中国科学院软件研究所
Institute of Software Chinese Academy of Sciences

中国科学院大学
University of Chinese Academy of Sciences

Efficient and Accurate Security Verification in Probing model

Feng Zhou[1,2], Yiming Yang[3], Hua Chen[2], Limin Fan[2], An Wang[3]

[1] University of Chinese Academy of Sciences,
[2] TCA Laboratory, Institute of Software, Chinese Academy of Sciences,
[3] Beijing Institute of Technology

OPTIMIST 2025@Kuala Lumpur, Malaysia

Motivation

Problem

Various security notions has been proposed through the development of masking.

✓ **Existing formal verification tools that verify various security notions**
- maskVerif: Fast but not accurate
- IronMask: Fast but not accurate, only limited to standard gadgets.
- SILVER: Accurate but slow

Goal

✓ **Tools that are both efficient and accurate**

- **Probing security: Prover [TCHES 2025]**
- **NI, SNI, PINI: ProverNG (this talk's focus) [ICICS 2025]**
- **They are both based on SILVER [AsiaCrypt 2020].**

Background

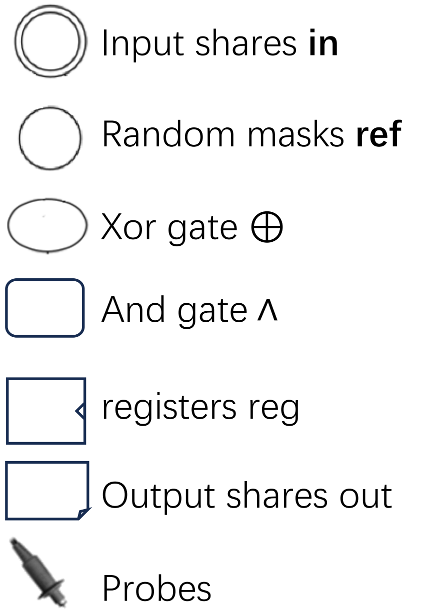
Masked Circuits and Security Models

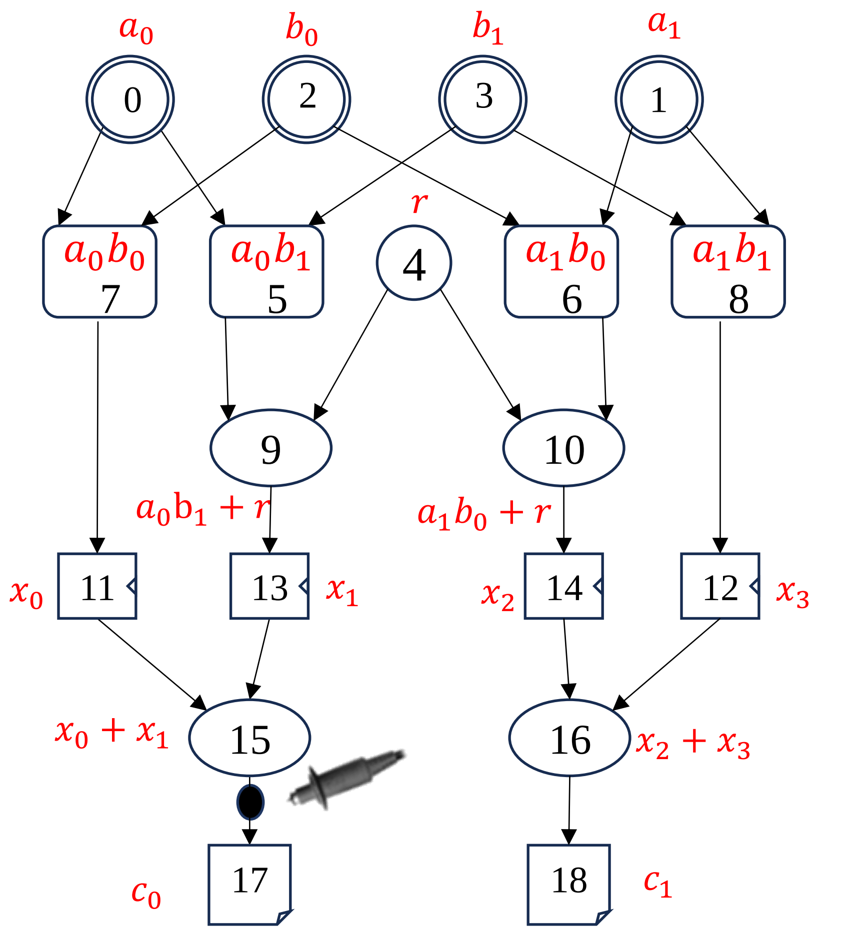
Secret variables:

