CryptoExperts

WE INNOVATE TO SECURE YOUR BUSINESS

The relevance and challenges of random probing security for post-quantum algorithms

Application to Raccoon Signature Scheme

Mélissa Rossi

WRACH, Roscoff, 23/04/2025

Based on the paper *New Techniques for Random Probing Security and Application to Raccoon Signature Scheme*
S. Belaïd, M. Rivain and M. Rossi, published in Eurocrypt 2025, https://eprint.iacr.org/2025/278

1) The relevance of the random probing model

2) Scaling up is a big challenge

3) Random-probing Raccoon

4) Ideas for future works

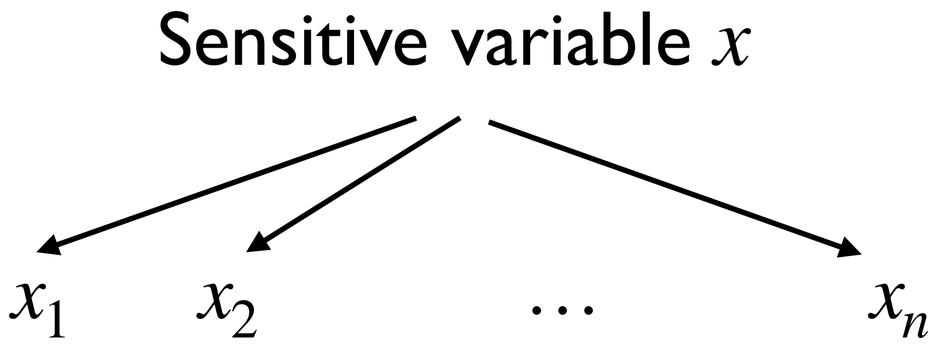
1) The relevance of the random probing model

2) Scaling up is a big challenge

3) Random-probing Raccoon

4) Ideas for future works

Masking



Each strict subset of $(x_i)_{1 \le i \le n}$ is independent from x

$$x_1, x_2, \ldots, x_{n-1} \leftarrow \$$$

$$x_n \leftarrow x - (x_1 + x_2 + \ldots + x_{n-1})$$

Masking in Practice

- Masking linear operations

$$z \leftarrow x \oplus y$$

$$x = x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1}$$
$$y = y_0 \oplus y_1 \oplus \ldots \oplus y_{n-1}$$

$$\mathbf{z} = (x_0 \oplus y_0, x_1 \oplus y_1, \ldots, x_{n-1} \oplus y_{n-1})$$

- Masking non linear operations
  - Cannot be done share by share
  - Example of multiplication for $n = 2$

