Synthesizing Hardware-Software Leakage Contracts for RISC-V Open-Source Processors

Gideon Mohr Marco Guarnieri Jan Reineke

SIC Saarland Informatics Campus
UNIVERSITÄT DES SAARLANDES
institute IMDEA software

The Need for Hardware-Software Leakage Contracts

High-level language — **Independent** of microarchitecture

No guarantees about side channels

Instruction Set Architecture (ISA)

Impossible to program securely!

Microarchitecture — Allows **arbitrary optimizations**

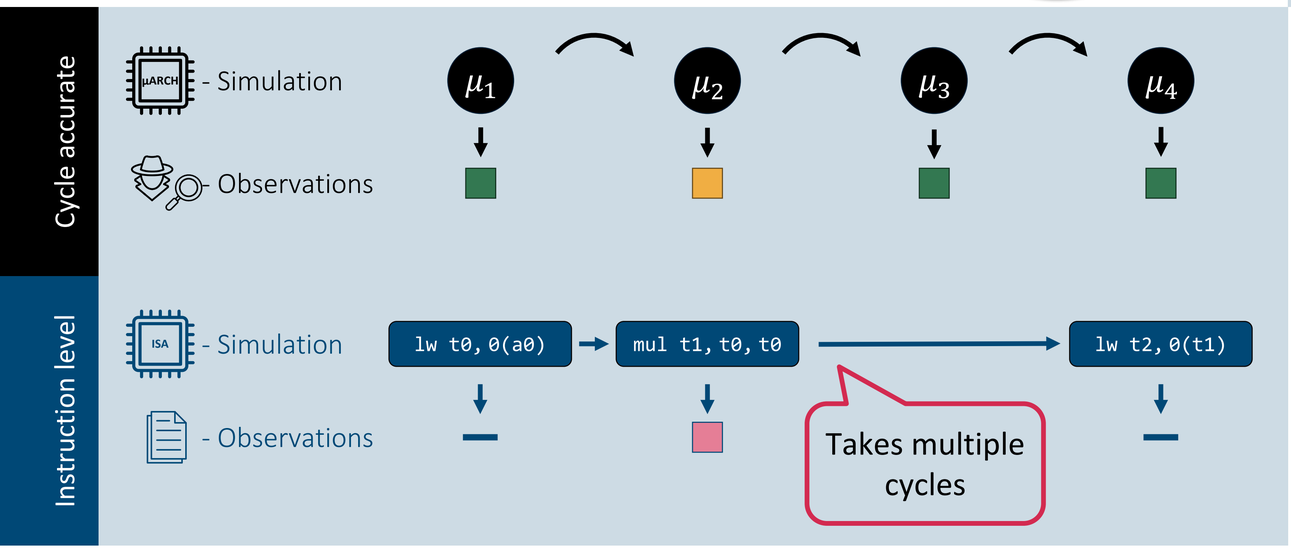
High-level language — Program **securely** using the contract

Captures possible information leakage at ISA level

Leakage Contracts: ISA + Instruction-level leakage specification

Microarchitecture

? But how can we obtain a leakage contract for **existing** microarchitectures?



A correct contract must satisfy this implication:

$$\text{attacker}_1 \neq \text{attacker}_2 \Rightarrow \text{contract}_1 \neq \text{contract}_2$$

Attacker can distinguish two executions

Contract yields different observations

! The contract captures what makes two executions attacker distinguishable

Defining a Space of Contracts

"For every division, the divisor leaks due to timing and
for every branch we can observe whether the branch was taken."