$a = a_0 + a_1, b = b_0 + b_1$

✓ Masked Circuits

- A direct acyclic gragh (DAG)
  - $Nodes$: Gates = {in, out, ref, reg, not, and, xor, or, nand, xnor, nor}
  - $Edges$: Wires
  - Each node corresponds to a function





$$\begin{array}{lcccc}
g_0(a_0, b_0) & = & a_0 b_0 & \to x_0 & \\
g_1(a_0, b_1, r) & = & a_0 b_1 + r & \to x_1 & x_0 + x_1 = c_0 \\
g_2(a_1, b_0, r) & = & a_1 b_0 + r & \to x_2 & x_2 + x_3 = c_1 \\
g_3(a_1, b_1) & = & a_1 b_1 & \to x_3 &
\end{array}$$

Background

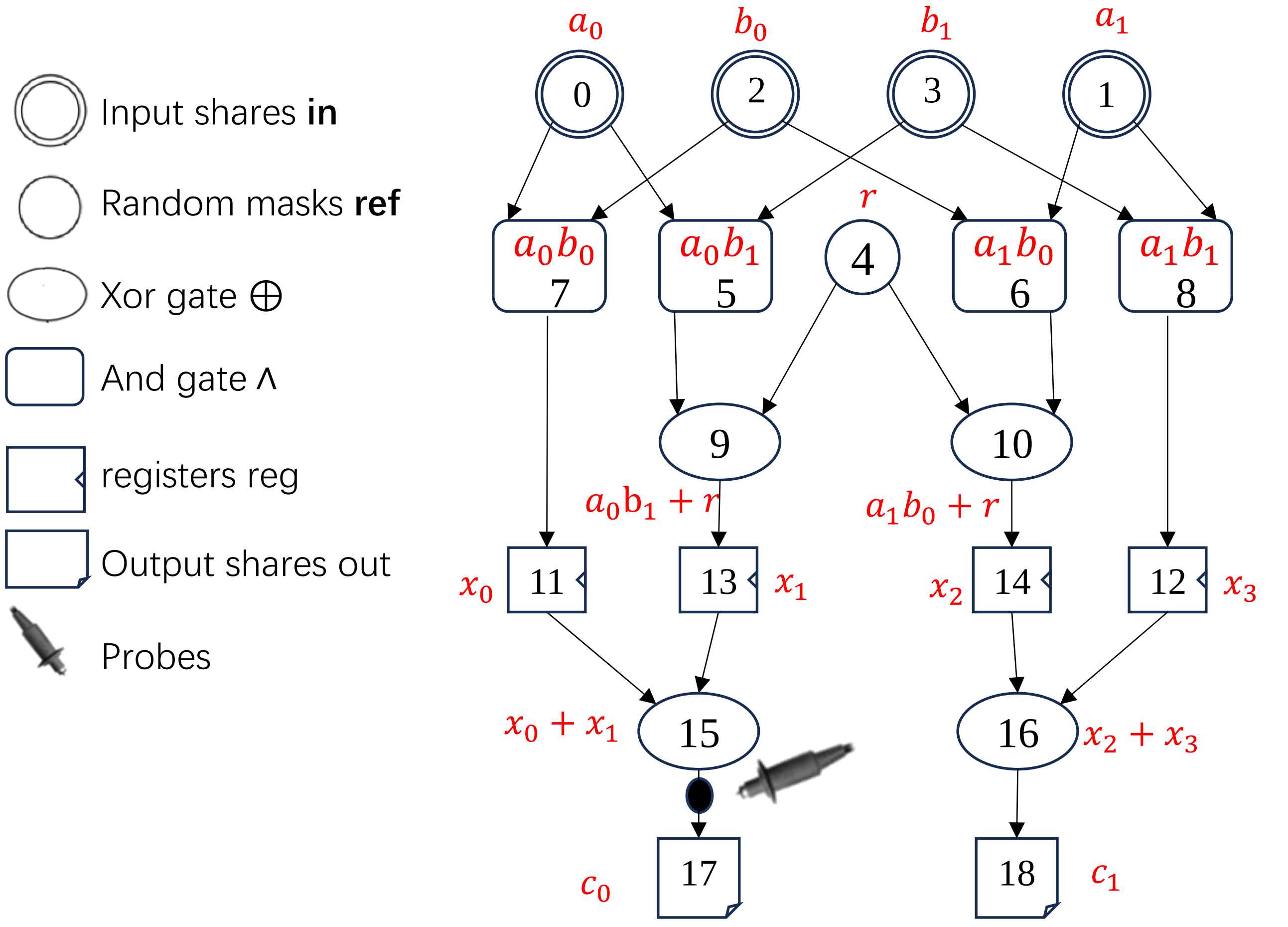
Masked Circuits and Security Models

Secret variables:

$$a = a_0 + a_1, b = b_0 + b_1$$

✓ Security Models

- Standard probing model
  - A Probe placed at output wire of gate 15
    - $O_d = \{x_0 + x_1\}$
- Glitch-extended probing model
  - A Probe placed at output wire of gate 15
    - $O_d = \{x_0, x_1\}$
- Definition of d probing security
  - Any observation set given by d probes is **statistically independent** of secret inputs to the masked circuit.



Composability and Statistical Independence

d probing security is not composable!

Security notions for composability: NI, SNI, PINI

Verification for NI, SNI, and PINI: For each observation $\boldsymbol{Q}$, find a simulation set $\boldsymbol{S}$ which simulates $\boldsymbol{Q}$ under the set of input shares $Sh(X)$.

$\boldsymbol{S}$ simulates $\boldsymbol{Q}$: The distribution of $\boldsymbol{Q}$ depends only on the shares in $\boldsymbol{S}$ but not shares in $\bar{S} = Sh(\boldsymbol{X}) \setminus \boldsymbol{S}$.