$$x = x_0 \oplus x_1$$
$$y = y_0 \oplus y_1$$

$$z \leftarrow x_0y_0 \oplus x_0y_1 \oplus x_1y_0 \oplus x_1y_1$$

$$z_0 \leftarrow x_0y_0 \oplus x_0y_1$$
$$z_1 \leftarrow x_1y_1 \oplus x_1y_0$$

From a gadget to a circuit

x_1 x_2 k_1 k_2

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

From a gadget to a circuit

x_1 x_2 k_1 k_2

r

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$r \leftarrow \$$

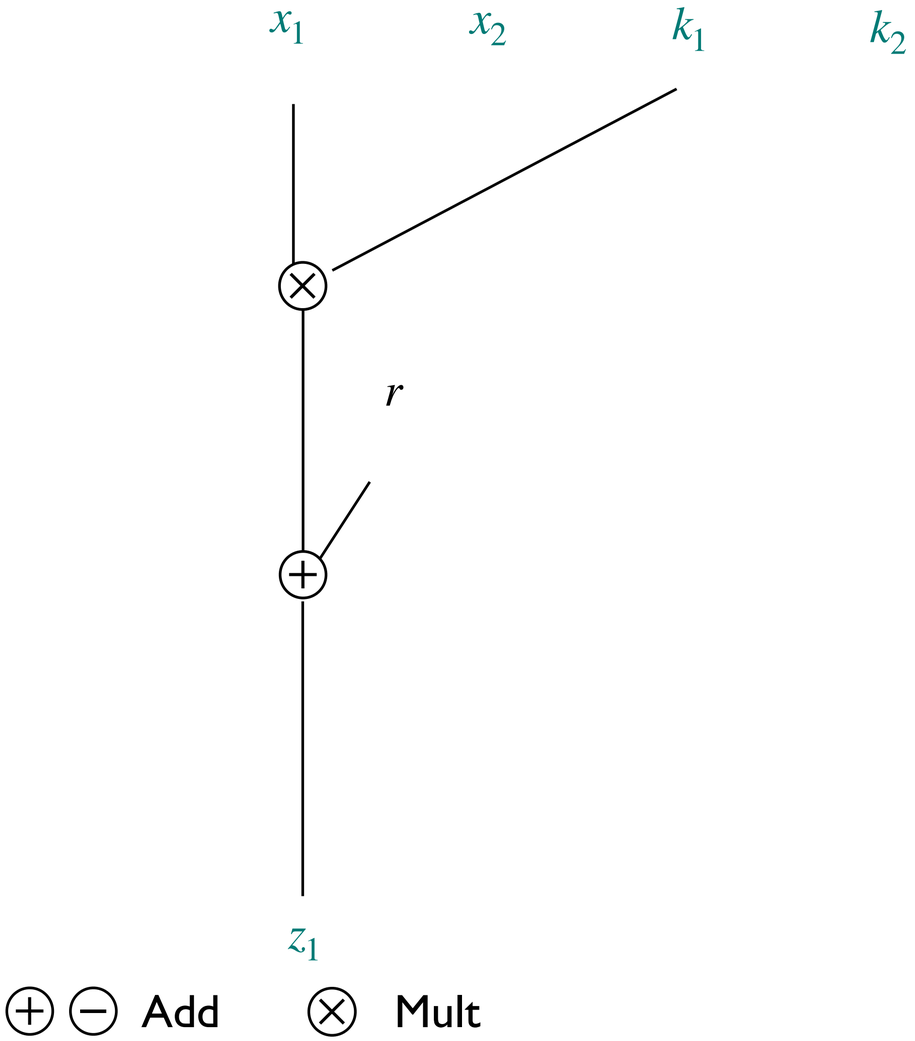
$z_1 \leftarrow x_1 k_1 + r$

$r' \leftarrow x_1 k_2 - r$

$r'' \leftarrow r' + x_2 k_1$

$z_2 \leftarrow r'' + x_2 k_2$

From a gadget to a circuit



A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$r \leftarrow \$$

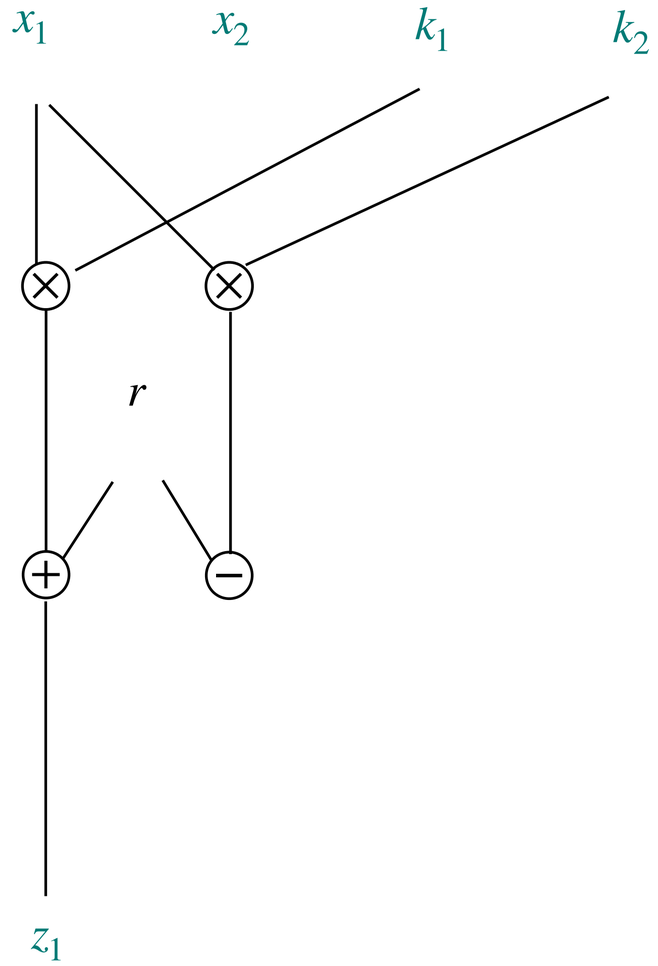
$z_1 \leftarrow x_1 k_1 + r$

$r' \leftarrow x_1 k_2 - r$

$r'' \leftarrow r' + x_2 k_1$

$z_2 \leftarrow r'' + x_2 k_2$

From a gadget to a circuit



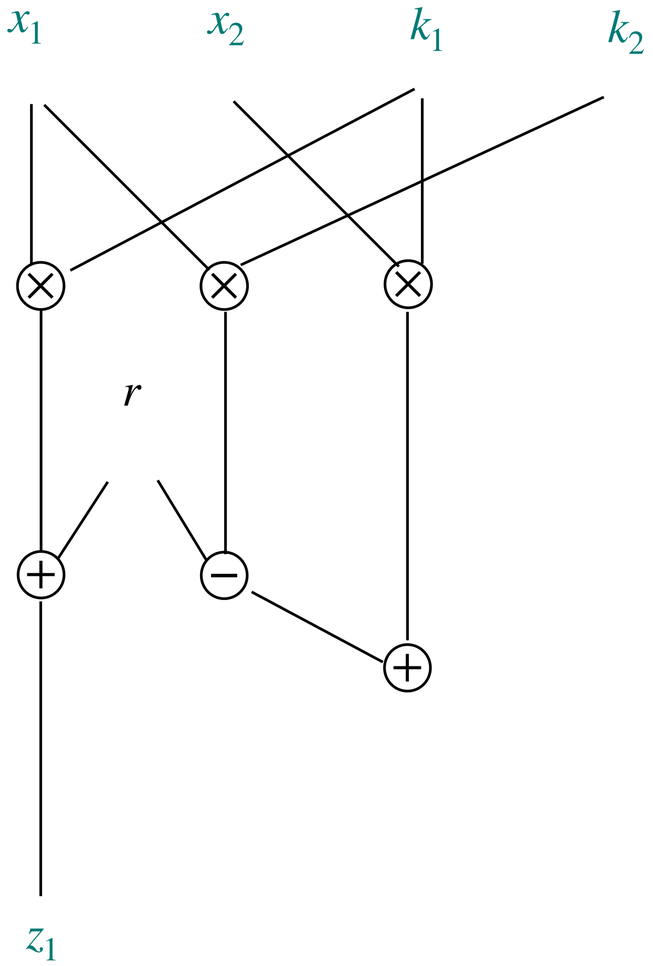
⊕ ⊖ Add ⊗ Mult

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

From a gadget to a circuit



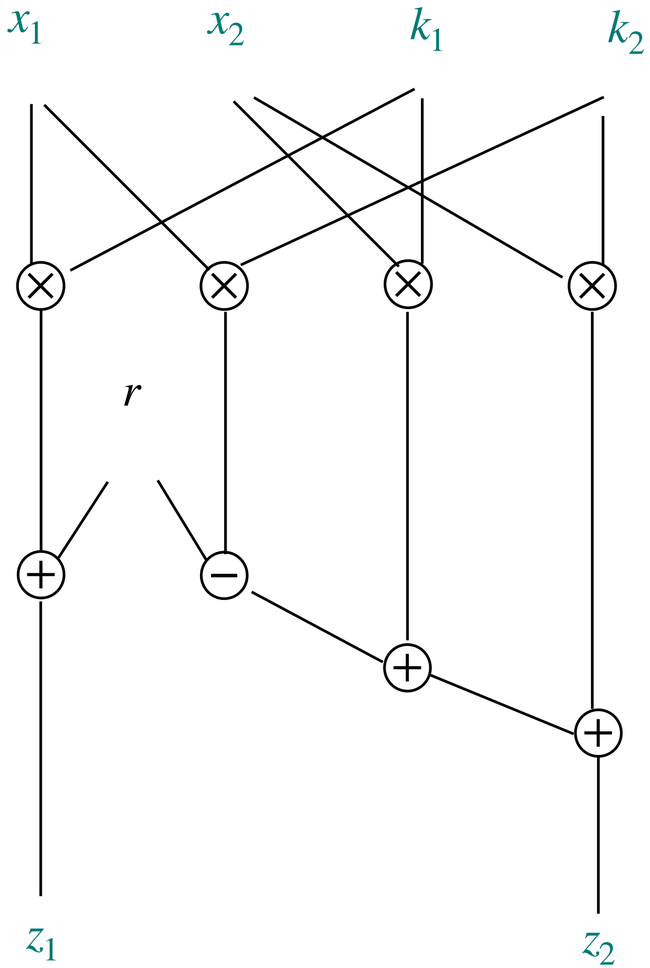
⊕ ⊖ Add ⊗ Mult

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

From a gadget to a circuit



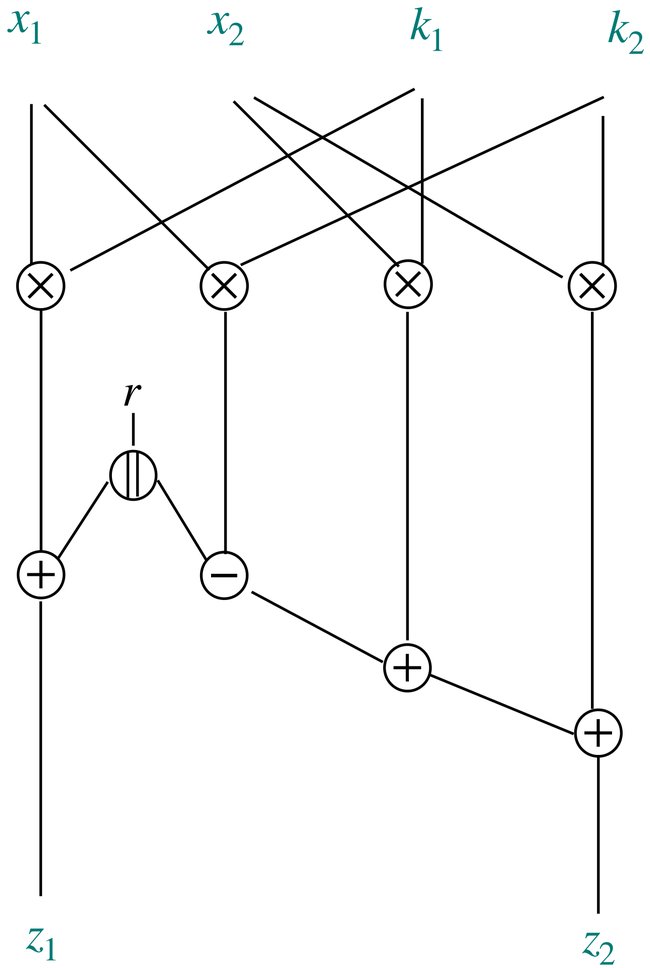
⊕ ⊖ Add ⊗ Mult

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

From a gadget to a circuit



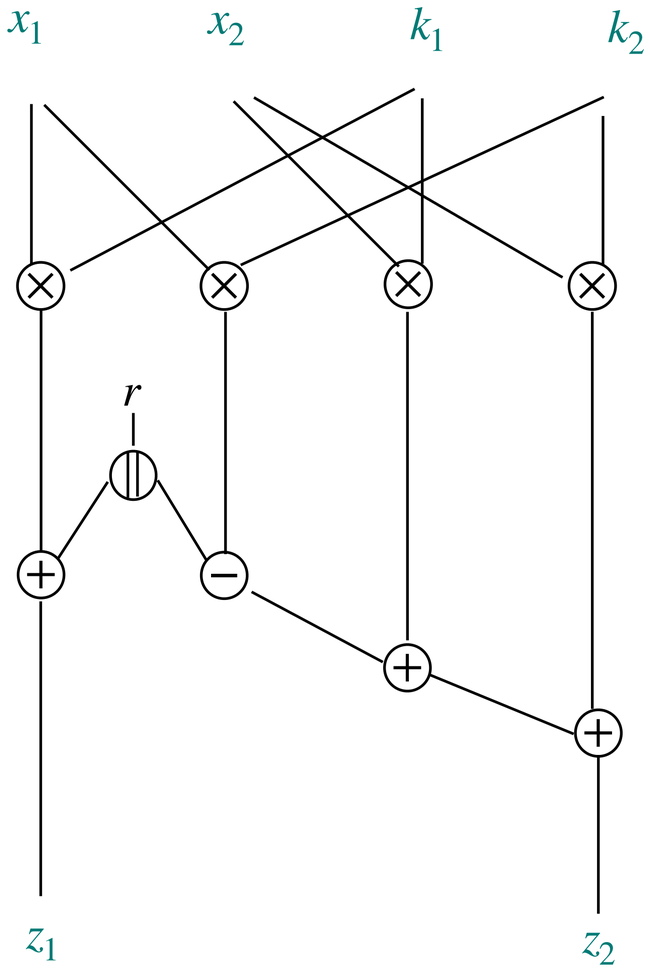
⊕ ⊖ Add ⊗ Mult ⫼ Copy

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

From a gadget to a circuit



⊕ ⊖ Add ⊗ Mult ⫴ Copy

A Multiplication gadget

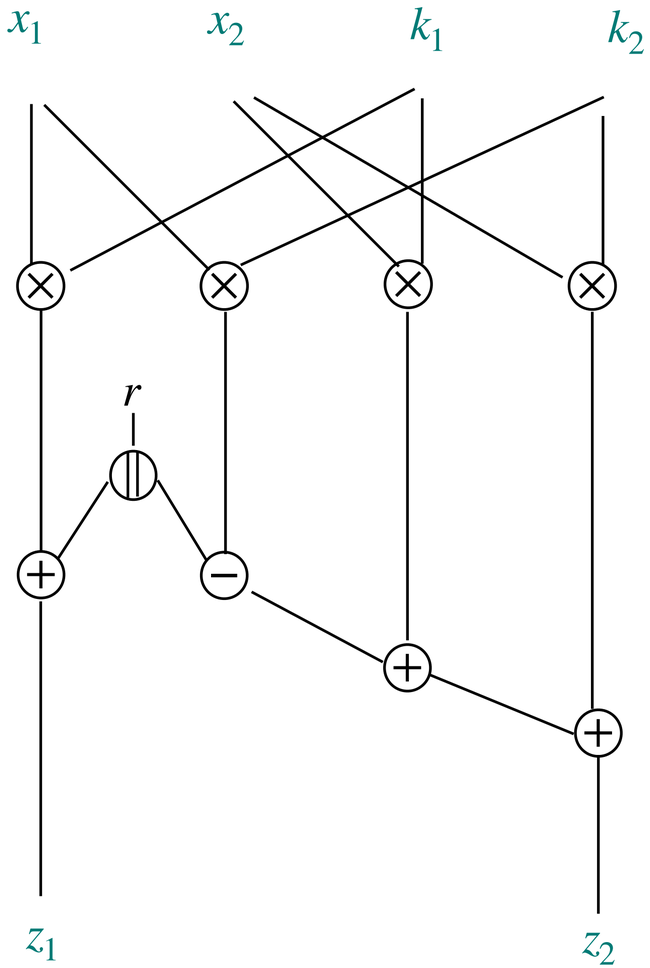
$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

Attacker model

The attacker is given the design of the circuit and some extra information about the wires. He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

From a gadget to a circuit



⊕ ⊖ Add ⊗ Mult ◫ Copy

A Multiplication gadget

$$z_1 + z_2 = (x_1 + x_2) \cdot (k_1 + k_2)$$

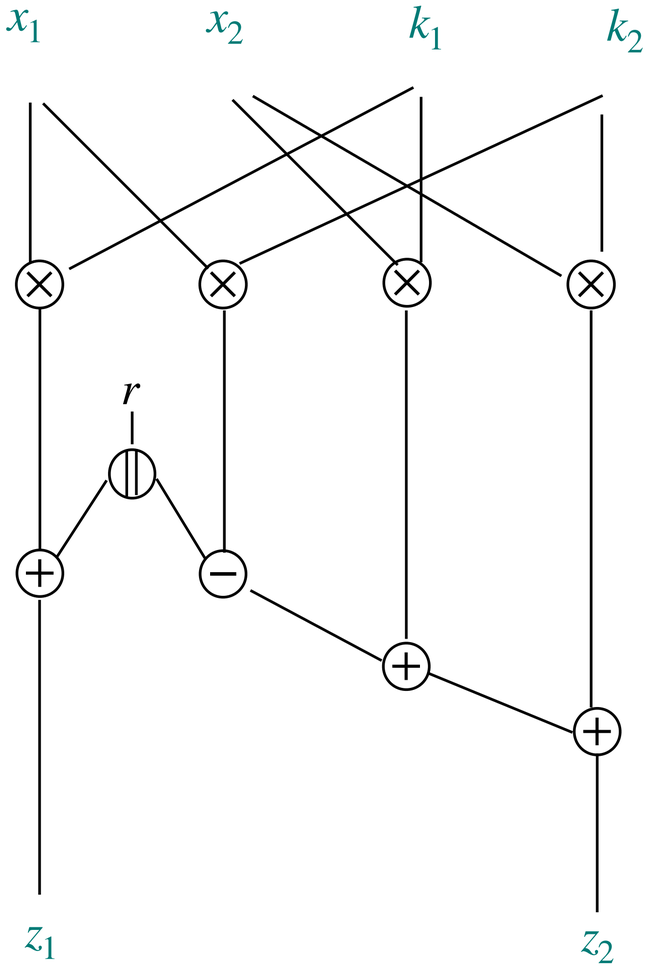
$$r \leftarrow \$$$
$$z_1 \leftarrow x_1 k_1 + r$$
$$r' \leftarrow x_1 k_2 - r$$
$$r'' \leftarrow r' + x_2 k_1$$
$$z_2 \leftarrow r'' + x_2 k_2$$

Attacker model

The attacker is given the design of the circuit and some extra information about the wires. He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

3 flavours
1. t-probing model
2. Random probing model
3. Noisy leakage model

t-Probing model

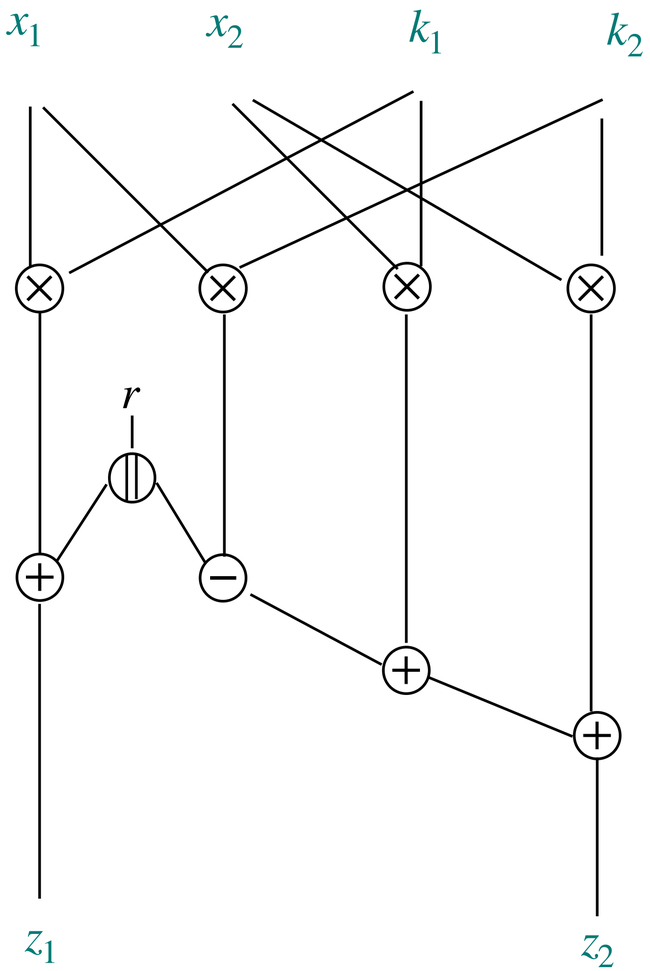


Attacker model

The attacker can ask for the value of at most t wires.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing model



Attacker model

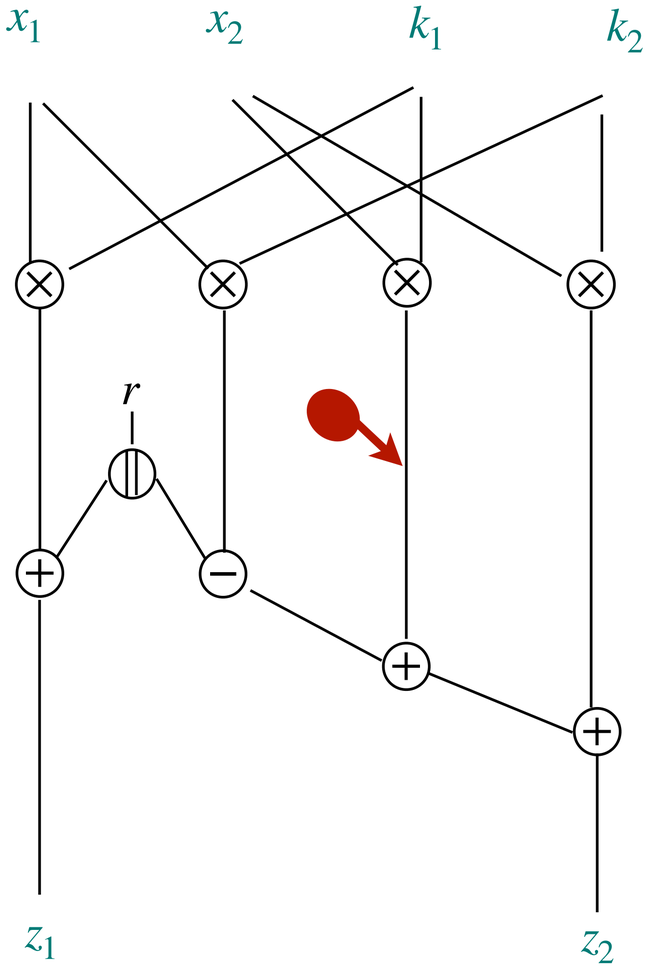
The attacker can ask for the value of at most t wires.

t-probing security

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
The distribution of $\mathscr{P}$ is independent from the secrets $\sum x_i$ and $\sum k_i$.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing model



Attacker model

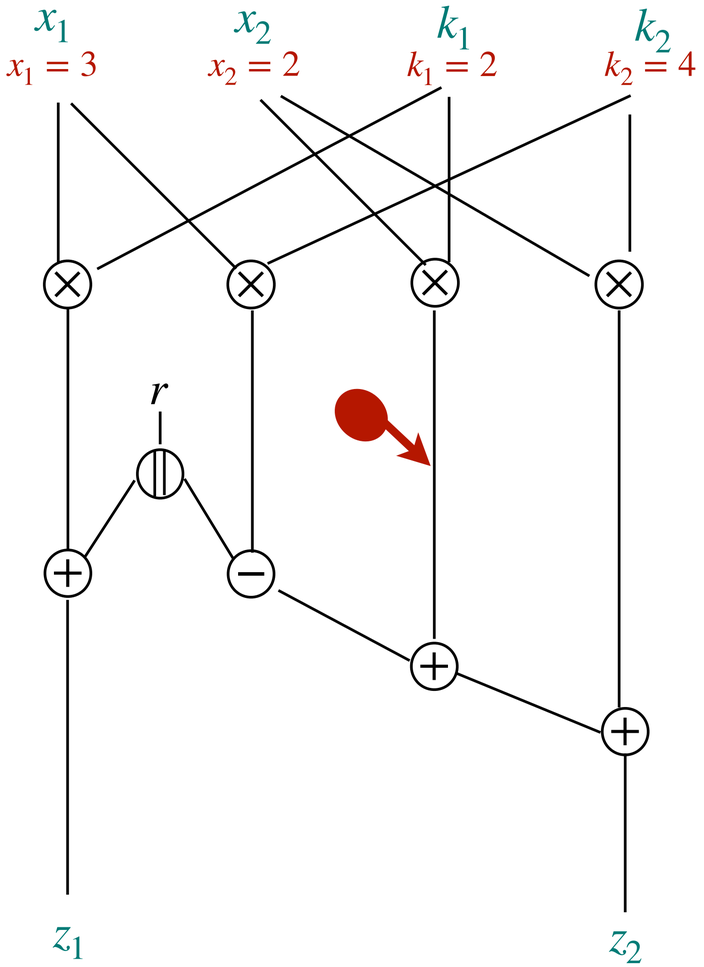
The attacker can ask for the value of at most t wires.

t-probing security

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
The distribution of $\mathscr{P}$ is independent from the secrets $\sum x_i$ and $\sum k_i$.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing model



Attacker model

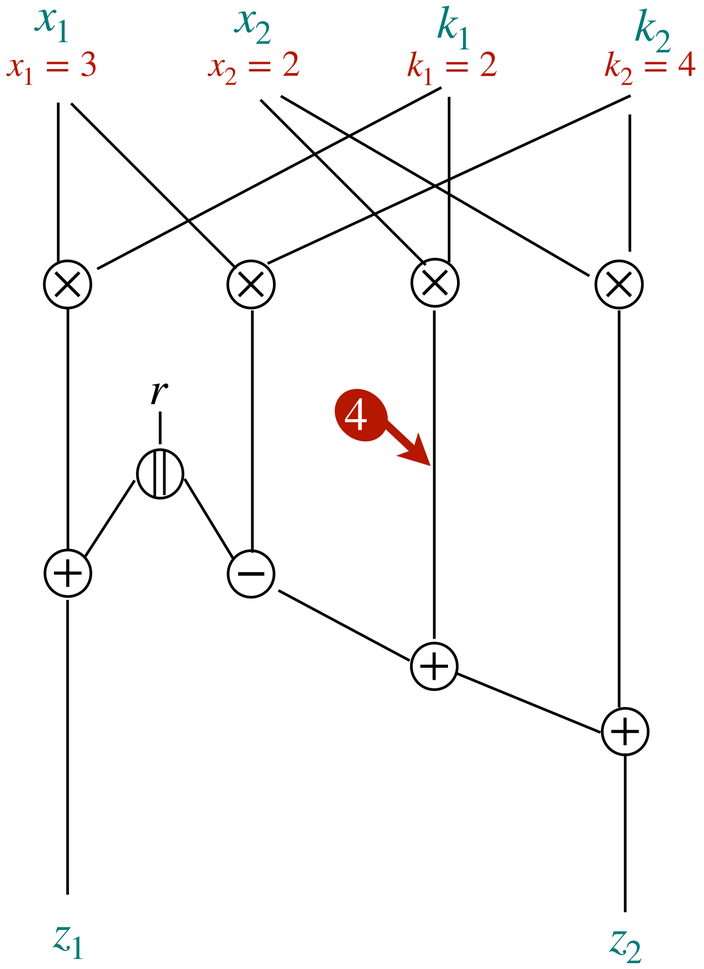
The attacker can ask for the value of at most t wires.

t-probing security

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
The distribution of $\mathscr{P}$ is independent from the secrets $\sum x_i$ and $\sum k_i$.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing model



Attacker model

The attacker can ask for the value of at most t wires.

