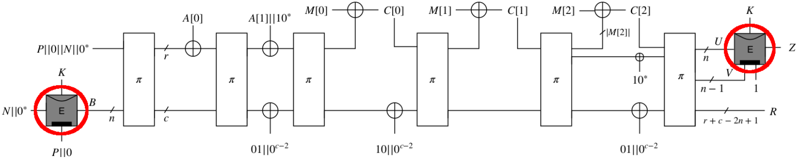
Proving masked implementations using composability

Gaëtan Cassiers   François-Xavier Standaert

July 4, 2019

How to make a gray box ?



Outline

Masking

Composition

Beyond the t-probing model

Outline

Masking

Composition

Beyond the t-probing model

t-probing model & masking

Probing model at order t:

The adversary observes t intermediate values.

t-probing model & masking

Probing model at order t:

The adversary observes t intermediate values.

Masking a sensitive bit x:

$$x = \underbrace{x_0 \oplus \cdots \oplus x_{d-2}}_{\text{random}} \oplus x_{d-1}$$

with $d = t + 1$.

Compute only on sharing $(x_0, \ldots, x_{d-1})$!

Computing with sharings: XOR

Operation:

$$z = x \oplus y$$

XOR gadget:

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} x_0 \oplus y_0 \\ x_1 \oplus y_1 \\ x_2 \oplus y_2 \end{pmatrix}$$

t-probing secure:

Each probe reveals at most one share of each input.

Computing on sharings: AND

Operation:

$$z = x \otimes y$$

AND gadget:

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} x_0 \otimes y_0 & \oplus & (x_0 \otimes y_1 \oplus r_0) & \oplus & (x_0 \otimes y_2 \oplus r_1) \\ (x_1 \otimes y_0 \oplus r_0) & \oplus & x_1 \otimes y_1 & \oplus & (x_1 \otimes y_2 \oplus r_2) \\ (x_2 \otimes y_0 \oplus r_1) & \oplus & (x_2 \otimes y_1 \oplus r_2) & \oplus & x_2 \otimes y_2 \end{pmatrix}$$

Requires randomness !

Outline

Composability flaw

Complex circuit: Computing on non-independent values.

Composability flaw

Complex circuit: Computing on non-independent values.

Trivial example:

$$z = x \otimes x$$

Composability flaw

Complex circuit: Computing on non-independent values.

Trivial example:

$$z = x \otimes x$$

Not 2-probing secure !

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} x_0 \otimes x_0 & \oplus & (x_0 \otimes x_1 \oplus r_0) & \oplus & (x_0 \otimes x_2 \oplus r_1) \\ (\mathbf{x_1} \otimes \mathbf{x_0} \oplus r_0) & \oplus & x_1 \otimes x_1 & \oplus & (x_1 \otimes x_2 \oplus r_2) \\ (\mathbf{x_2} \otimes x_0 \oplus r_1) & \oplus & (x_2 \otimes x_1 \oplus r_2) & \oplus & x_2 \otimes x_2 \end{pmatrix}$$

Proving probing security

Small gadgets:

- by hand (any order)
- automated exhaustive check (order-specific)

Proving probing security

Small gadgets:

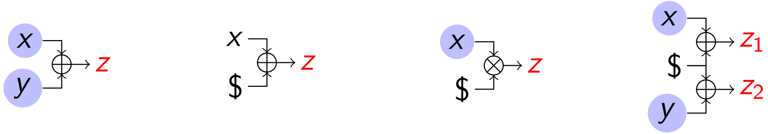
- by hand (any order)
- automated exhaustive check (order-specific)

Larger functionalities (S-box, block cipher):

- automated exhaustive check: often infeasible
- composable definitions:
  - more demanding at gadget level
  - general composition theorems

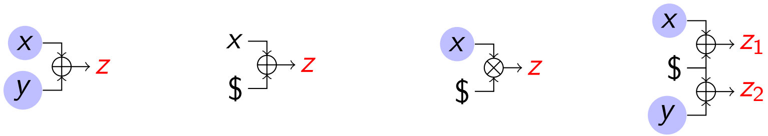
Simulatability

Inputs that are needed to simulate probes in presence of randomness $:



Simulatability

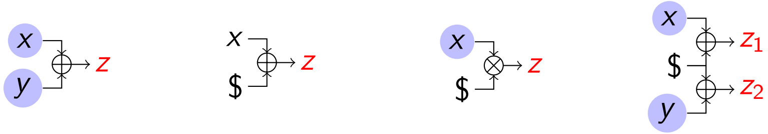
Inputs that are needed to simulate probes in presence of randomness \$:



Simulatability → Probe **propagation**

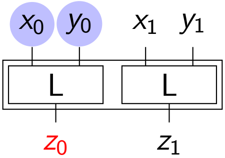
Simulatability

Inputs that are needed to simulate probes in presence of randomness \$:



Simulatability → Probe **propagation**

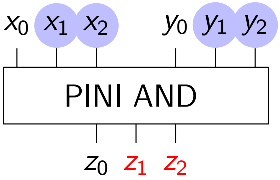
Linear gadgets: **share isolation**, easy composition.