If $\boldsymbol{S}$ simulates $\boldsymbol{Q}$, $\boldsymbol{Q} \cup \boldsymbol{S}$ is statistically independent of $\overline{\boldsymbol{S}}$ where $Sh(\boldsymbol{X}) = \boldsymbol{S} \uplus \overline{\boldsymbol{S}}$. [KSM20]

Verification Process of SILVER

Motivation

Verifying 1-NI: whether the distribution of $\boldsymbol{Q}$ only depends on at most 1 share of both a and b?

Input shares: $a = a_0 + a_1, b = b_0 + b_1$

Obesevation set: $q_0 = a_0 b_0, q_1 = a_0 b_1 + r_1$

A subset of $supp(Q) \cap Sh(X)$

Which combination of $\boldsymbol{Q} \cup \boldsymbol{S}$ and $\overline{\boldsymbol{S}}$ is statistically independent of each other?

$\boldsymbol{Q}$:	Possible $\boldsymbol{S}$:	$\boldsymbol{Q} \cup \boldsymbol{S}$:	Corresponding $\overline{\boldsymbol{S}}$:
	$\emptyset$	$\{a_0b_0, a_0b_1 + r\} \cup \emptyset$	$\{a_0, a_1, b_0, b_1\}$
$\{a_0b_0, a_0b_1 + r\}$	$\{a_0\}$	$\{a_0b_0, a_0b_1 + r\} \cup \{a_0\}$	$\{a_1, b_0, b_1\}$
	$\{b_0\}$	$\{a_0b_0, a_0b_1 + r\} \cup \{b_0\}$	$\{a_0, a_1, b_1\}$
	$\{b_1\}$	$\{a_0b_0, a_0b_1 + r\} \cup \{b_1\}$	$\{a_0, a_1, b_0\}$
	$\{a_0, b_0\}$	$\{a_0b_0, a_0b_1 + r\} \cup \{a_0, b_0\}$	$\{a_1, b_1\}$
	$\{a_0, b_1\}$	$\{a_0b_0, a_0b_1 + r\} \cup \{a_0, b_1\}$	$\{a_1, b_0\}$

ROBDD verification complexity: $(2^{|\boldsymbol{Q} \cup \boldsymbol{S}|} - 1) \times (2^{|\overline{\boldsymbol{S}}|} - 1)$

How to find $\boldsymbol{S}$ efficiently?

How to reduce the size of $\boldsymbol{Q} \cup \boldsymbol{S}$ and $\overline{\boldsymbol{S}}$?