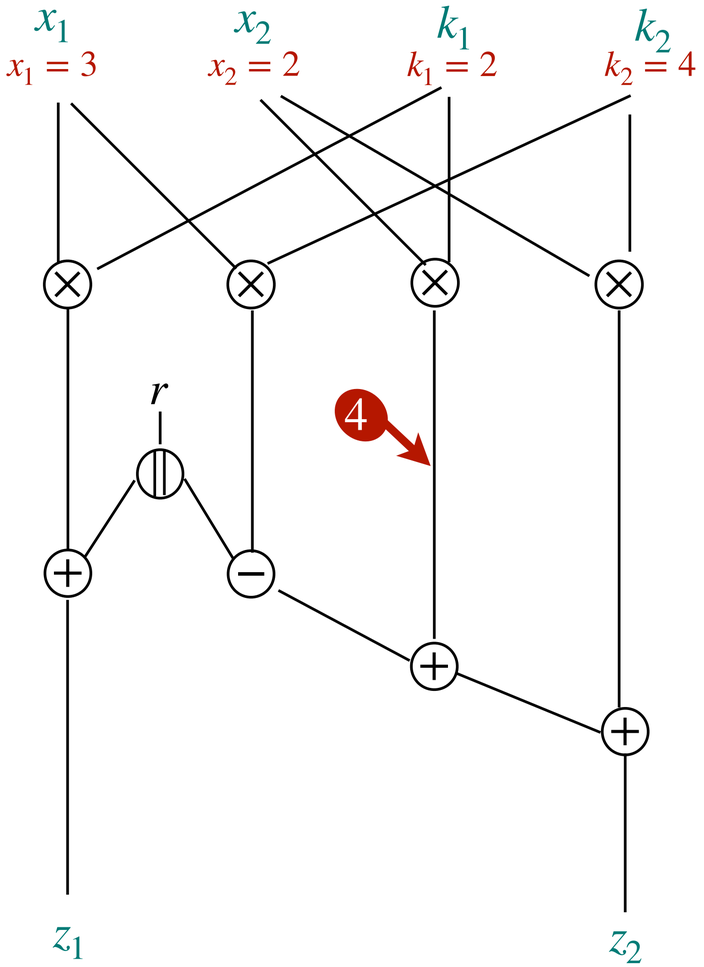
t-probing security

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
The distribution of $\mathscr{P}$ is independent from the secrets $\sum x_i$ and $\sum k_i$.

$\mathscr{L} = 4$

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing model



Attacker model

The attacker can ask for the value of at most t wires.

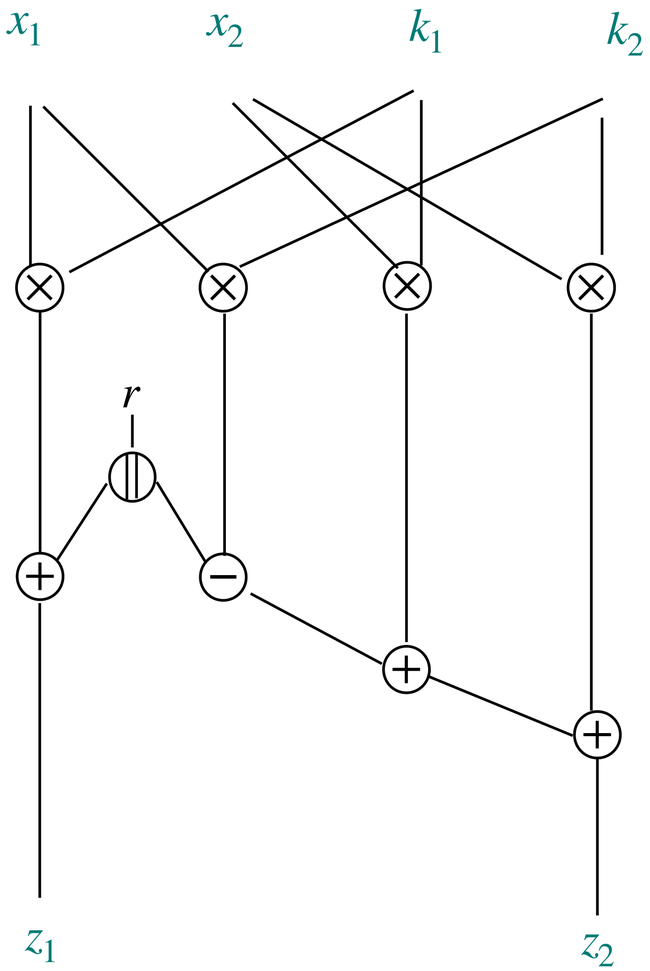
t-probing security

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
The distribution of $\mathscr{P}$ is independent from the secrets $\sum x_i$ and $\sum k_i$.

$\mathscr{L} = 4$
Knowing $\mathscr{L} = 4$ the probability of $(x = 5, \quad k = 6)$ is the same as for any other value.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003,

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

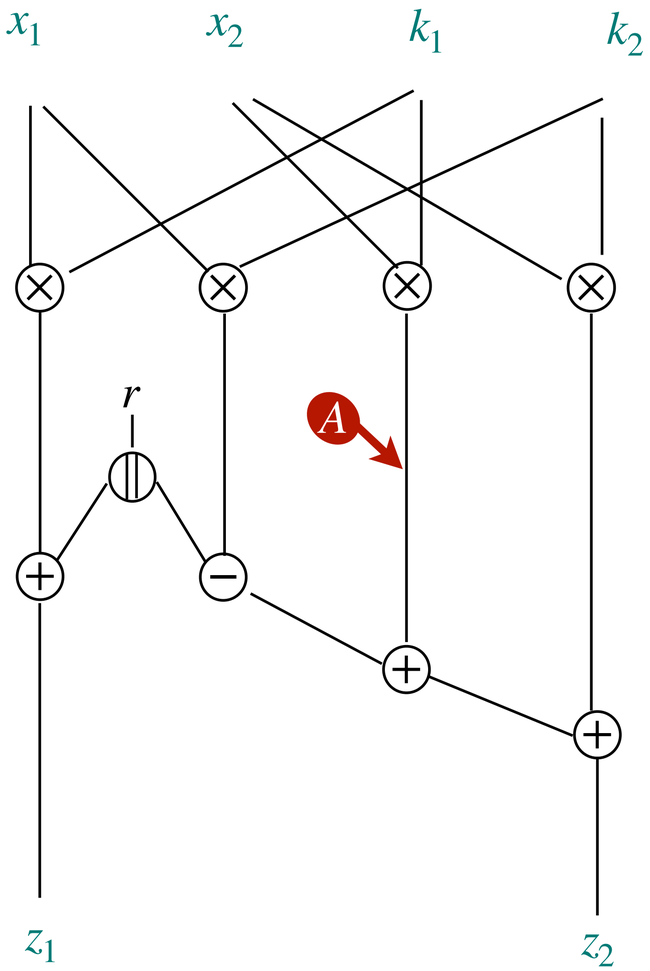
There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

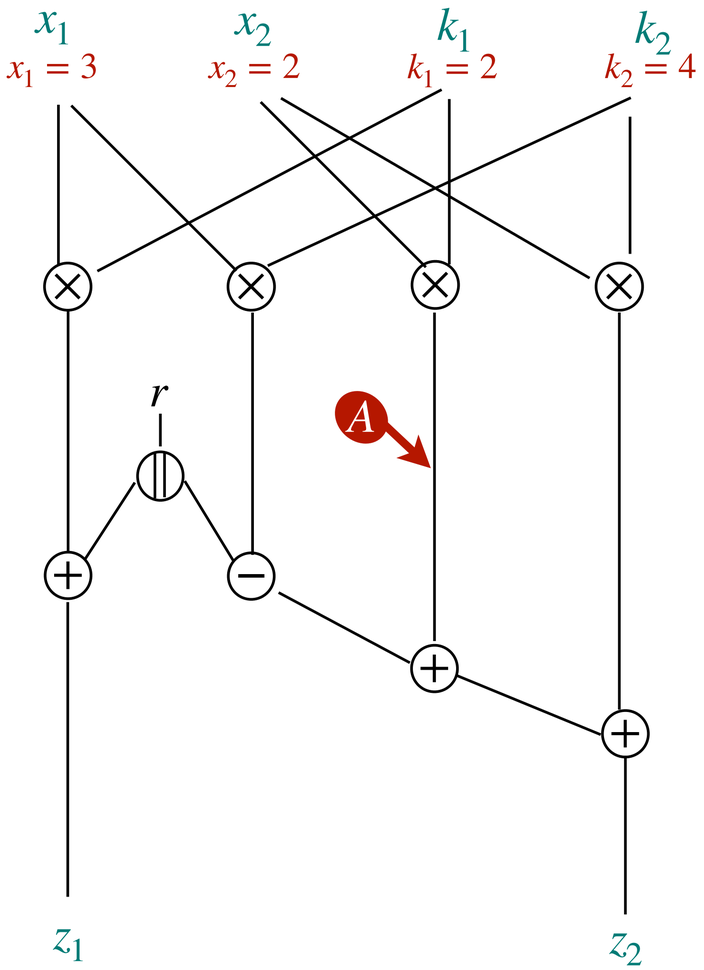
There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini.
Strong non-interference and type-directed higher-order masking. ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

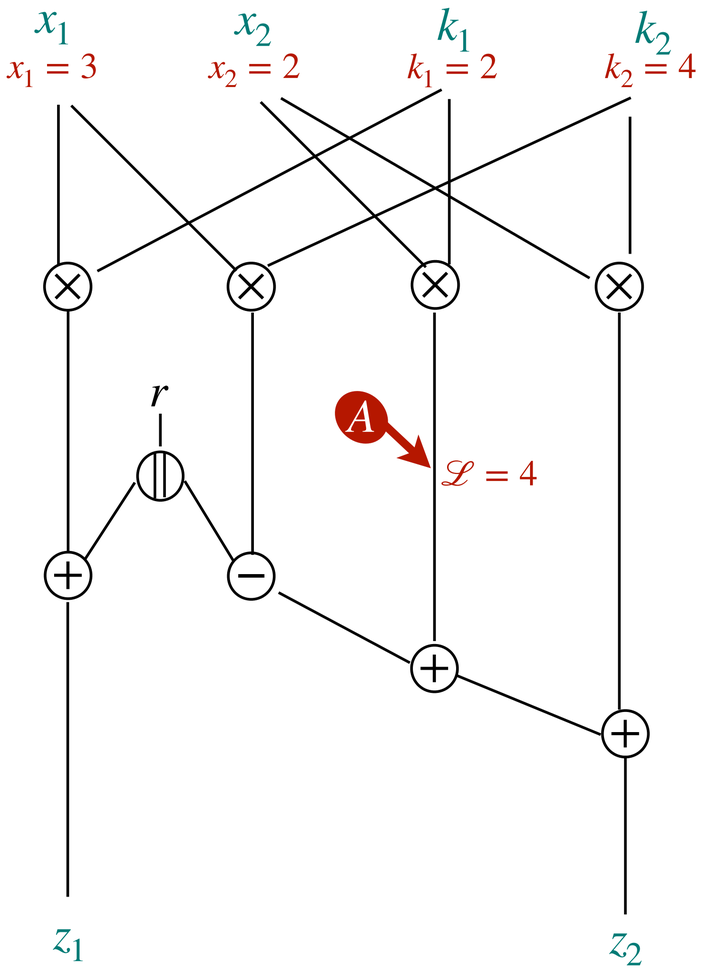
There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

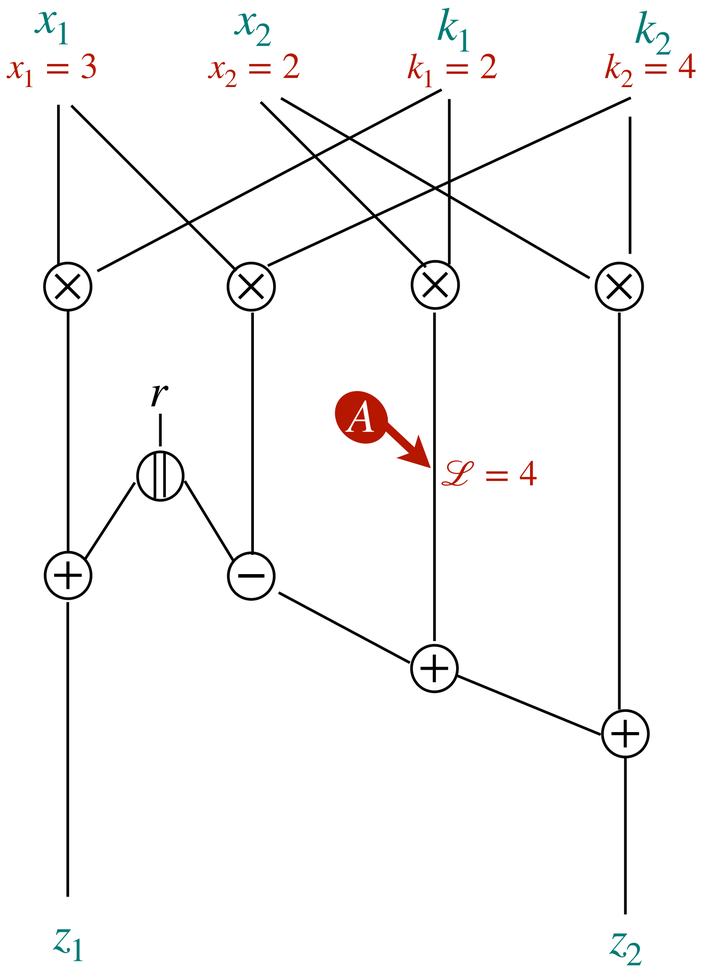
There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini.
Strong non-interference and type-directed higher-order masking. ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

