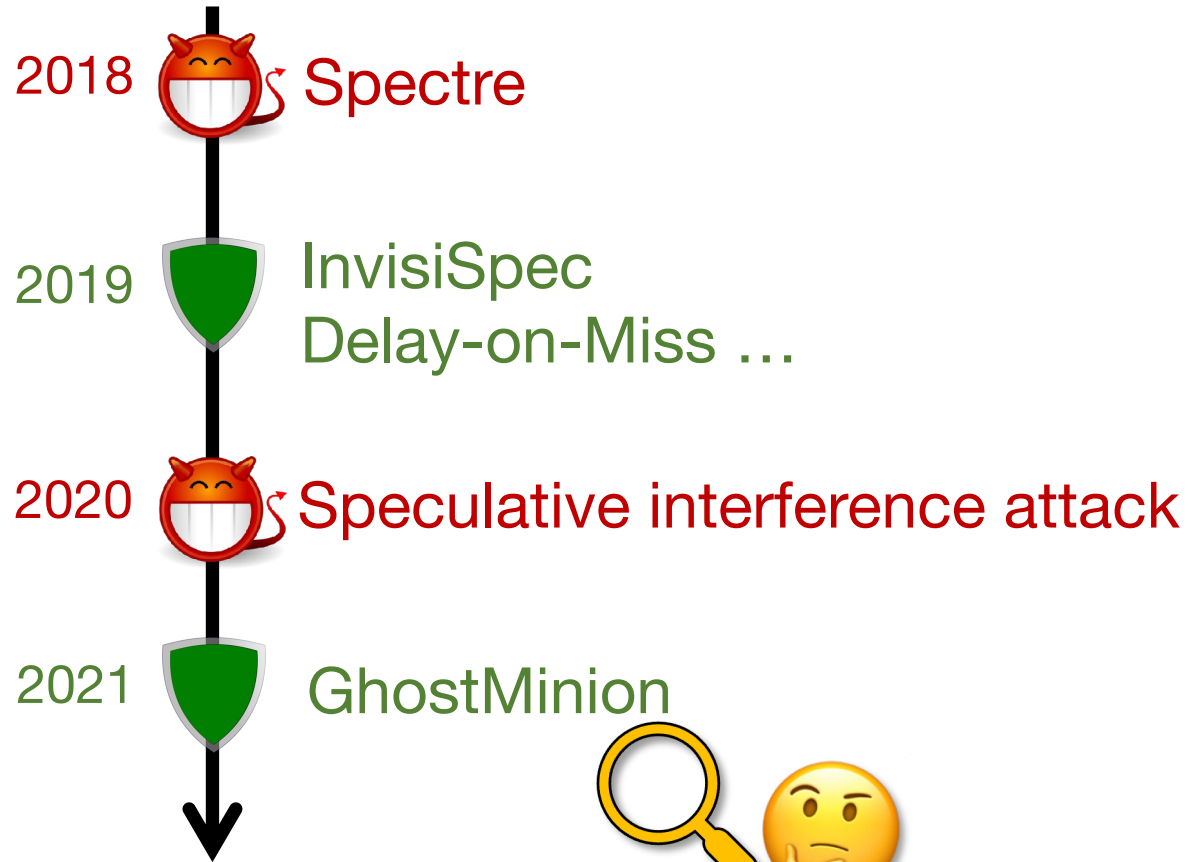


# **Pensieve: Microarchitectural Modeling for Security Evaluation**

**Yuheng Yang, Thomas Bourgeat, Stella Lau, Mengjia Yan**



# Problem: the Cat-and-Mouse