Reduction Rules

Reduction rules

How to reduce the size of $\boldsymbol{Q} \cup \boldsymbol{S}$ and $\overline{\boldsymbol{S}}$

1. If $q \in \boldsymbol{Q}$ is protected by a perfect mask that is not used by $\boldsymbol{Q} \setminus \{q\}$, $\boldsymbol{Q} \cup \boldsymbol{S}$ can be reduced to $\boldsymbol{Q} \cup \boldsymbol{S} \setminus \{q\}$.
2. If $s \notin sh(X) \cap supp(\boldsymbol{Q})$, $\overline{\boldsymbol{S}}$ can be reduced to $\overline{\boldsymbol{S}} \setminus \{s\}$.

Which combination of $\boldsymbol{Q} \cup \boldsymbol{S}$ and $\overline{\boldsymbol{S}}$ is statistically independent of each other?

$\boldsymbol{Q} \cup \boldsymbol{S}$:	Corresponding $\overline{\boldsymbol{S}}$:
$\{a_0b_0, a_0b_1 + r\} \cup \emptyset$	$\{a_0, a_1, b_0, b_1\}$
$\{a_0b_0, a_0b_1 + r\} \cup \{a_0\}$	$\{a_1, b_0, b_1\}$
$\{a_0b_0, a_0b_1 + r\} \cup \{b_0\}$	$\{a_0, a_1, b_1\}$
$\{a_0b_0, a_0b_1 + r\} \cup \{b_1\}$	$\{a_0, a_1, b_0\}$
$\{a_0b_0, a_0b_1 + r\} \cup \{a_0, b_0\}$	$\{a_1, b_1\}$
$\{a_0b_0, a_0b_1 + r\} \cup \{a_0, b_1\}$	$\{a_1, b_0\}$

Complexity: $(2^{|\boldsymbol{Q} \cup \boldsymbol{S}|} - 1) \times (2^{|\overline{\boldsymbol{S}}|} - 1)$

Domain Inference

Domain Inference

Assign each intermediate variable a domain!

We distinguish three types of domains
Share domain: $\mathrm{I} = \{0,1,\cdots,d\}$
Random domain: R
Unknown domain: U

Guessing rules	Observations	Simulation set
$Dom(q) = i \Rightarrow a_i, b_i \in \boldsymbol{S}$	$a_2 \cdot b_2$	$\{a_2, b_2\}$
$Dom(q) = R \Rightarrow \boldsymbol{S} = \emptyset$	$a_2 \cdot b_1 + r_1$	$\emptyset$
$Dom(q) = U \Rightarrow \text{Enum } \boldsymbol{S}.$	$a_2 \cdot b_0$	?

$a_2 b_1 + r_1$ R

2 $a_2 b_2$

a_2 b_2

2 2

$a_0 b_1$ U

a_0 b_1

0 1

$Dom(a_2 b_1 + r_1) = \mathrm{R}$ $Dom(a_2 b_2) = 2 \in \mathrm{I}$ $Dom(a_0 b_1) = \mathrm{U}$

Examples of domain inference rules

Domain Inference

Domain Inference

For a gate: $n = n.lft \circ n.rgt$

When $perf(n) \neq \emptyset$, $D(n) = R$

$perf(n) = \emptyset$			
$D(\boldsymbol{n.lft})$	$D(\boldsymbol{n.rgt})$	$D(\boldsymbol{n})$	Example
$i \in \mathrm{I}$	$i \in \mathrm{I}$	$i \in \mathrm{I}$	$t_0 = a_0 \cdot b_0$
$i \in \mathrm{I}$	$j \in \mathrm{I}$	U	$t_1 = a_0 \cdot b_1$
U	R/I	U	Usually not the case
$i \in \mathrm{I}$	R	I	$t_2 = r_0 a_0$