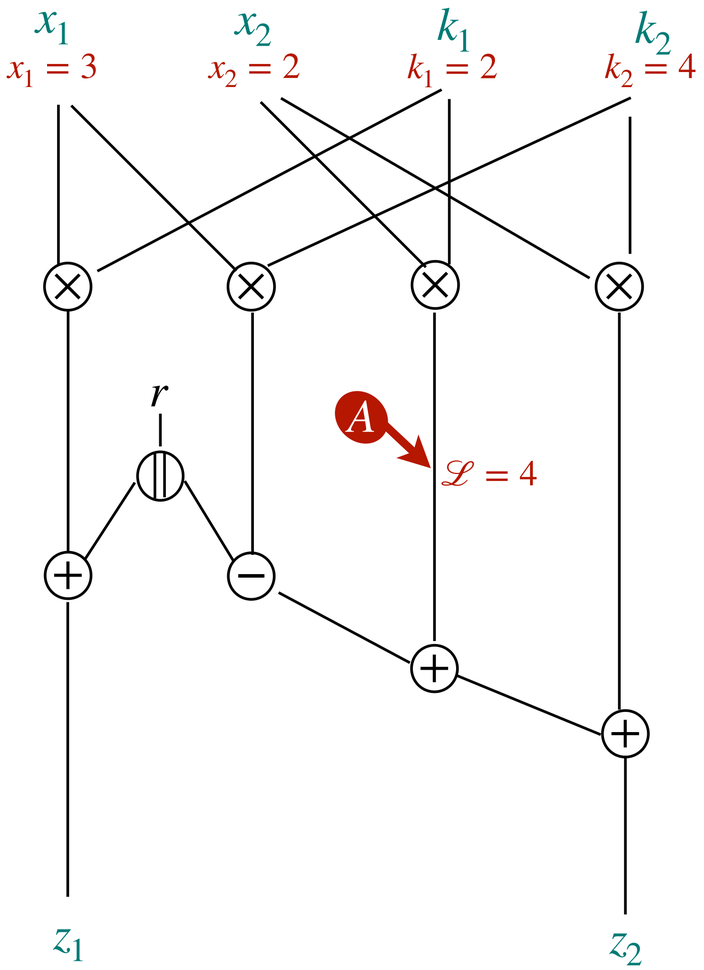
$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1$ $I_x = 2$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

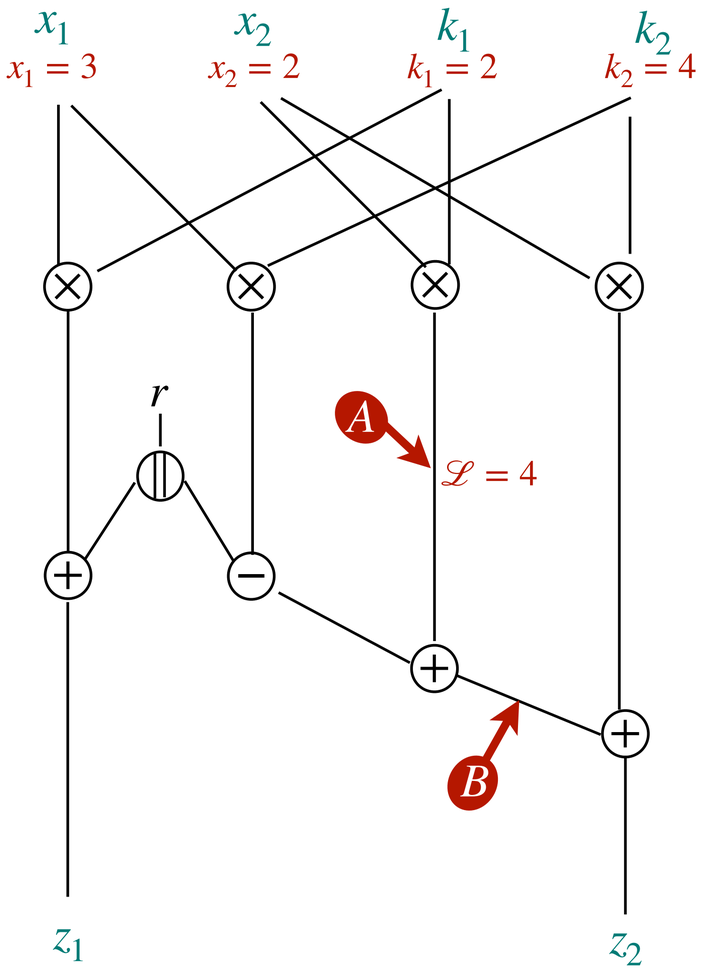
$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

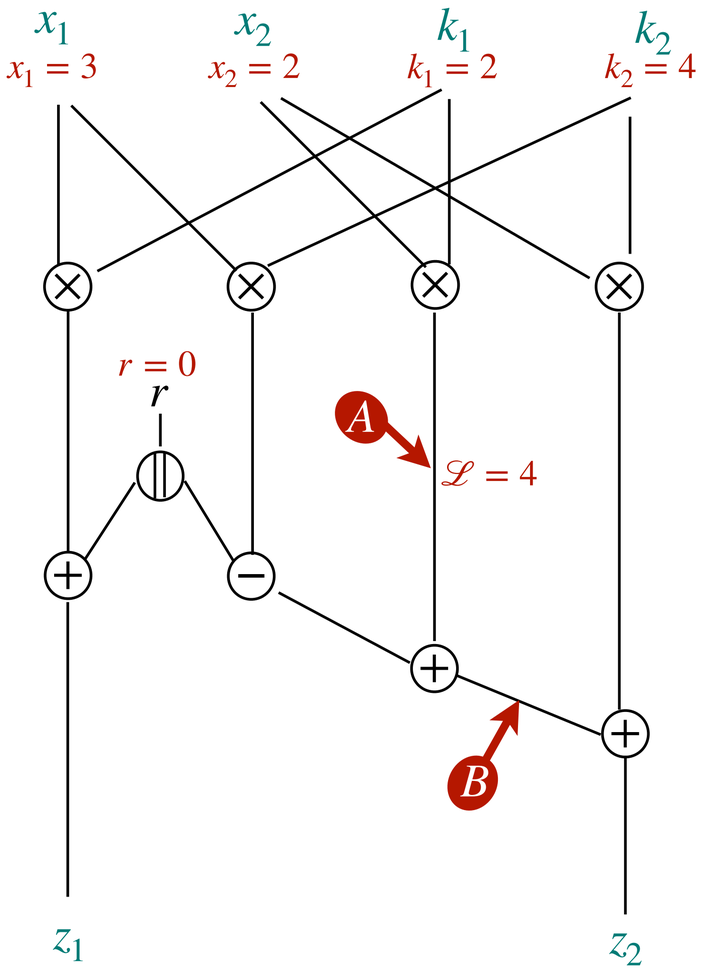
$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

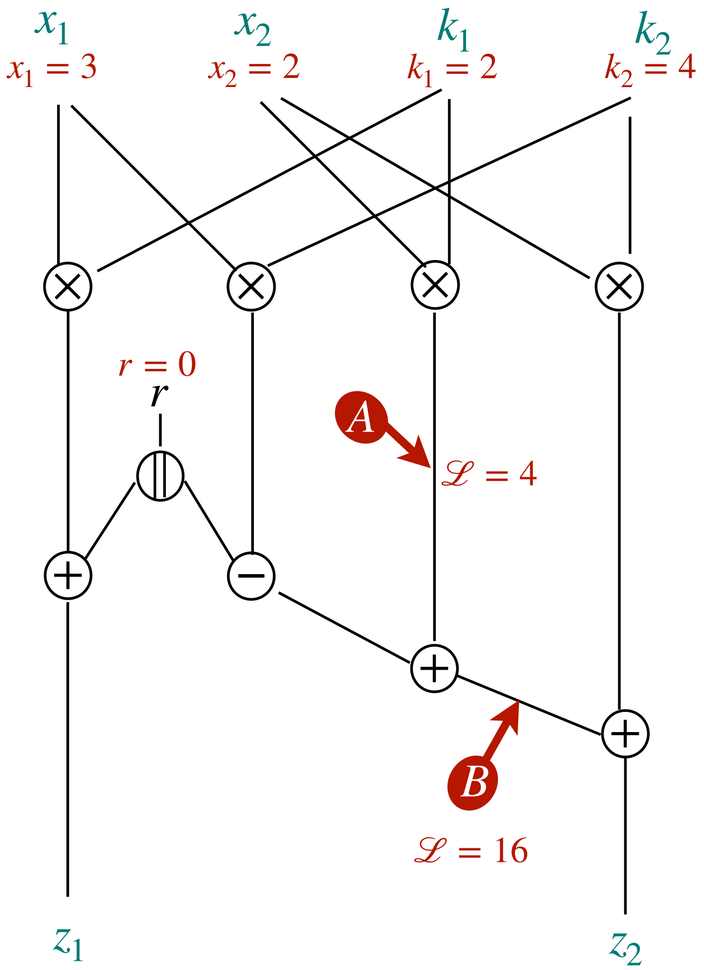
$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

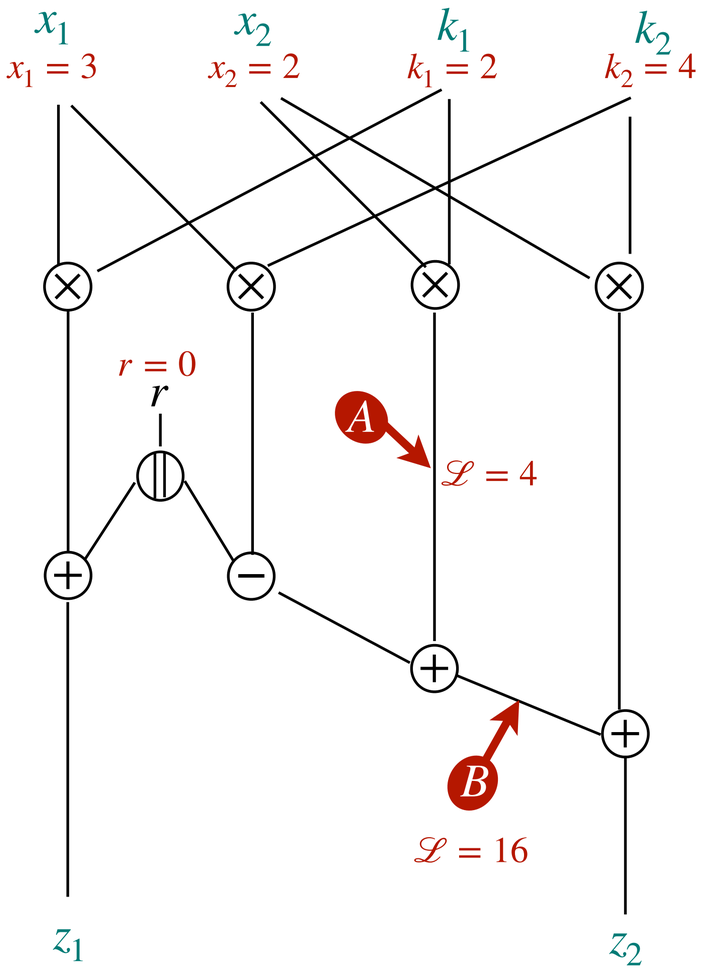
$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

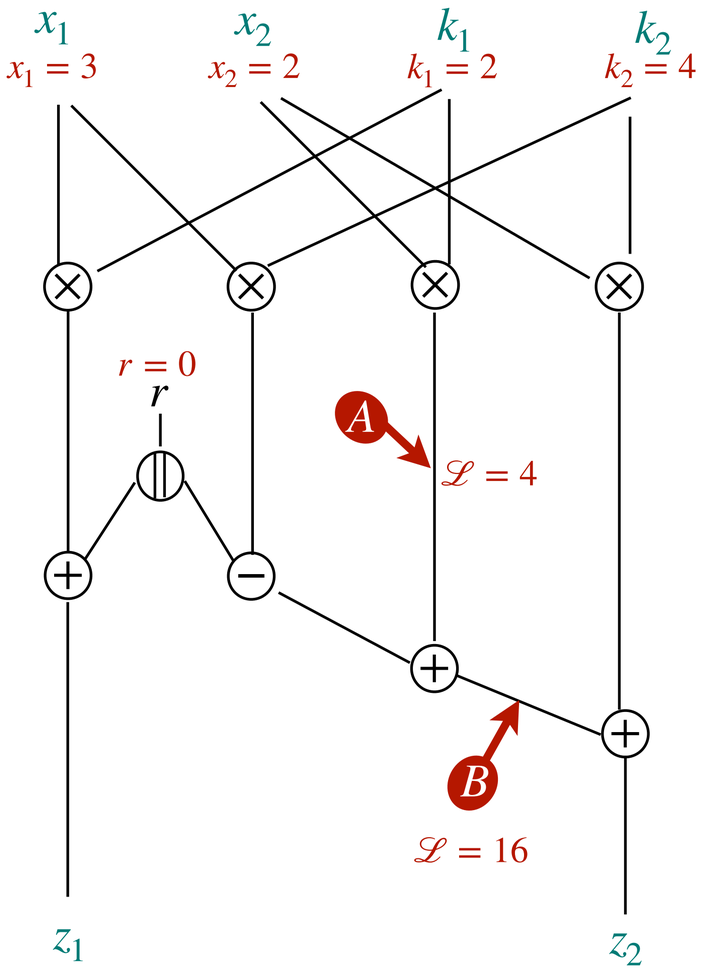
A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

B $I_x = \varnothing \quad I_k = \varnothing$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \text{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \text{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \text{SimOut}(x_{|I_x}, k_{|I_k})$$