Game

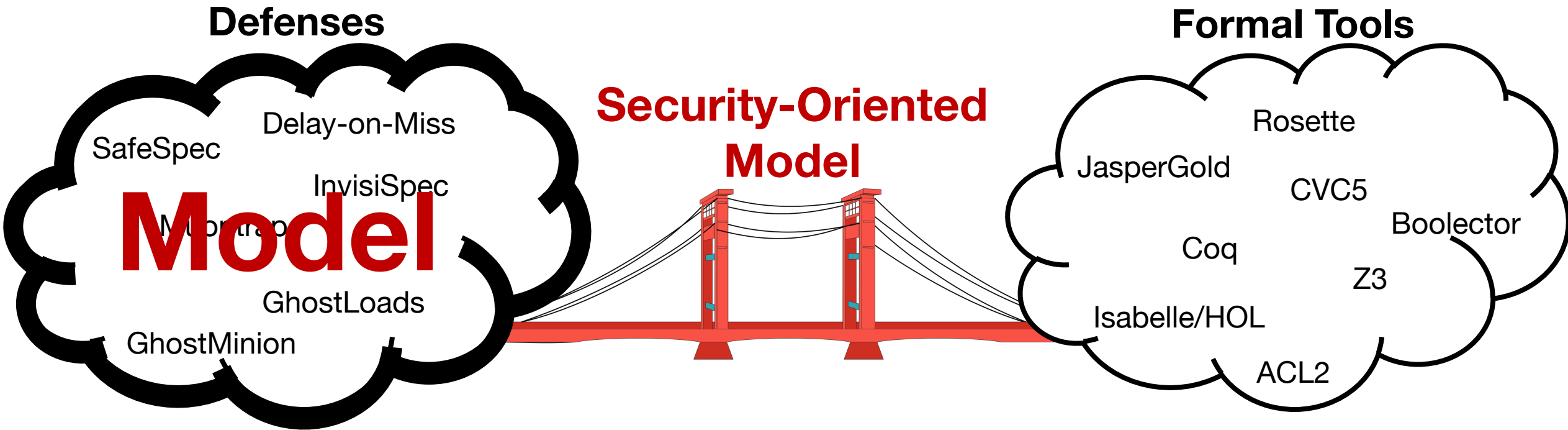


# Problem: Weak Security Evaluation



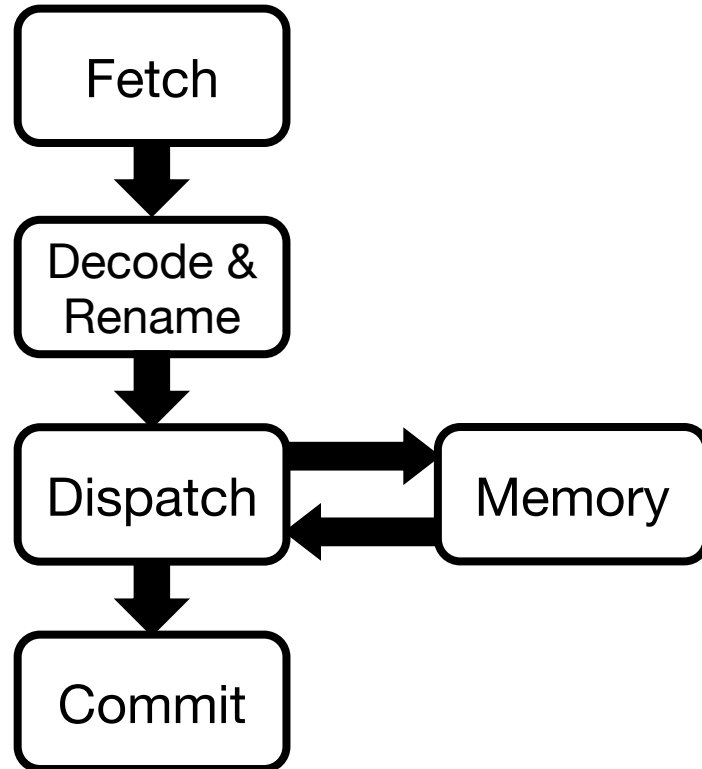
**We need a principled, trustworthy security evaluation framework!**