Domain inference in Standard Probing Model

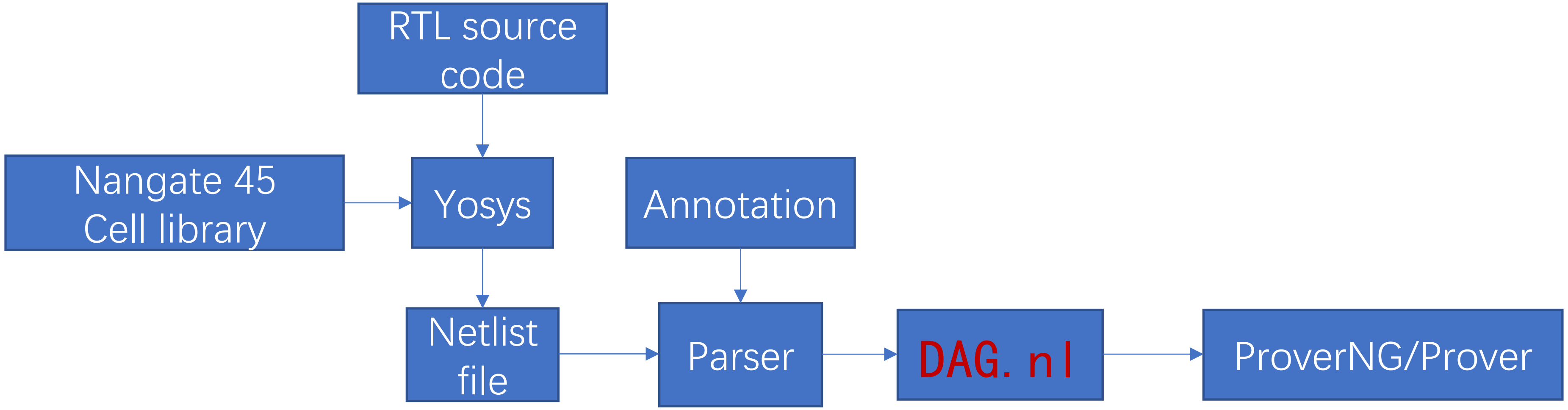
When $perf(n) \neq \emptyset$, and $op(n) = reg$, $D(n) = R$

$op(n) \neq \boldsymbol{reg}$			
$D(\boldsymbol{n.lft})$	$D(\boldsymbol{n.rgt})$	$D(\boldsymbol{n})$	Example
$i \in \mathrm{I}$	$i \in \mathrm{I}$	$i \in \mathrm{I}$	$t_0 = a_0 \cdot b_0$
$i \in \mathrm{I}$	$j \in \mathrm{I}$	U	$t_1 = a_0 \cdot b_1$
U	R/I	U	$t_2 = a_0 b_1 + r$
$i \in \mathrm{I}$	R	I	$t_3 = r_0 a_0$

Domain inference in Glitch-extended Probing Model

ProverNG – the tool

Work flow of ProverNG



Work flow of ProverNG

Evaluation

Benchmarks

- Order: 1-3
- Hardware Private Circuits
  - HPC1
  - HPC2
  - HPC3
- OPINI2

Security notions

Non-Interference

Strong Non-Interference

Probe Isolating Non-Interference

Evaluation

Comparison to SILVER, maskVerif, and IronMask

- d-NI verification results

Method	SILVER		maskVerif		IronMask		ProverNG	
Model	Std.	Gli.	Std.	Gli.	Std.	Gli.	Std.	Gli.
HPC1 o1	^{1}✓ **[<0.001s]**	^{1}✓ [0.001s]	^{1}✓ [0.001s]	^{1}✓ [0.001s]	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ [0.001s]	^{1}✓ **[<0.001s]**
HPC1 o2	^{2}✓ [0.047s]	^{2}✓ [0.223s]	^{2}✓ [0.002s]	^{2}✓ [0.005s]	^{2}✓ **[<0.001s]**	^{2}✓ **[<0.001s]**	^{2}✓ [0.024s]	^{2}✓ [0.030s]
HPC1 o3	^{3}✓ [12.330s]	^{3}✓ [318.736s]	^{3}✓ [0.027s]	^{3}✓ [0.027s]	^{3}✓ **[<0.001s]**	^{3}✓ **[<0.001s]**	^{3}✓ [4.343s]	^{3}✓ [20.590s]
HPC2 o1	^{1}✓ [0.001s]	^{1}✓ [0.001s]	^{1}✓ [0.001s]	^{1}✗ [0.001s]	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**
HPC2 o2	^{2}✓ [0.050s]	^{2}✓ [0.150s]	^{2}✓ [0.020s]	^{2}✗ [0.002s]	^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ **[0.017s]**	^{2}✓ **[0.024s]**
HPC2 o3	^{3}✓ [12.518s]	^{3}✓ [290.257s]	^{3}✗ [0.090s]	^{3}✗ [0.003s]	✖ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[2.968s]**	^{3}✓ **[41.125s]**
HPC3 o1	^{1}✓ **[0.001s]**	^{1}✓ [0.001s]	^{1}✓ **[0.001s]**	^{1}✓ [0.001s]	✖ [<0.001s]	^{1}✗ [<0.001s]	^{1}✓ **[0.001s]**	^{1}✓ **[<0.001s]**
HPC3 o2	^{2}✓ [0.085s]	^{2}✓ [1.161s]	^{2}✓ [0.005s]	^{2}✓ [0.003s]	^{2}✓ **[<0.001s]**	^{2}✗ [<0.001s]	^{2}✓ [0.037s]	^{2}✓ **[0.404s]**
HPC3 o3	^{3}✓ [17.422s]	^{3}✓ [19455.531s]	^{3}✓ **[0.039s]**	^{3}✓ **[0.018s]**	^{3}✗ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ [5.286s]	^{3}✓ [9246.018s]
OPINI2 o1	^{1}✓ [0.001s]	^{1}✓ **[<0.001s]**	^{1}✓ [0.001s]	^{1}✗ [0.001s]	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**
OPINI2 o2	^{2}✓ [0.086s]	^{2}✓ [0.360s]	^{2}✓ **[0.023s]**	^{2}✗ [0.002s]	^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ [0.034s]	^{2}✓ **[0.076s]**
OPINI2 o3	^{3}✓ [26.242s]	^{3}✓ [1083.457s]	^{3}✗ [0.065s]	^{3}✗ [0.002s]	✖ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[6.035s]**	^{3}✓ **[229.734s]**