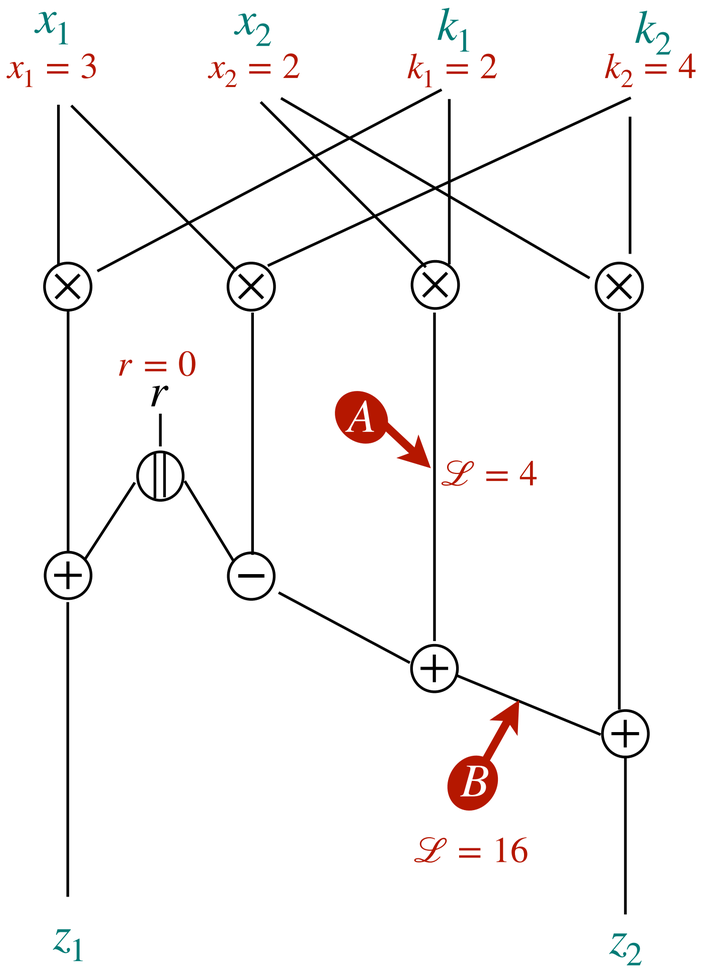
$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A: $I_k = 1 \quad I_x = 2$
$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

B: $I_x = \varnothing \quad I_k = \varnothing$
$\text{out} \leftarrow \$$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability



Composability : t-Non Interference

Let $\mathscr{P}$ be a set of at most t probes :

Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be encodings of inputs x and k.

Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{P}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{P}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\text{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\text{perfect}}{\approx} \mathscr{L}$$

A $I_k = 1 \quad I_x = 2$

$\text{out} = x_{|I_x} \cdot k_{|I_k} = x_2 k_1 = 4$

B $I_x = \varnothing \quad I_k = \varnothing$

$\text{out} \leftarrow \$$

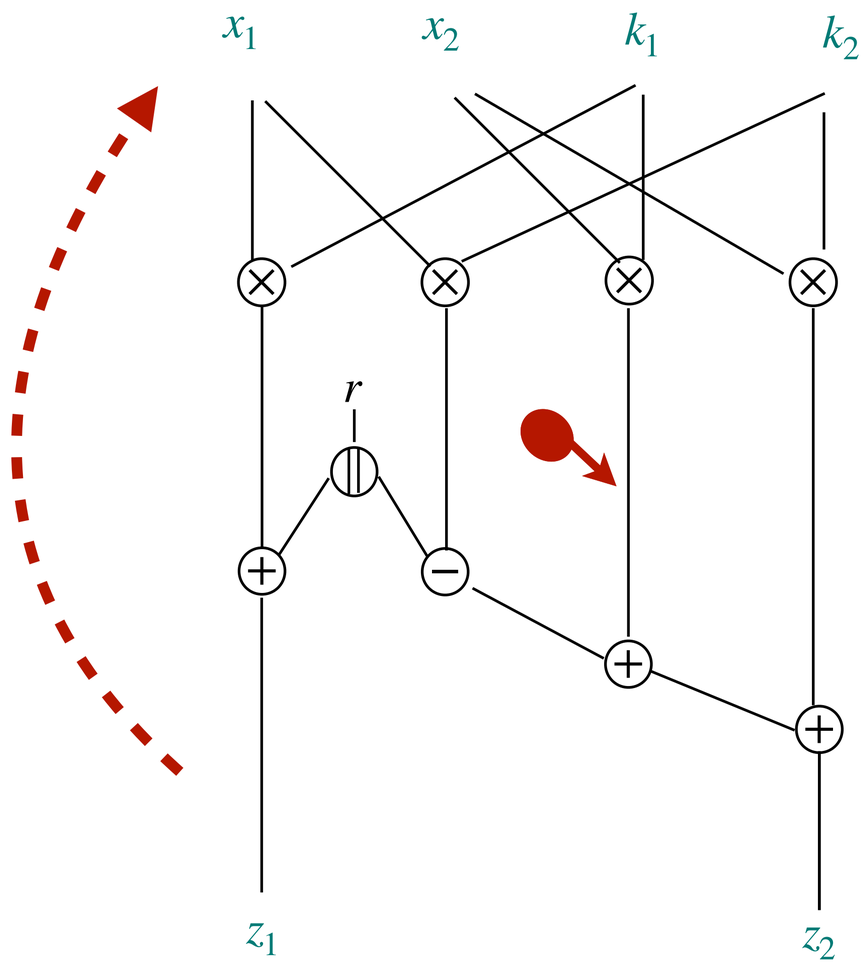
Any out is valid as it corresponds to drawing $r = \text{out} - x_2 k_1 - x_1 k_2$

[BBD+16] G. Barthe, S. Belaïd, F. Dupressoir, P.-A. Fouque, B. Grégoire, P.-Y. Strub, and R. Zucchini. *Strong non-interference and type-directed higher-order masking.* ACM CCS 2016

t-Probing Security & composability

<u>Non Interference:</u>

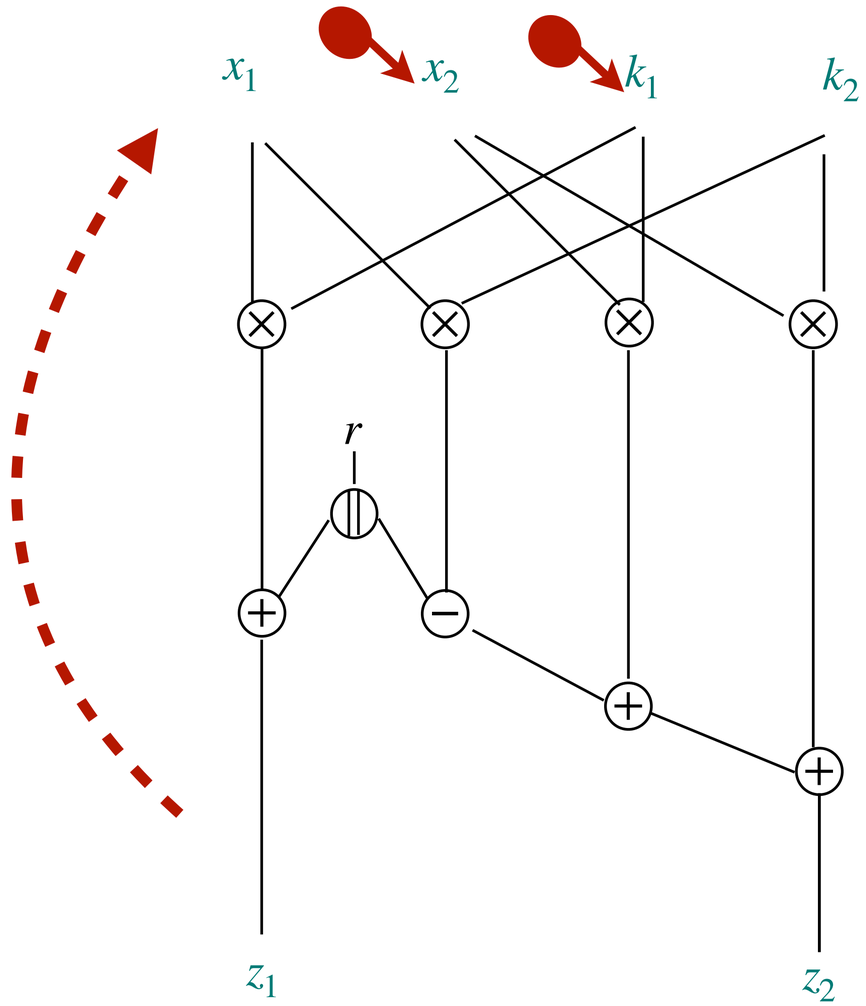
Propagation of the simulation from inside to gadget to its inputs