# Pensieve's Contribution



**Aligned** with architectural design flow.

# Defense Design Flow



Example: delay speculative requests

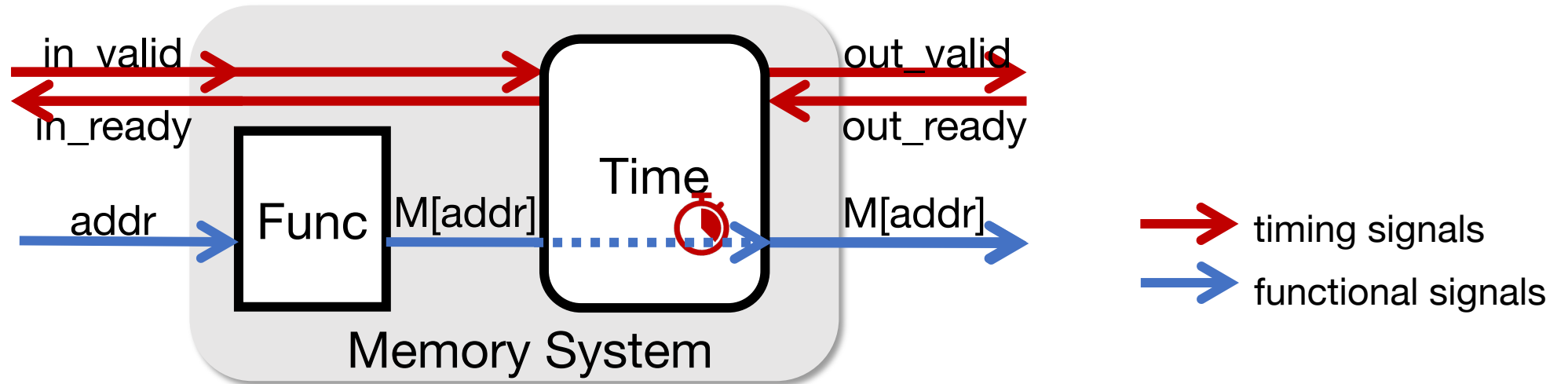


A modeling method should be:

1. Modular
2. Precise on describing timing behaviors
3. Represent a space of designs

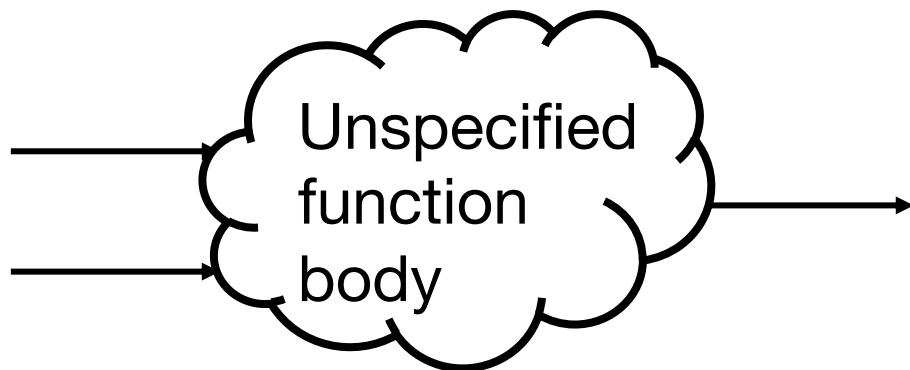
# Pensieve Modeling

#1 Decouple timing and functionality using the hand-shaking interface

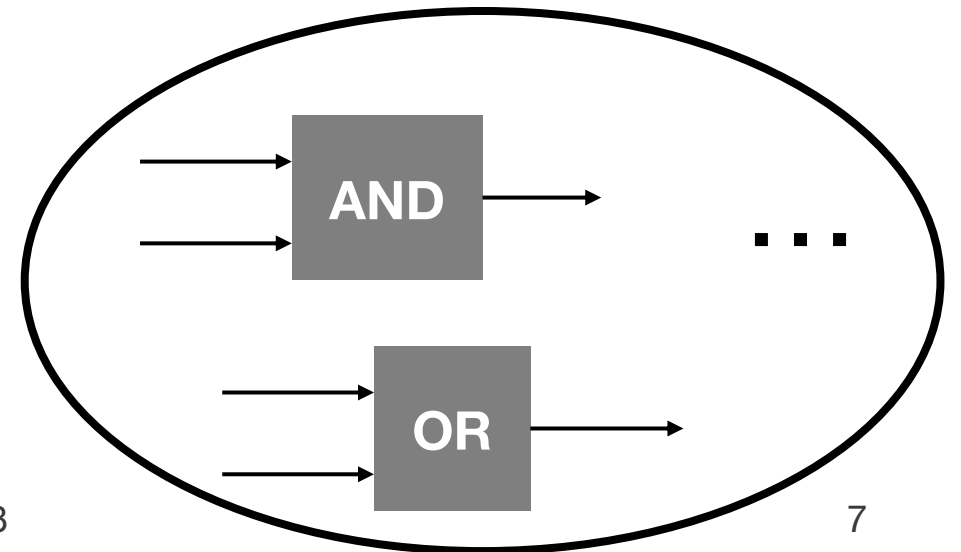


# Uninterpreted Function (UF)

- A UF represents space of functions with the 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