Evaluation

Comparison to SILVER, maskVerif, and IronMask

- d-SNI verification results

Method	SILVER		maskVerif		IronMask		ProverNG	
Model	Std.	Gli.	Std.	Gli.	Std.	Gli.	Std.	Gli.
HPC1 o1	^{1}✓[0.001s]	^{1}✗**[<0.001s]**	^{1}✓[0.001s]	^{1}✗[0.001s]	^{1}✓**[<0.001s]**	^{1}✓[<0.001s]	^{1}✓[0.001s]	^{1}✗**[<0.001s]**
HPC1 o2	^{2}✓[0.037s]	^{1}✗[0.004s]	^{2}✓**[0.004s]**	^{2}✗[0.001s]	^{2}✗[<0.001s]	^{2}✗**[<0.001s]**	^{2}✓[0.018s]	^{1}✗**[<0.001s]**
HPC1 o3	^{3}✓[13.747s]	^{1}✗[0.030s]	^{3}✓**[0.053s]**	^{3}✗[0.002s]	^{3}✗[<0.001s]	^{3}✗**[<0.001s]**	^{3}✓[5.805s]	^{1}✗[0.001s]
HPC2 o1	^{1}✓**[<0.001s]**	^{1}✗**[<0.001s]**	^{1}✓[0.002s]	^{1}✗[0.001s]	^{1}✓**[<0.001s]**	^{1}✓[<0.001s]	^{1}✓[0.001s]	^{1}✗**[<0.001s]**
HPC2 o2	^{2}✓[0.046s]	^{1}✗[0.003s]	^{2}✓[0.016s]	^{2}✗[0.001s]	^{2}✗[<0.001s]	^{2}✗**[<0.001s]**	^{2}✓**[0.012s]**	^{1}✗**[<0.001s]**
HPC2 o3	^{3}✓[14.259s]	^{1}✗[0.029s]	^{3}✗[0.137s]	^{3}✗[0.003s]	✖[<0.001s]	✖[<0.001s]	^{3}✓**[3.667s]**	^{1}✗**[0.001s]**
HPC3 o1	^{1}✓**[<0.001s]**	^{1}✗**[<0.001s]**	^{1}✓[0.002s]	^{1}✗[0.001s]	✖[<0.001s]	✖[<0.001s]	^{1}✓**[<0.001s]**	^{1}✗**[<0.001s]**
HPC3 o2	^{2}✓[0.076s]	^{1}✗[0.001s]	^{2}✓[0.005s]	^{2}✗[0.001s]	^{2}✓**[<0.001s]**	^{2}✗**[<0.001s]**	^{2}✓[0.029s]	^{1}✗[0.001s]
HPC3 o3	^{3}✓[19.696s]	^{1}✗[0.006s]	^{3}✓**[0.071s]**	^{3}✗[0.002s]	^{3}✗[<0.001s]	^{3}✗**[<0.001s]**	^{3}✓[5.232s]	^{1}✗[0.001s]
OPINI2 o1	^{1}✓**[<0.001s]**	^{1}✓**[<0.001s]**	^{1}✓[0.002s]	^{1}✗[0.002s]	^{1}✓**[<0.001s]**	^{1}✓**[<0.001s]**	^{1}✓**[<0.001s]**	^{1}✓[0.001s]
OPINI2 o2	^{2}✓[0.073s]	^{2}✓[0.345s]	^{2}✓**[0.019s]**	^{2}✗[0.003s]	^{2}✗[<0.001s]	^{2}✗[<0.001s]	^{2}✓[0.024s]	^{2}✓**[0.061s]**
OPINI2 o3	^{3}✓[28.706s]	^{3}✓[995.418s]	^{3}✗[0.596s]	^{3}✗[0.010s]	✖[<0.001s]	✖[<0.001s]	^{3}✓**[6.671s]**	^{3}✓**[235.078s]**