t-Probing Security & composability

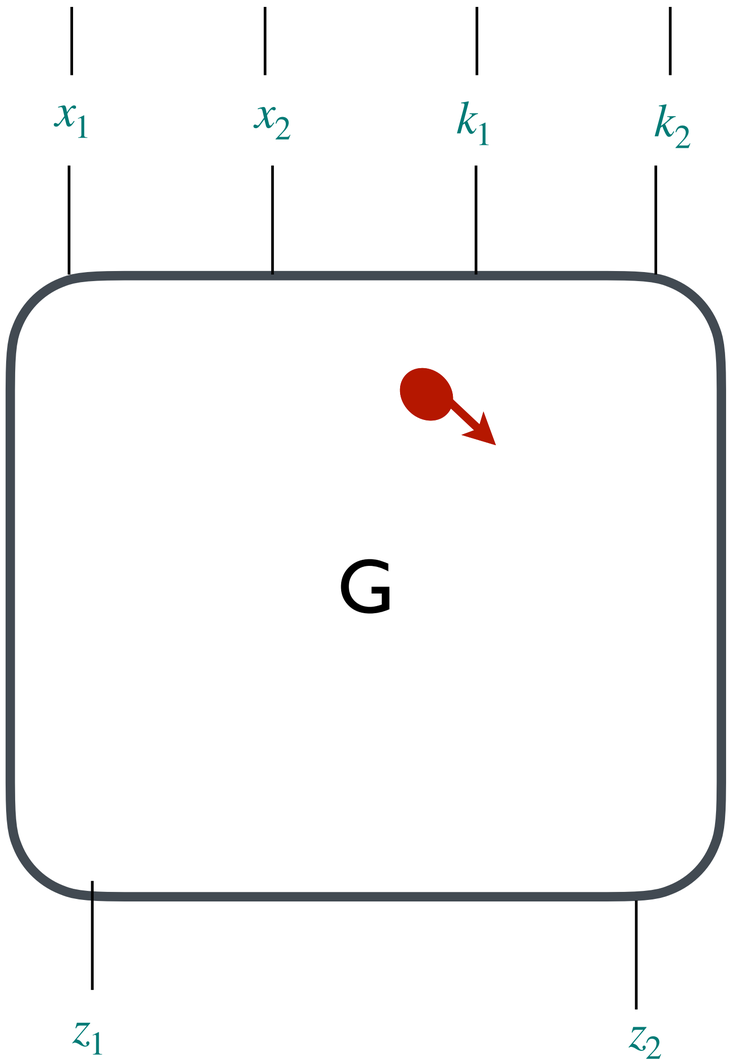
<u>Non Interference:</u>

Propagation of the simulation from inside to gadget to its inputs



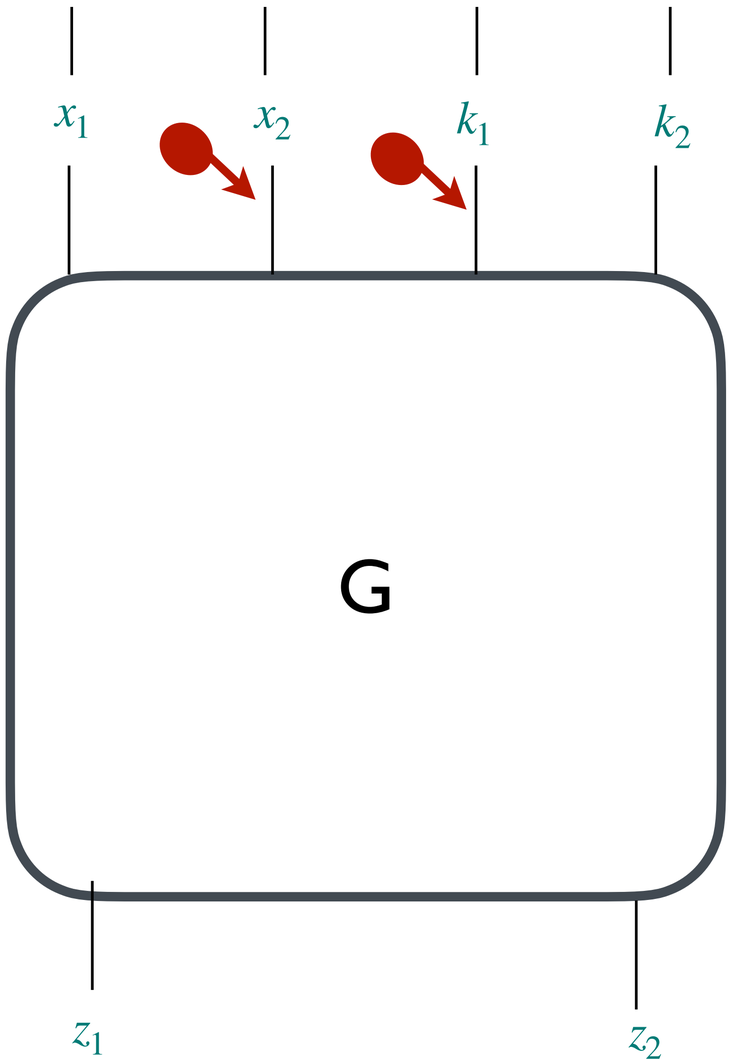
t-Probing Security & composability

Connecting gadgets in a sequential way



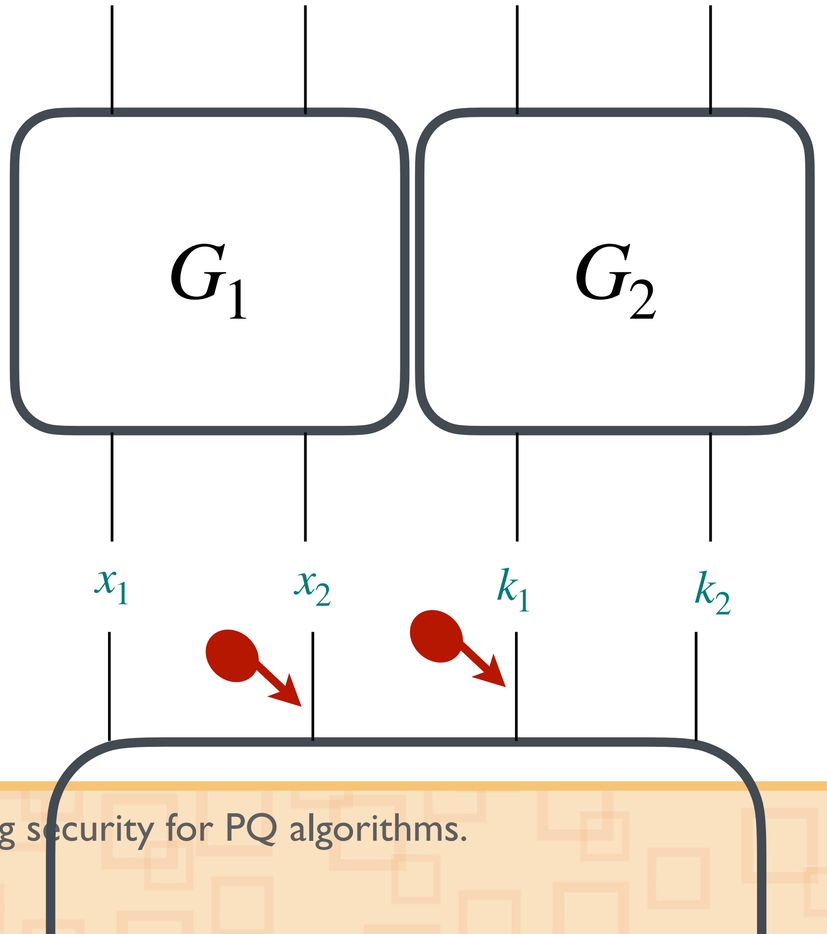
t-Probing Security & composability

Connecting gadgets in a sequential way



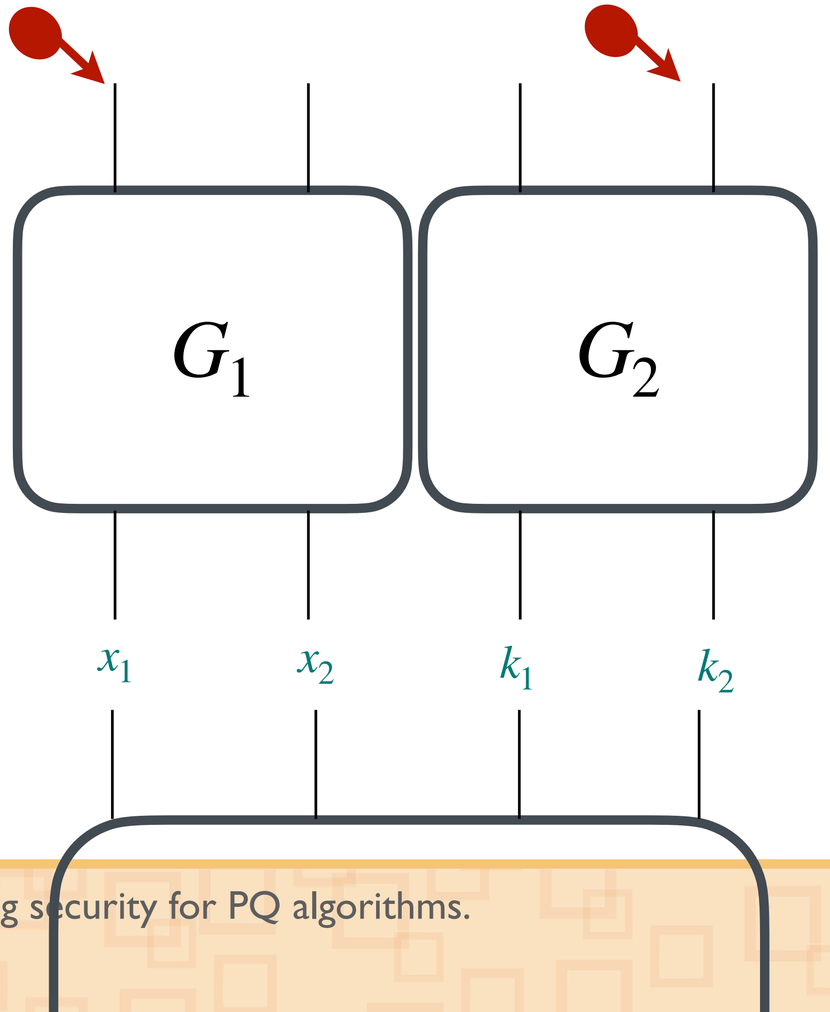
t-Probing Security & composability

Connecting gadgets in a sequential way



t-Probing Security & composability

Connecting gadgets in a sequential way



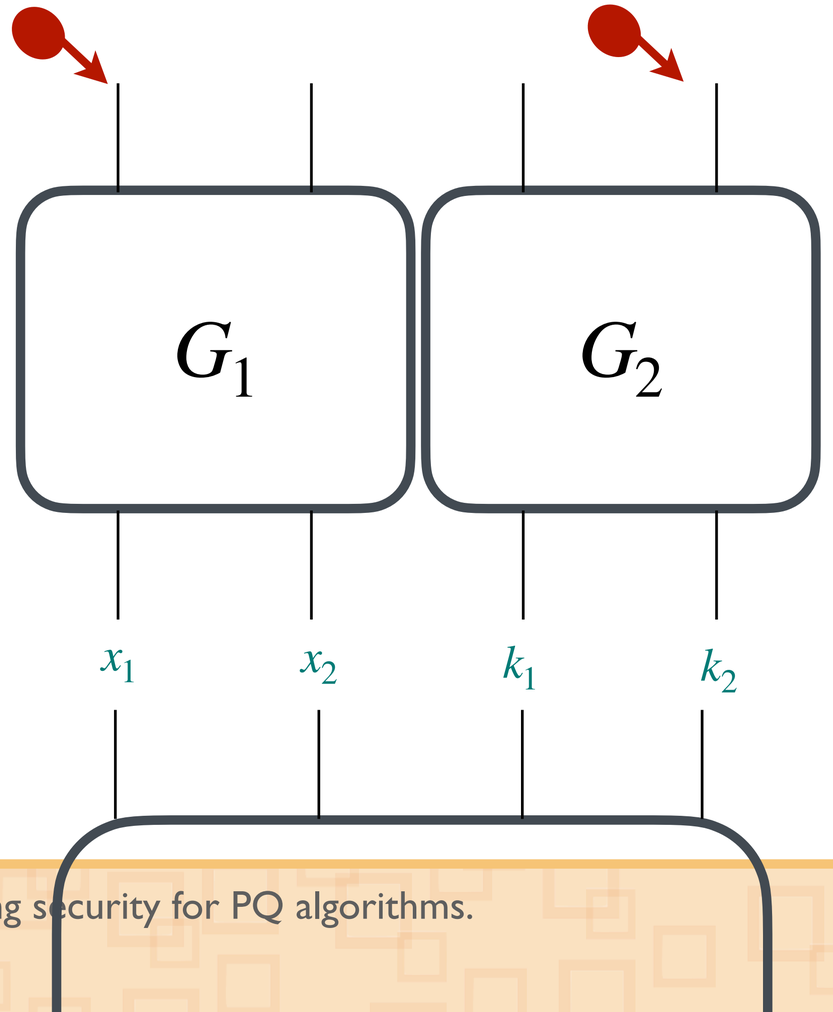
t-Probing Security & composability

Connecting gadgets in a sequential way

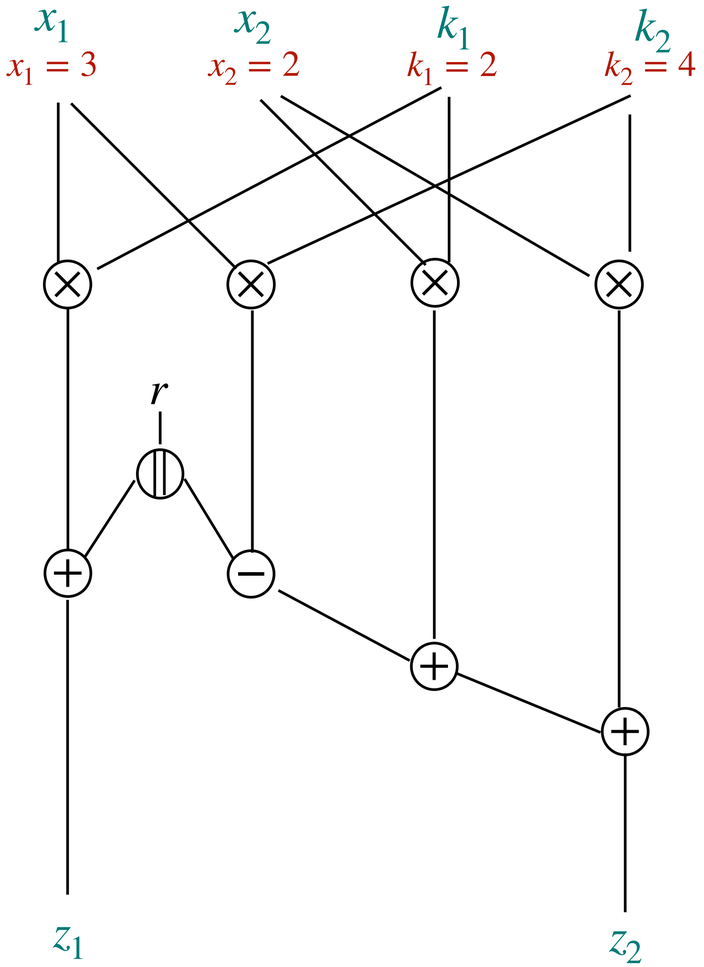
Composition

A sequential composition of t-Non interferent gadgets is also t-Non interferent.
$\Longrightarrow$ t-probing secure with a uniform input encoding.

There is a way to handle non sequential compositions by using refreshes to break dependencies.



Random Probing Security

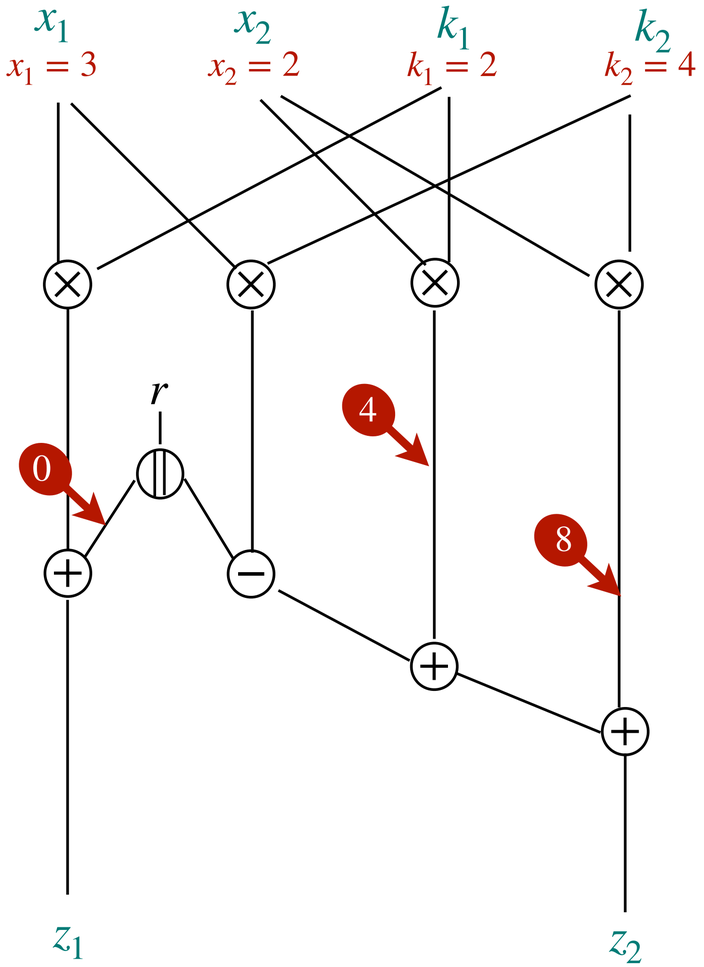


Attacker model

The attacker is given the value of each wire with probability p.

[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

Random Probing Security

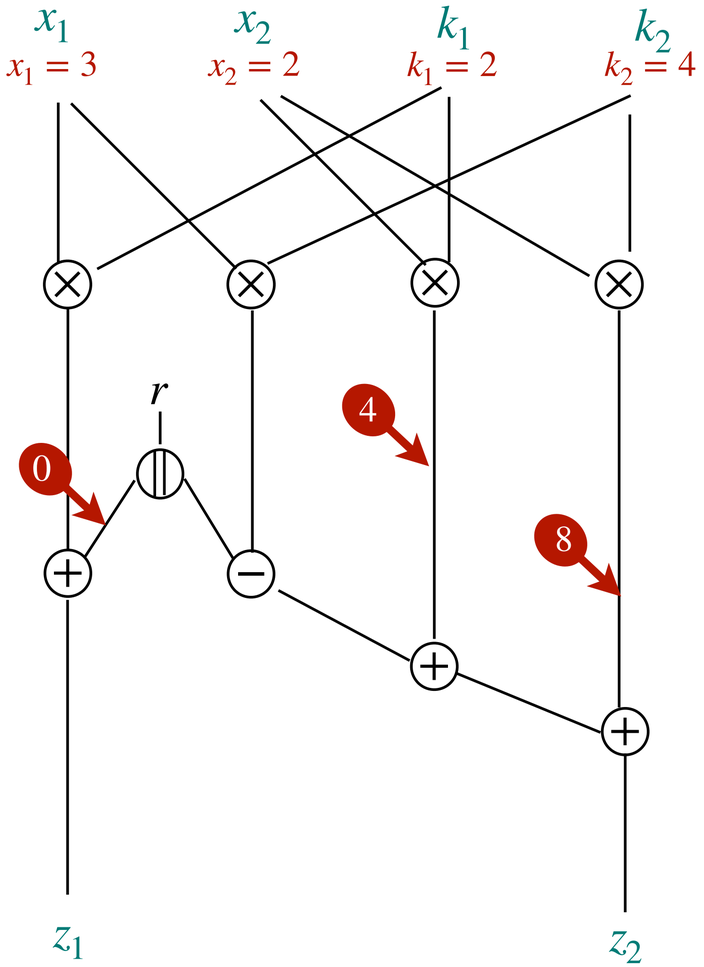


Attacker model

The attacker is given the value of each wire with probability p.

[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

Random Probing Security



Attacker model

The attacker is given the value of each wire with probability p.