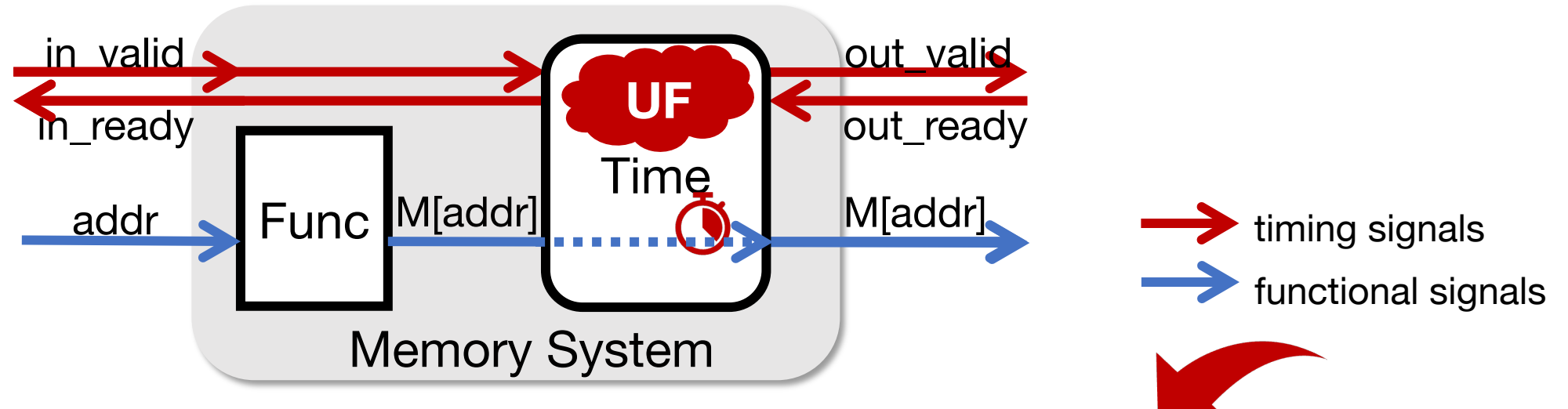
represents



# Pensieve Modeling

#1 Decouple timing and functionality using the hand-shaking interface

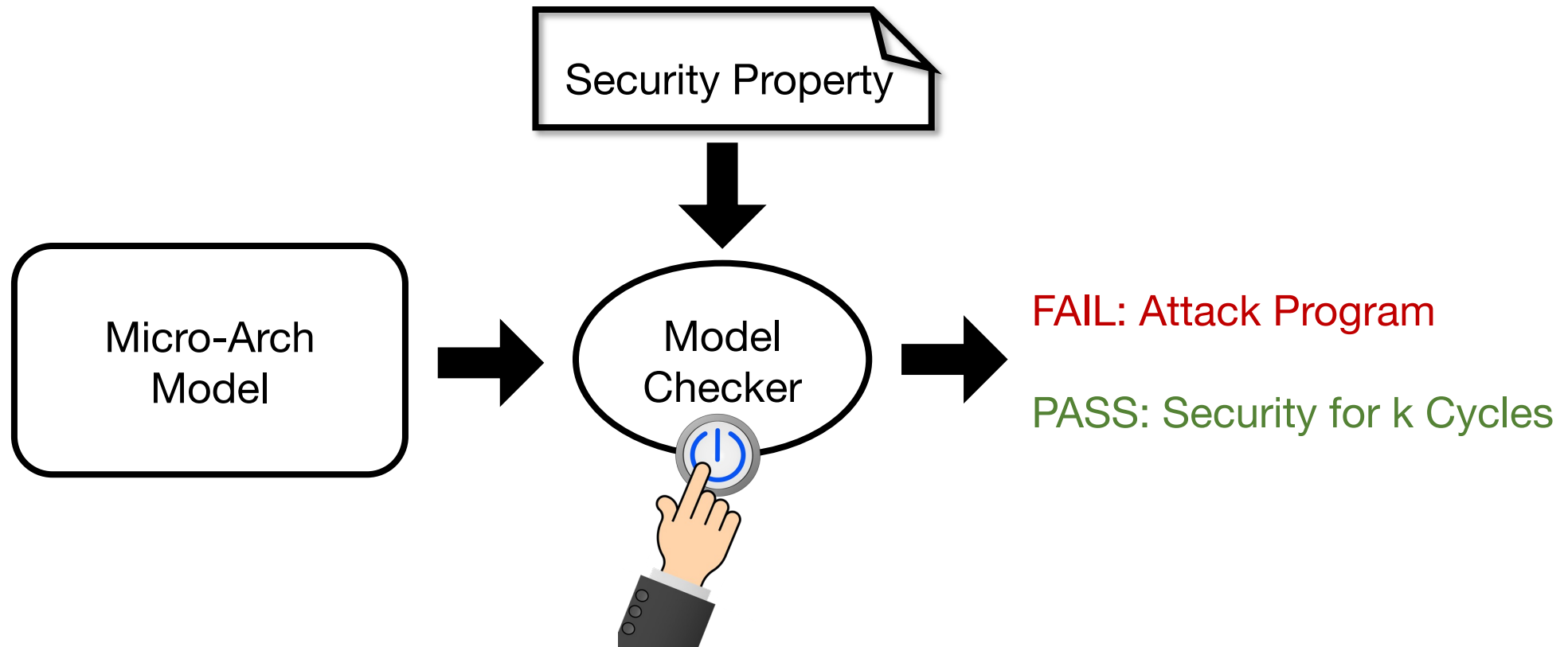
#2 Represent a space of timing behavior with uninterpreted functions