Probe Isolating Non-Interference (PINI)

Share isolation **emulation**:

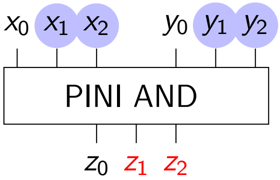
Gadgets should behave (w.r.t. simulatability) **as if shares were isolated.**



Probe Isolating Non-Interference (PINI)

Share isolation **emulation**:

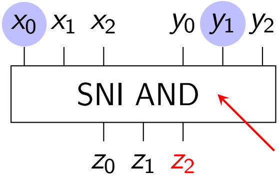
Gadgets should behave (w.r.t. simulatability) **as if shares were isolated.**

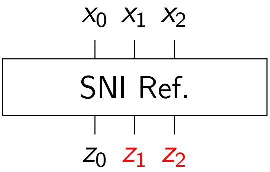


PINI AND gadget:

- hand-made
- **by composing SNI gadgets**

Strong Non-Interference (SNI)





Internal probes

→ 1 share of each input

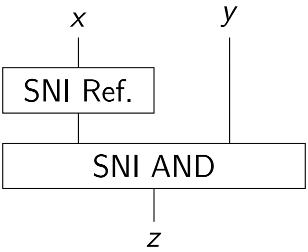
Output probes

→ no propagation

SNI Refresh:

- identity function
- blocks probe propagation

Composite PINI AND Gadget



That's all it takes for a composable masked circuit !

Implementation costs

	XOR	AND	Random
Refresh	$4d$	0	$2d$
SNI AND	$2d(d-1)$	d^2	$d(d-1)/2$
Clyde	23 808	1536	0
Msk Clyde	$3072d^2 + 26\,880d$	$1536d^2$	$768d^2 + 2304d$

Outline

Masking

Composition

Beyond the t-probing model

Hardware challenges

Some physical effects that are not captured by the t-probing model:

- Glitches: transient computations due to signal delays.
- Transitions: leakage from succession of values on the same wire.

Hardware challenges

Some physical effects that are not captured by the t-probing model:

- Glitches: transient computations due to signal delays.
- Transitions: leakage from succession of values on the same wire.

Improved model: **Robust probing model**:

- *Where to put registers to prevent harmful glitches ?*
- *Do I have problematic transitions ?*

Hardware challenges

Some physical effects that are not captured by the *t*-probing model:

- Glitches: transient computations due to signal delays.
- Transitions: leakage from succession of values on the same wire.

Improved model: **Robust probing model**:

- *Where to put registers to prevent harmful glitches ?*
- *Do I have problematic transitions ?*

Harder to model: couplings between wires, non-independence issues. . .

Left to hardware designers (?)

Horizontal attacks

Use all available information (compared to univariate attacks).

Horizontal attacks

Use all available information (compared to univariate attacks).

Masked multiplication: d uses of each input share.

- leakage increases with d
- critical for software implementations: high SNR

Horizontal attacks

Use all available information (compared to univariate attacks).

Masked multiplication: d uses of each input share.

- leakage increases with d
- critical for software implementations: high SNR

$\rightarrow$ Multiplication gadget with improved protection (cost x2).

Tools

Gadget-level

- order-specific
- check all sets of probes
- computationally expensive
  - refresh: $d \leq 16$
  - multiplication: $d \leq 7$

Tools

Gadget-level

- order-specific
- check all sets of probes
- computationally expensive
  - refresh: $d \leq 16$
  - multiplication: $d \leq 7$

Cipher-level

- *Are all gadgets PINI ?*
- Other issues (HW implementations):
  - mixing valid & invalid data
  - shuffling wires ?
  - randomness timing
  - ...

Conclusion

- Use a provably secure masking scheme.
- PINI: one technique, proven security.
  - Still a lot of freedom for performance trade-offs.
- t-probing model: a first step, but not sufficient.

Thank you!