Evaluation

Comparison to SILVER, maskVerif, and IronMask

- d-PINI verification results

Method	SILVER		maskVerif		IronMask		ProverNG	
Model	Std.	Gli.	Std.	Gli.	Std.	Gli.	Std.	Gli.
HPC1 o1	^{1}✓ **[<0.001s]**	^{1}✓ [0.001s]			^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**
HPC1 o2	^{2}✓ [0.060s]	^{2}✓ [0.269s]			^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ **[0.011s]**	^{2}✓ **[0.020s]**
HPC1 o3	^{3}✓ [27.569s]	^{3}✓ [324.608s]			^{3}✗ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[5.203s]**	^{3}✓ **[19.450s]**
HPC2 o1	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**			^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**
HPC2 o2	^{2}✓ [0.048s]	^{2}✓ [0.405s]			^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ **[0.006s]**	^{2}✓ **[0.027s]**
HPC2 o3	^{3}✓ [19.065s]	^{3}✓ [3275.217s]			^{3}✗ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[3.005s]**	^{3}✓ **[55.778s]**
HPC3 o1	^{1}✓ **[0.001s]**	^{1}✓ **[<0.001s]**	N/A		^{1}✗ [<0.001s]	^{1}✗ [<0.001s]	^{1}✓ **[0.001s]**	^{1}✓ **[<0.001s]**
HPC3 o2	^{2}✓ [0.067s]	^{2}✓ [1.785s]			^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ **[0.020s]**	^{2}✓ **[0.188s]**
HPC3 o3	^{3}✓ [21.269s]	^{3}✓ [23956.321s]			^{3}✗ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[4.582s]**	^{3}✓ **[2699.785s]**
OPINI2 o1	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**			^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**	^{1}✓ **[<0.001s]**
OPINI2 o2	^{2}✓ [0.084s]	^{2}✓ [1.000s]			^{2}✗ [<0.001s]	^{2}✗ [<0.001s]	^{2}✓ **[0.014s]**	^{2}✓ **[0.071s]**
OPINI2 o3	^{3}✓ [33.592s]	^{3}✓ [14508.664s]			^{3}✗ [<0.001s]	^{3}✗ [<0.001s]	^{3}✓ **[5.885s]**	^{3}✓ **[279.057s]**

Discussion and Conclusion

Bottlenecks

- Complexity
  - Exponential Explosion
    - ✓ S-boxes, round-functions: too many variables when constructing ROBDDs
  - Combinatorial Explosion
    - ✓ Higher order masking: too many combinations to be verified
  - …

Conclusion

- ProverNG, an efficient and sound tool to verify compositional security notions under probing model.

中国科学院软件研究所
Institute of Software Chinese Academy of Sciences

中国科学院大学
University of Chinese Academy of Sciences

Thank you!