Pensieve uses **simple** models with **UF** to cover **space** of microarchitectures with **complex** timing behaviors



# Pensieve Security Evaluation Framework



Pensieve finds **unknown** security vulnerabilities in GhostMinion, the latest speculative execution defense

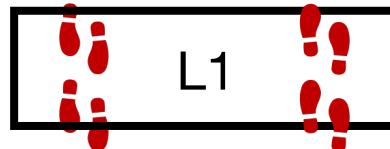
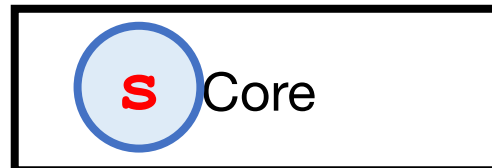
# GhostMinion

## #1: Invisible Speculation

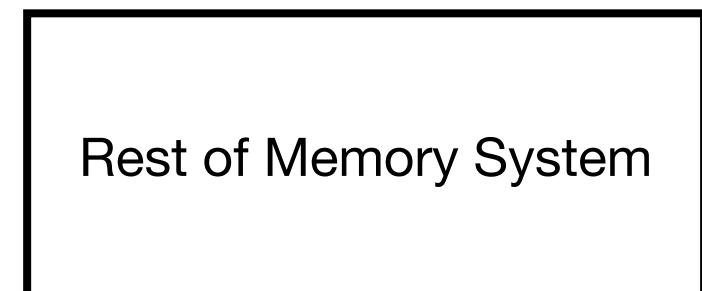
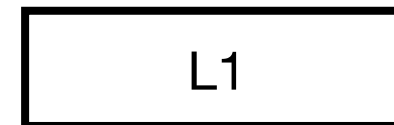
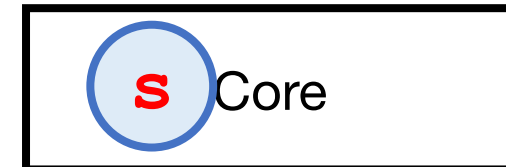
### Spectre v1

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

### Insecure Baseline



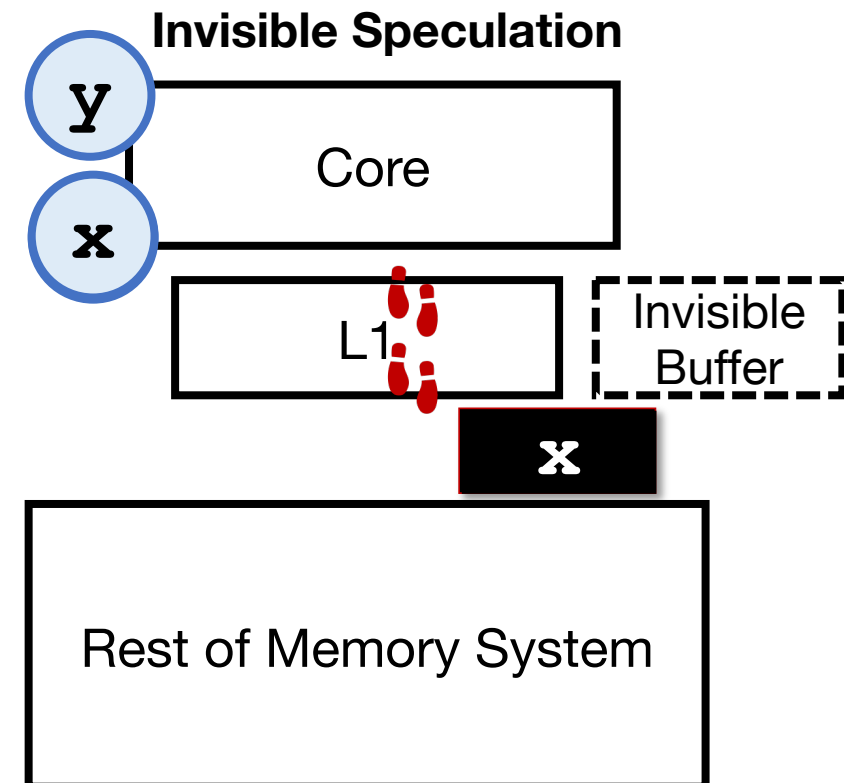
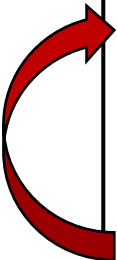
### Invisible Speculation



# Speculative Interference Attack

- **Younger** speculative loads interfere with **older** bound-to-commit loads.
- Many other contention structures: non-pipelined ALU, cache port, bank contention, network-on-chip, etc.