(p, ϵ)-random-probing security

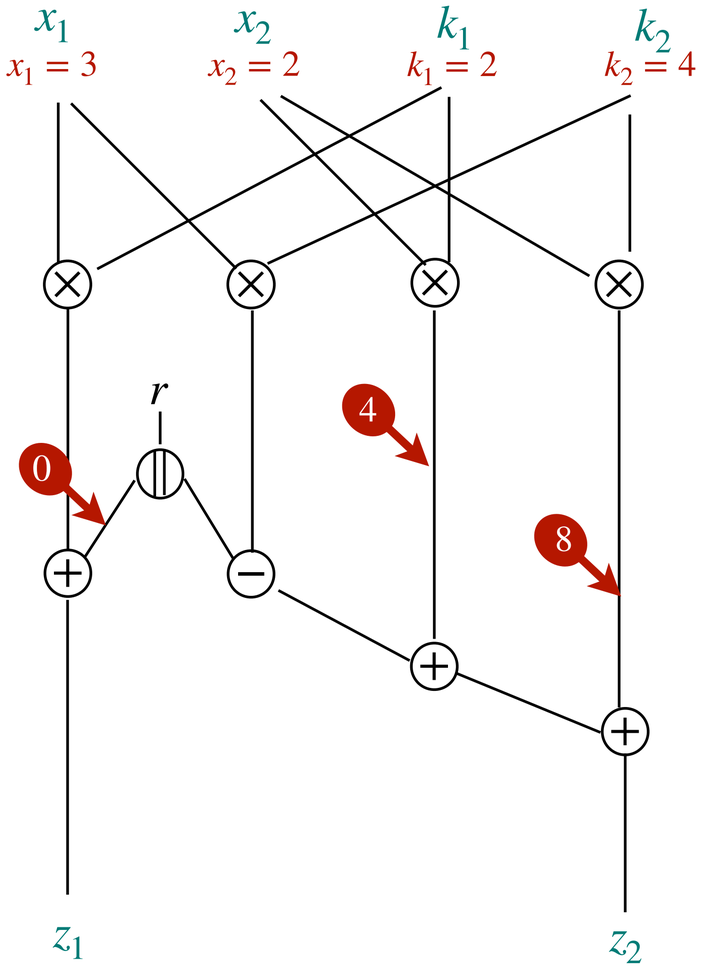
Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \text{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a simulator such that

$$\text{Sim}() \overset{\epsilon}{\approx} \mathscr{L}$$

[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

Random Probing Security



Attacker model

The attacker is given the value of each wire with probability p.

(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \text{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

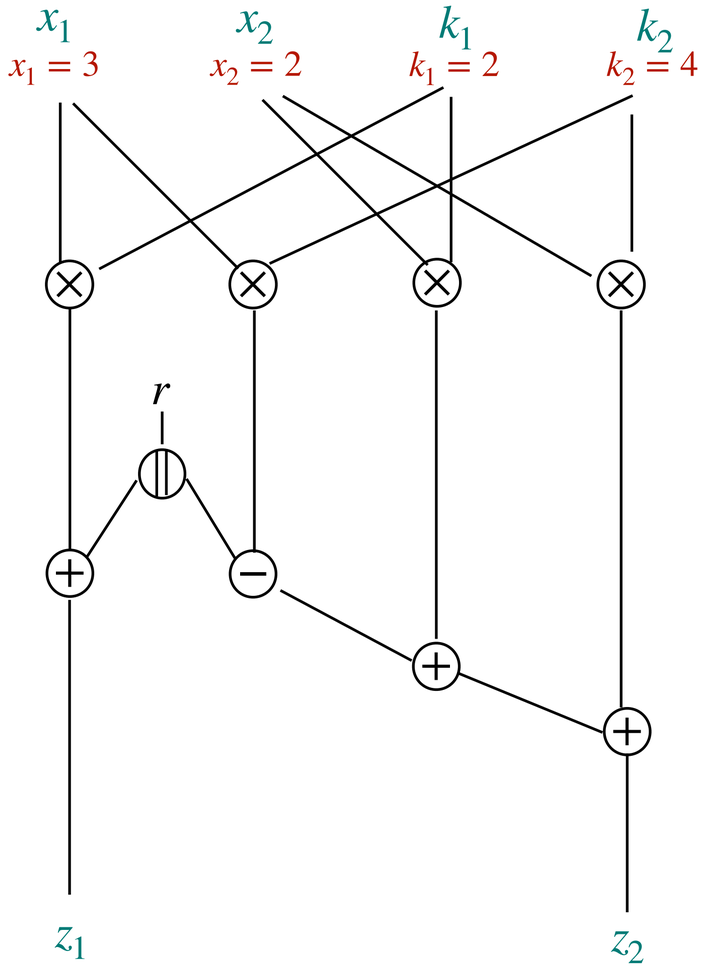
There exists a simulator such that

$\mathscr{L}$ can contain up to 19 values, one for each wire !

$$\text{Sim}() \overset{\epsilon}{\approx} \mathscr{L}$$

[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

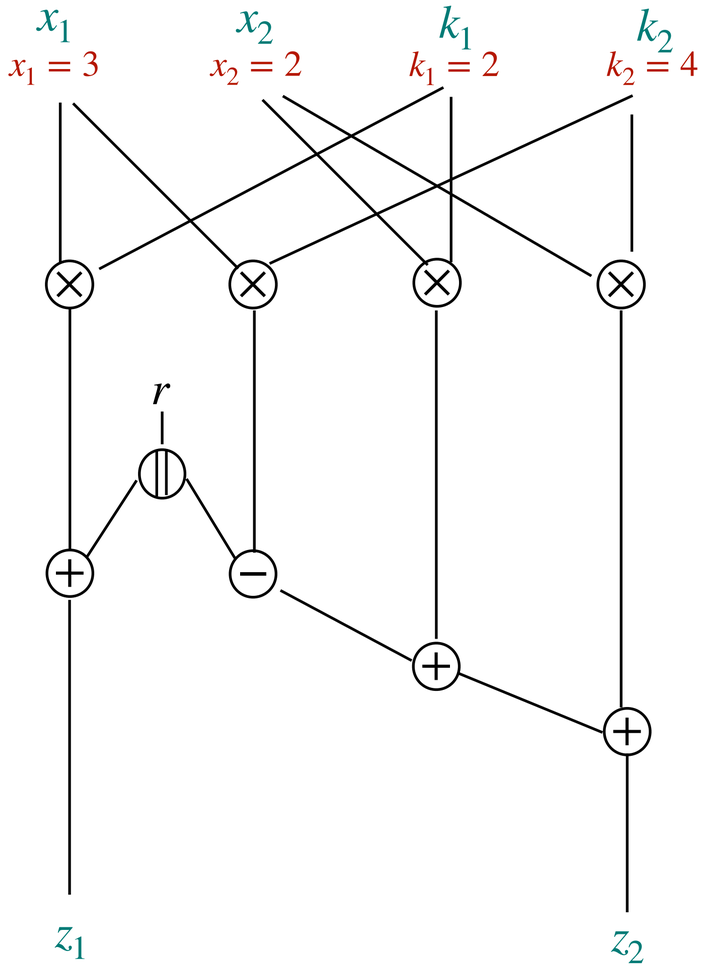
Random Probing Security



Reality

Simulation

Random Probing Security

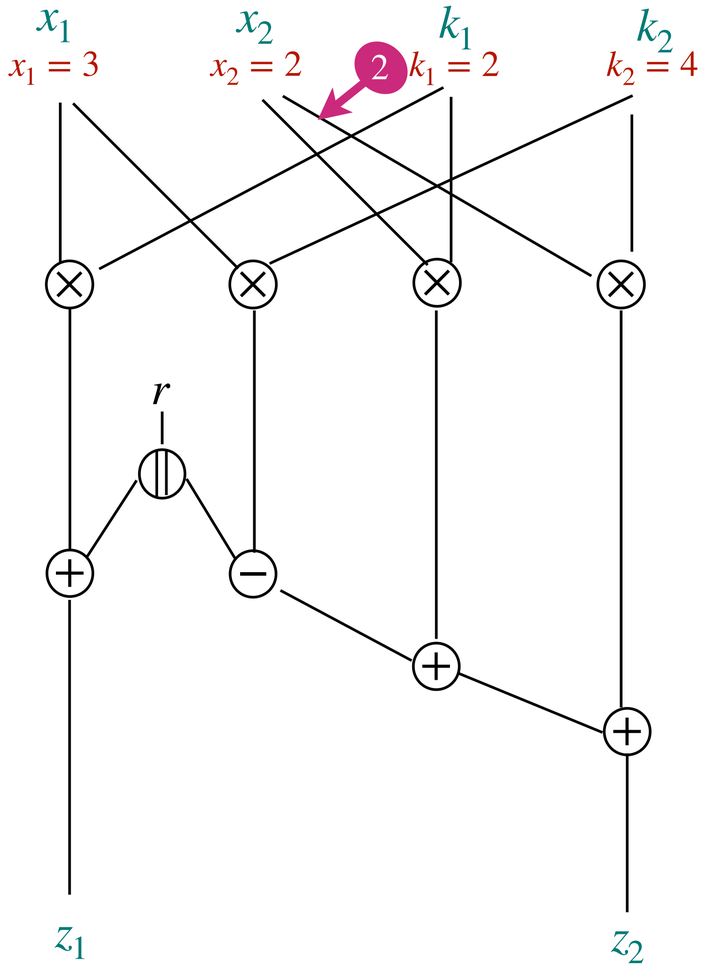


Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

Simulation

Random Probing Security



Reality

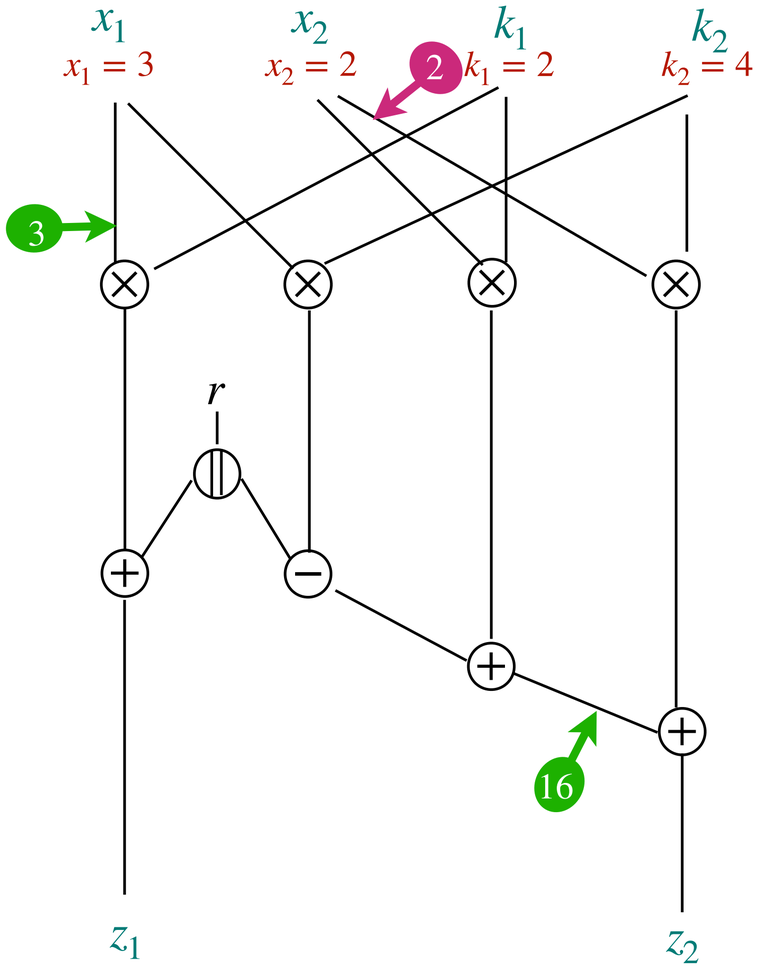
$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

Simulation

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

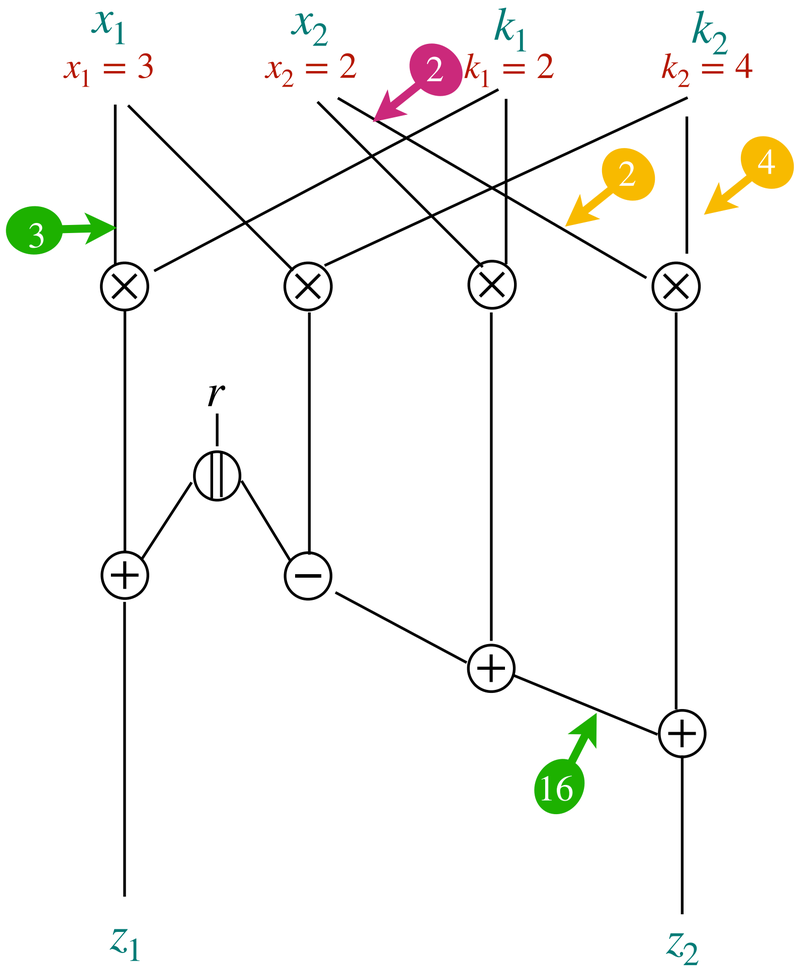
$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