Condition on ISA state — Applies to instructions of a certain type

Leakage Part of the ISA state — Leaks a property of the instruction's inputs

Composition of multiple contract atoms

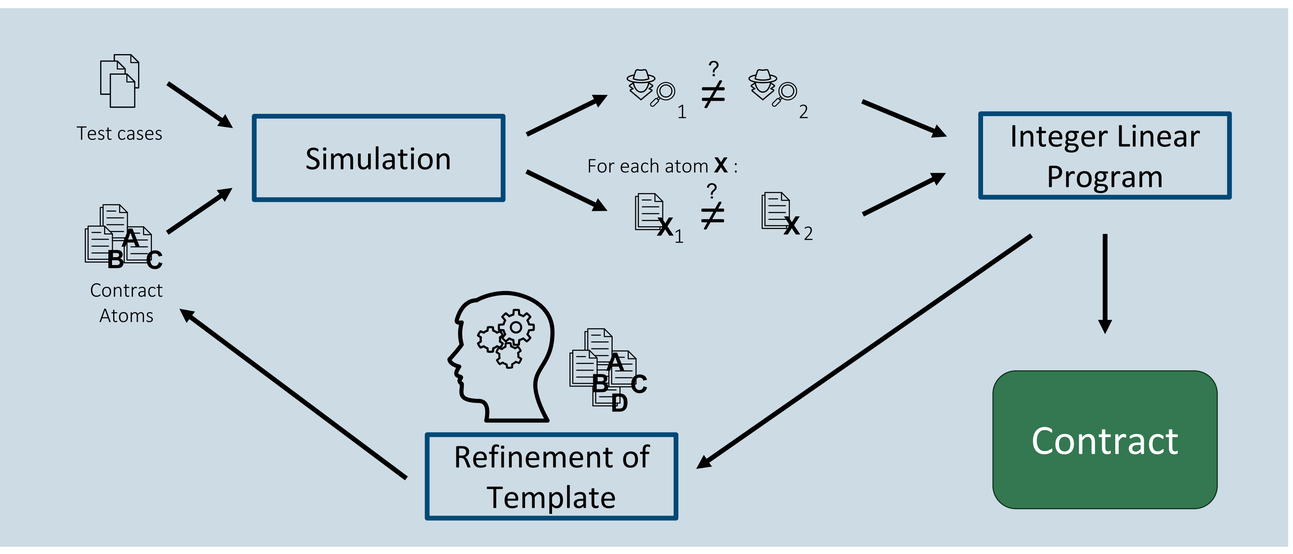
Instruction type — Register names & values

```
mul a0, t0, t1      a0 ← t0 * t1
lw  a1, -4(t2)      a1 ← mem[t2 + (-4)]
```

Immediate values — Memory addresses & values

What could leak about the executed instruction?

Synthesizing Contracts from Simulation Results



Contract Synthesis as ILP

Objective function:

$$\min \sum_{i \in Indist.} cover_i$$

Minimize false positives

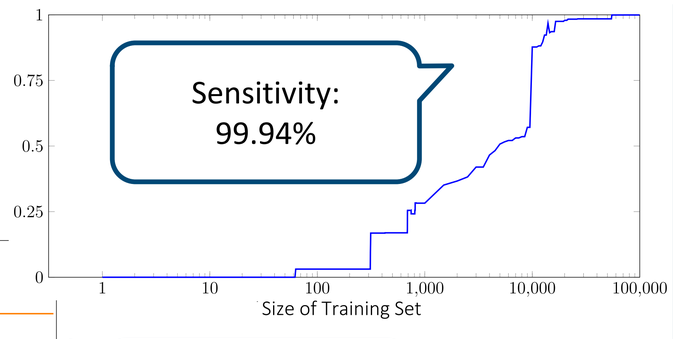
Constraints:

Correctness: $\sum_{a \in \texttt{dist\_by}(d)} select_a \geq 1$ — for all attacker-distinguishable test cases *d*

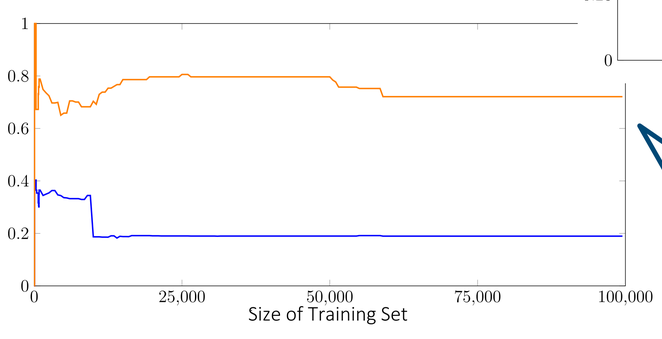
Precision: $\forall a \in \texttt{dist\_by}(i).\ select_a \leq cover_i$ — for all attacker-indistinguishable test cases *i*

Experimental Evaluation

Obtained on the Ibex core, w.r.t an evaluation set with 2,000,000 test cases.



Sensitivity: 99.94%



Precision after refinement: 72.11%

Refining the contract template

```
li a0, 0x100      a0 ← 0x100
lw a1, 0(a0)      a1 ← mem[a0]
```

Program 1

```
li a0, 0x102      a0 ← 0x102
lw a1, 0(a0)      a1 ← mem[a0]
```

Program 2

Leak alignment of addresses

For more details, have a look at our paper and the presentation on YouTube.
All results are available on Zenodo.