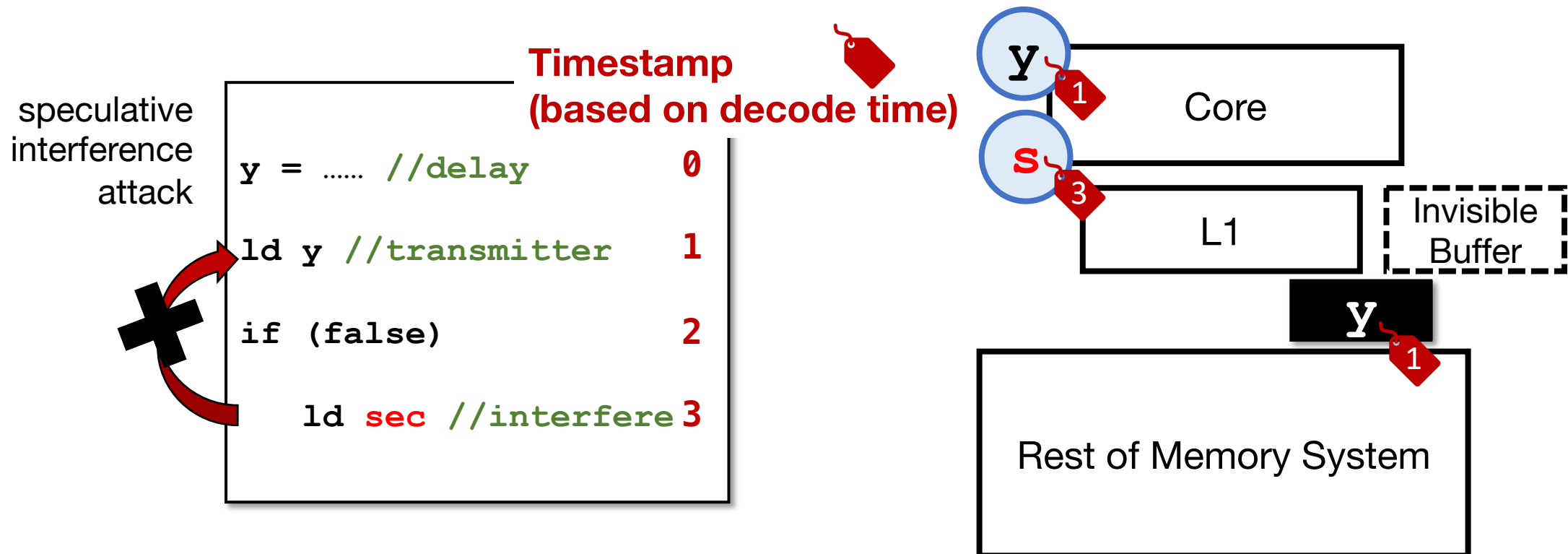
```
y = ..... //delay
ld y // transmitter
if (false)
    ld sec // interfere
```