ProverNG: https://github.com/Lucien98/ProverNG

Background

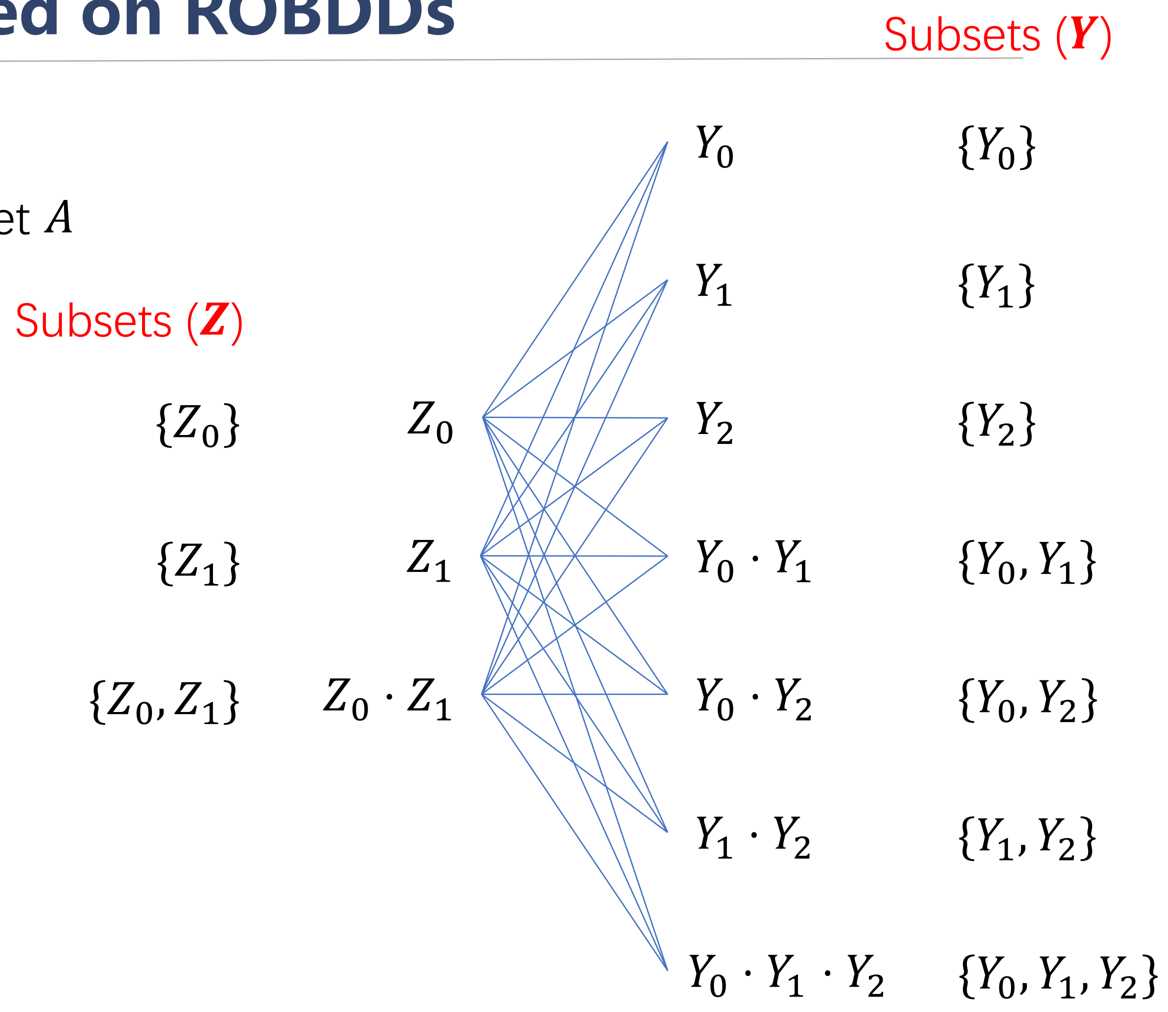
Statistical Independence Check Based on ROBDDs

Product over a set A: $\Pi_{x\in A}x$, the product of all elements in set A

[KSM20]: Given two sets of Boolean random variables, $\boldsymbol{Z} = \{Z_0, Z_1, \cdots, Z_{n-1}\}$ and $\boldsymbol{Y} = \{Y_0, Y_1, \cdots, Y_{m-1}\}$, $\boldsymbol{Z}$ is statistically independent of $\boldsymbol{Y}$ iff the product over any non-empty subset of $\boldsymbol{Z}$ is statistically independent of the product over any non-empty subset of $\boldsymbol{Y}$.

Verification complexity: $(2^n - 1)(2^m - 1)$

Slow Performance



Example: $n = 2, m = 3$

Domain Inference

Guessing share domains for the variables in the correct simulation set $\boldsymbol{S}$.

Input shares: $a = a_0 + a_1, b = b_0 + b_1$

We distinguish three types of domains
Share domain: $\mathrm{I} = \{0,1,\cdots,d\}$
Random domain: R
Unknown domain: U

$Dom(a_0) = 0 \in \mathrm{I}$

$Dom(r_0) = \mathrm{R}$

$Dom(a_0 b_1) = \mathrm{U}$

[·]: register stages