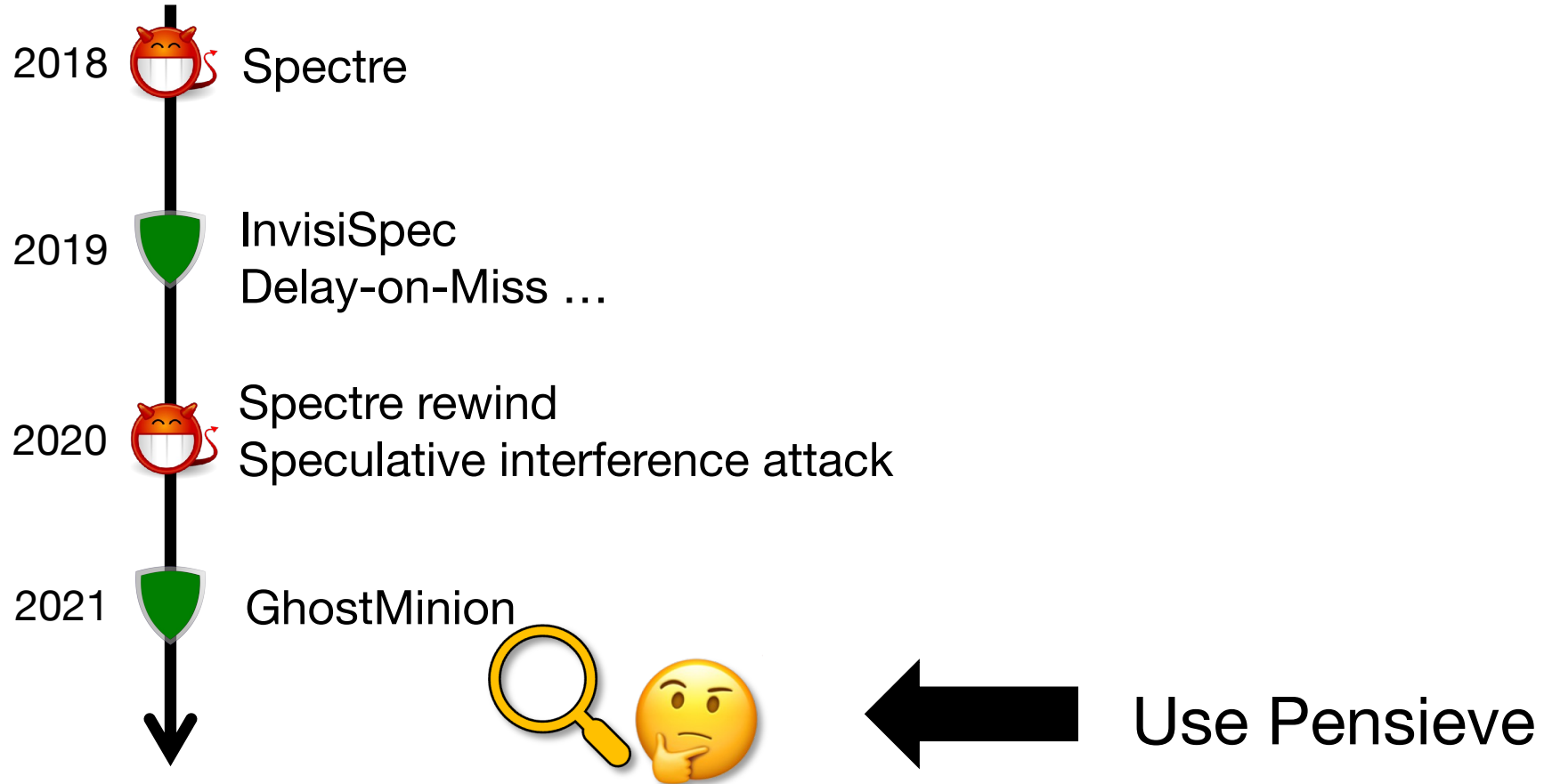
# GhostMinion

#1: Invisible Speculation

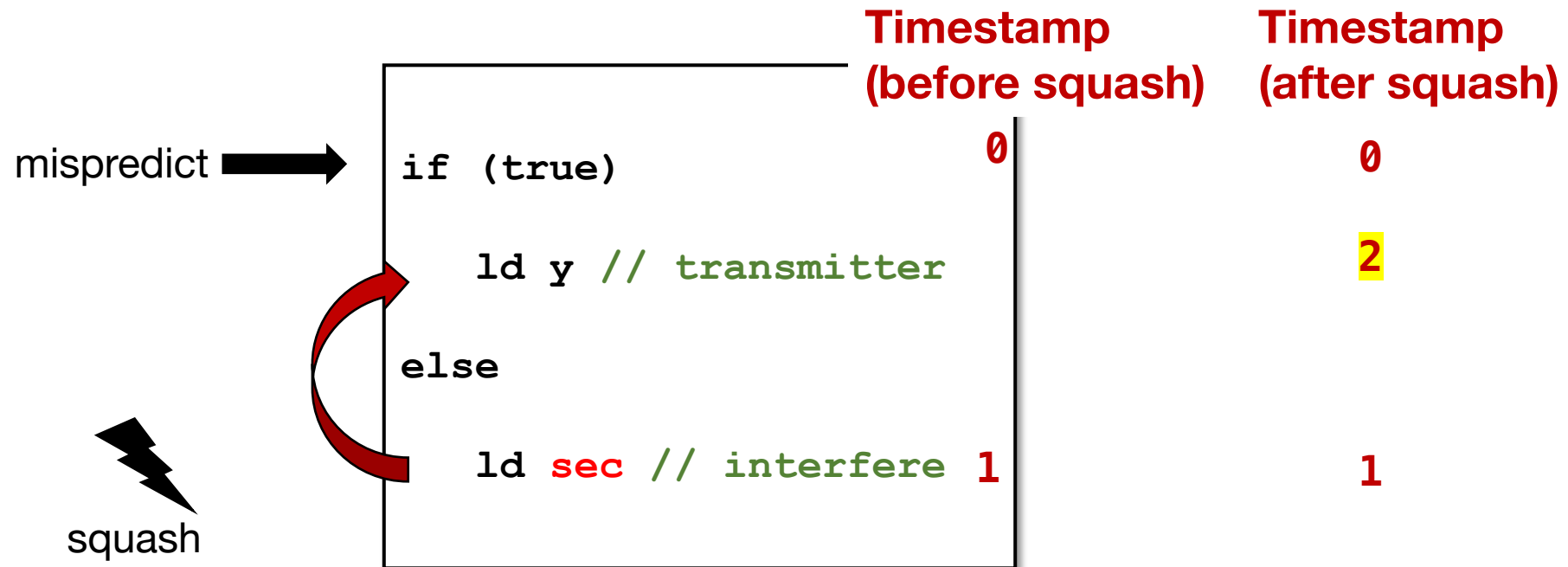
#2: Prioritize Older Instructions through Timestamps



# So Far ...



# Pensieve Found A New Attack Variant



Speculative load is older this time!

→ Speculative load can interfere with bound-to-commit load

# New Attack on GhostMinion Summary

speculative interference attack

	<code>y = .....</code>	<code>0</code>
Older	<code>ld y // transmitter</code>	<code>1</code>
	<code>if (false)</code>	<code>2</code>
Younger	<code>ld sec // interfere</code>	<code>3</code>

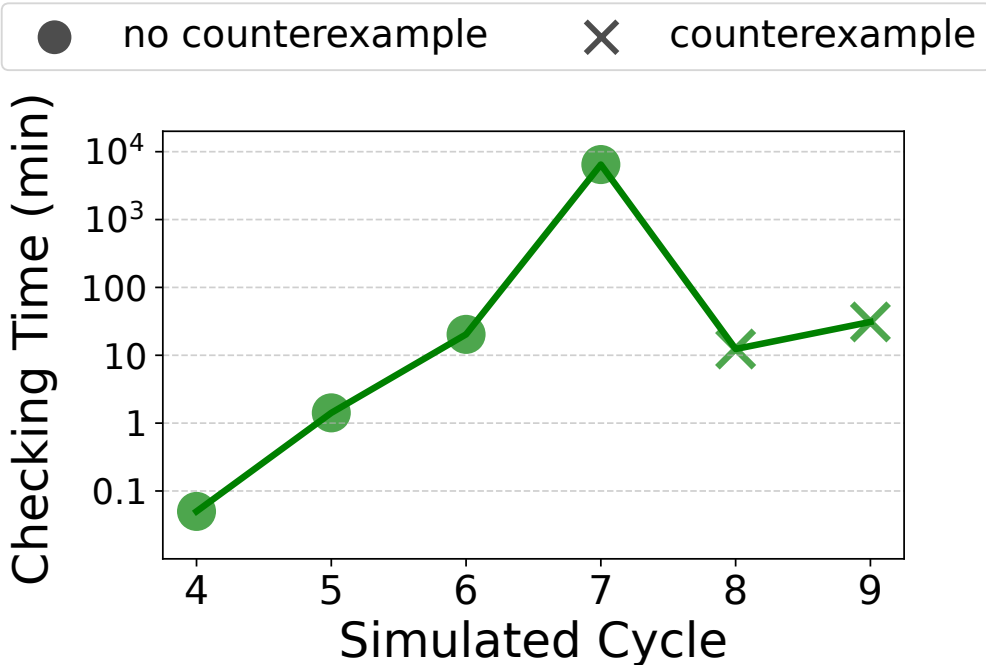


new attack variant

	<code>if (true)</code>	<code>0</code>
	<code>ld y // transmitter</code>	<code>2</code>
	<code>else</code>	
	<code>ld sec // interfere</code>	<code>1</code>

**Takeaway: Manual evaluation can easily be unsound, we need Pensieve, a trustworthy evaluation tool**

# 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



# Pensieve Summary

- Pensieve provides a modeling principle that **aligns** with architecture design flow, and **links** computer architects to accessible formal-methods tools.
- Pensieve finds **unknown** security vulnerabilities in GhostMinion