Simulation

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

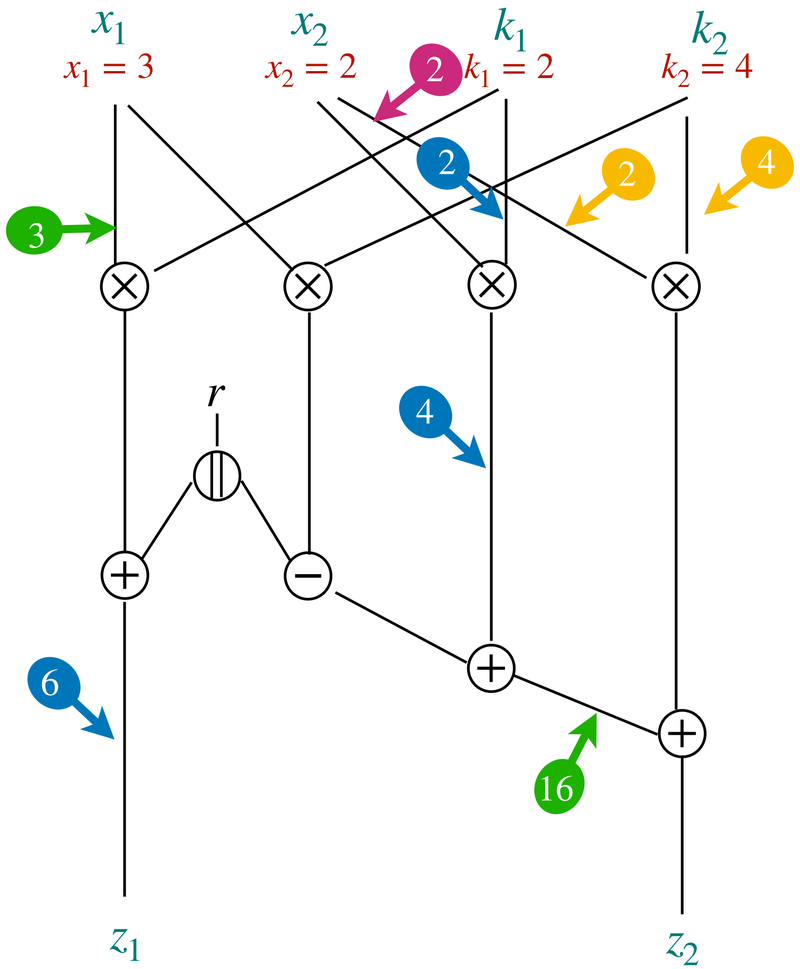
$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

Simulation

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

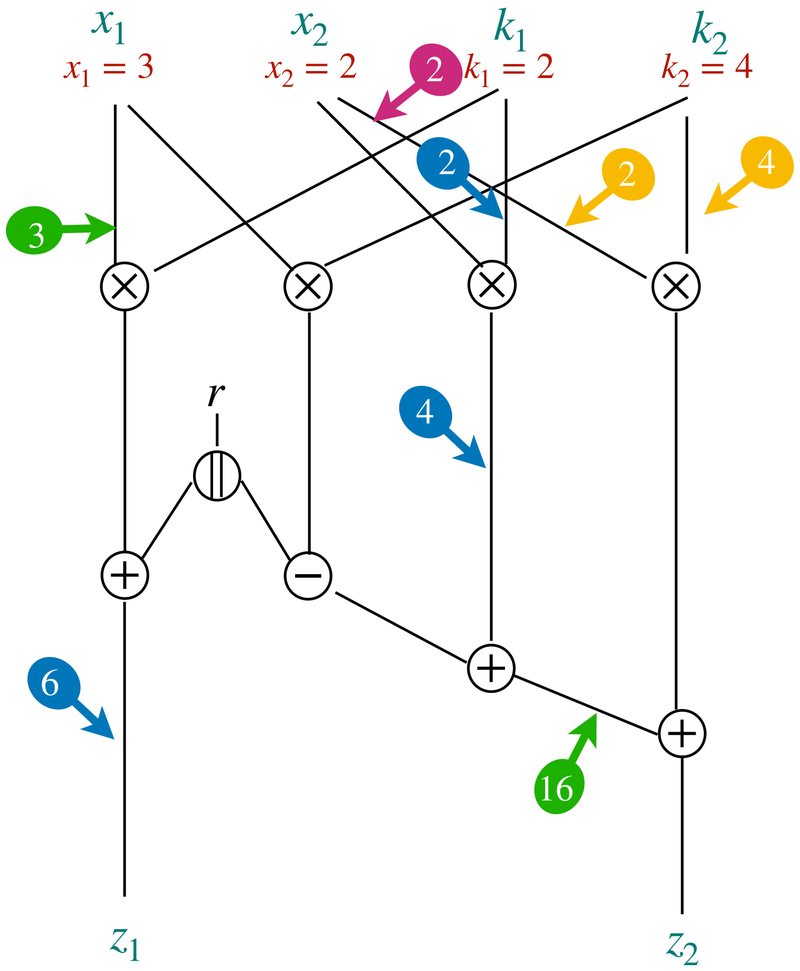
$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

Simulation

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

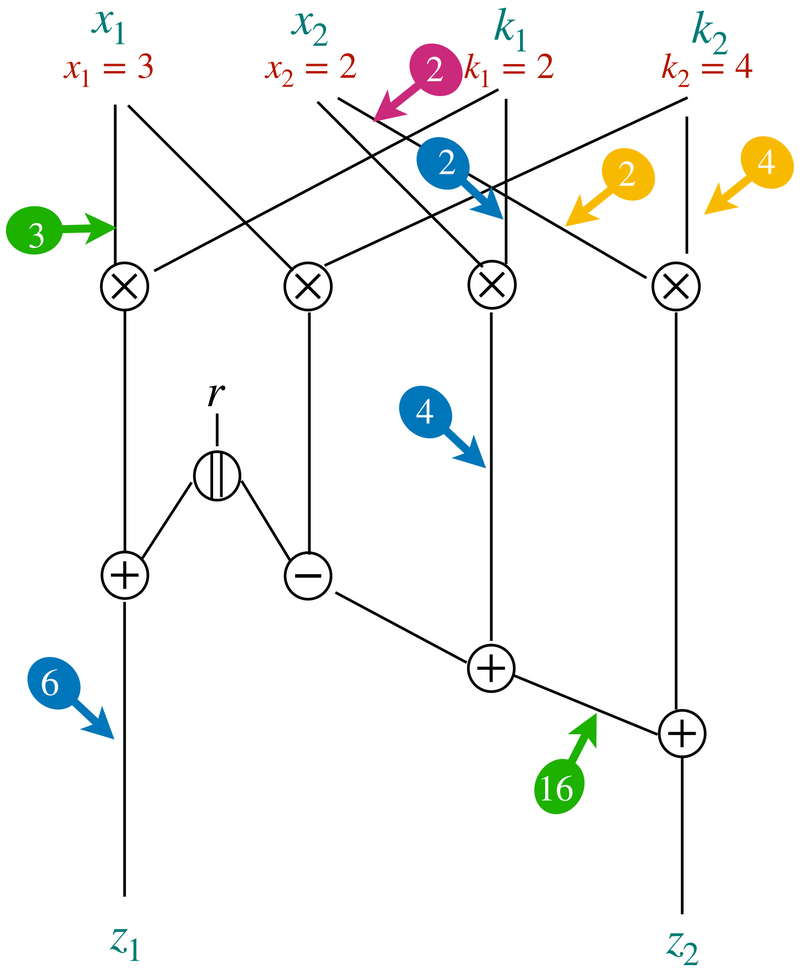
$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

Simulation

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

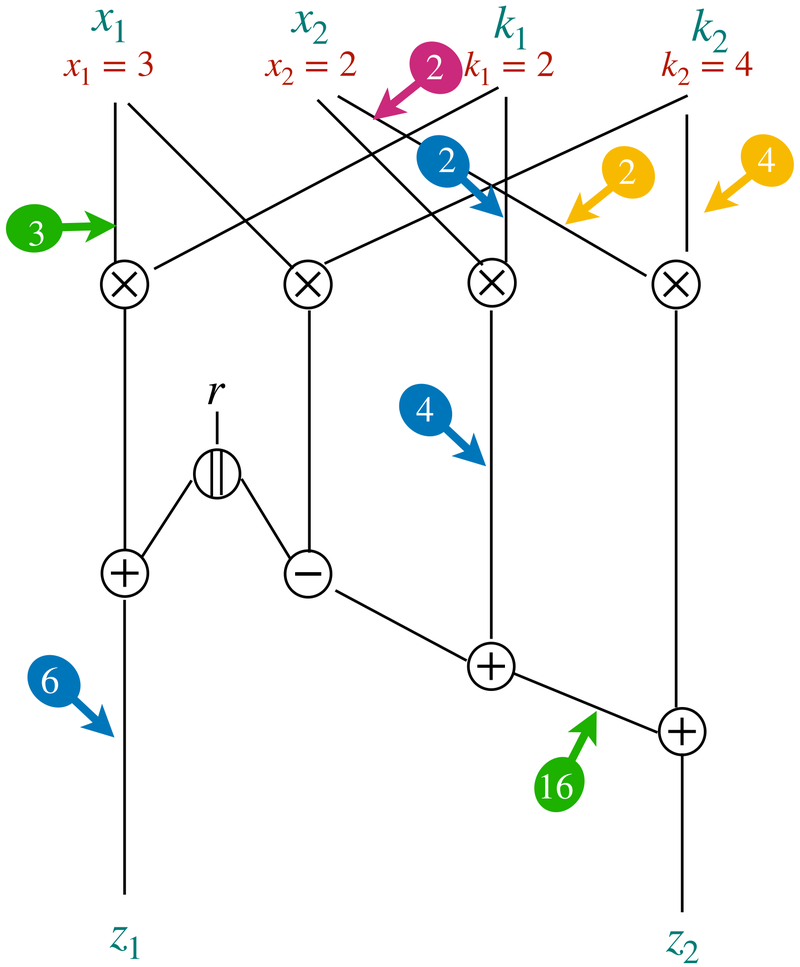
$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

Simulation

1. Draw $\mathscr{W}$
2. Simulate the corresponding $\mathscr{L}$

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

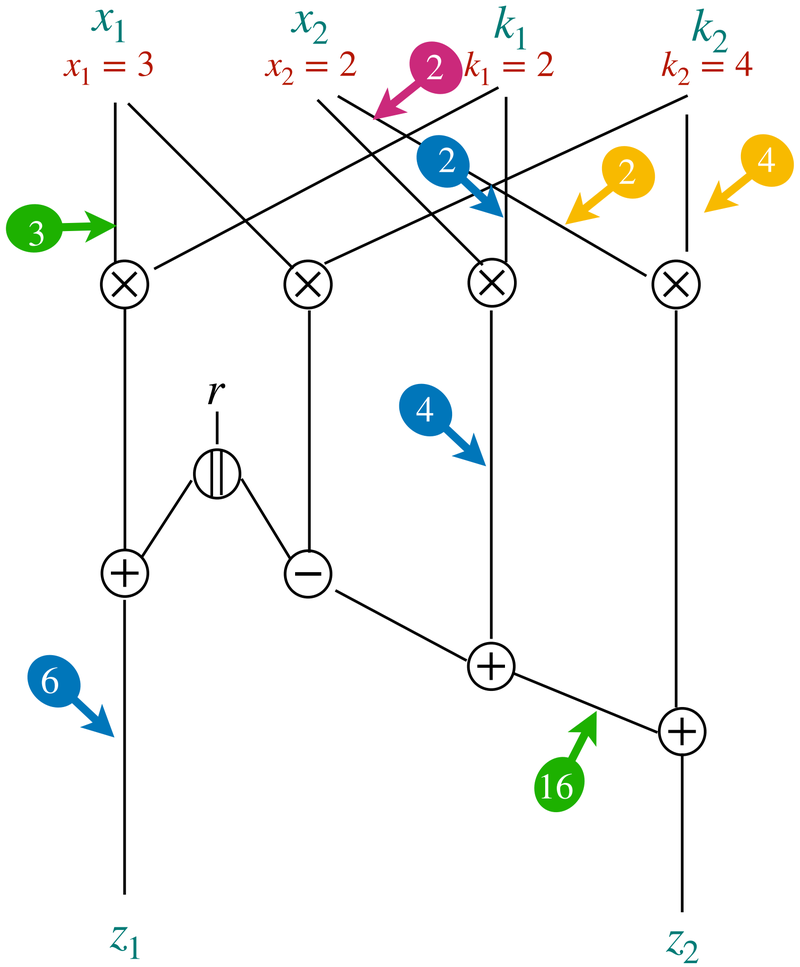
… $\approx 2^{19}$ possible sets $\mathscr{W}$

Simulation

1. Draw $\mathscr{W}$
2. Simulate the corresponding $\mathscr{L}$

The distribution of some $\mathscr{W}$ depend on the secret, they cannot be simulated.

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

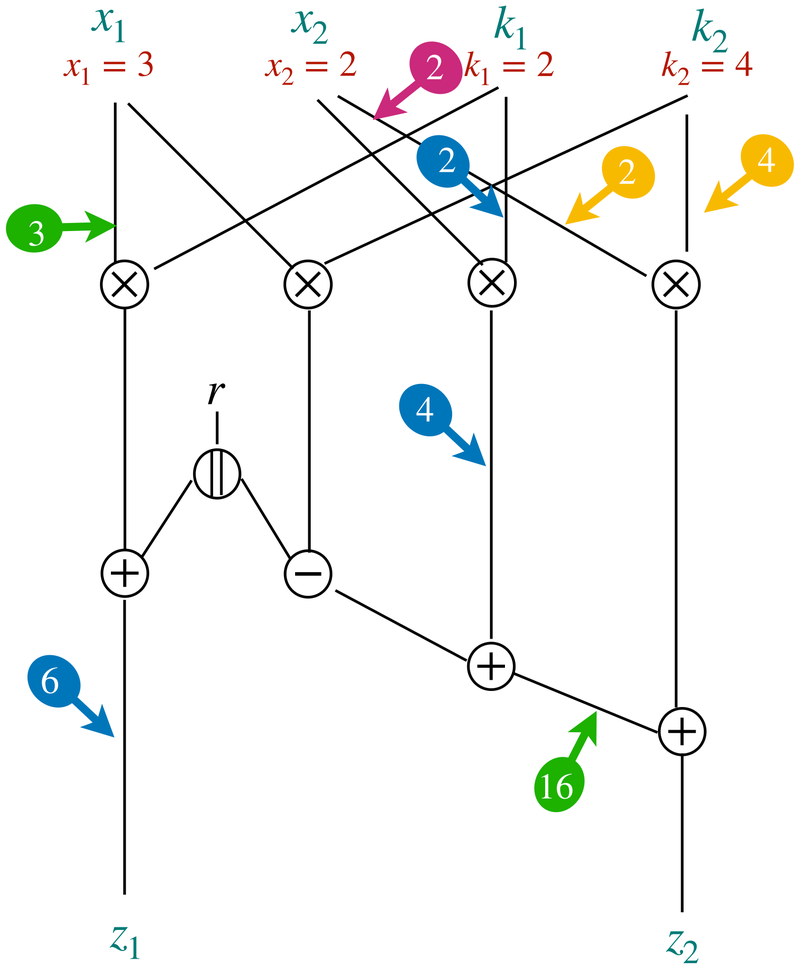
Simulation

1. Draw $\mathscr{W}$
2. Simulate the corresponding $\mathscr{L}$

The distribution of some $\mathscr{W}$ depend on the secret, they cannot be simulated.

Sim() $\overset{\epsilon}{\approx}$ $\mathscr{L}$

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

Simulation

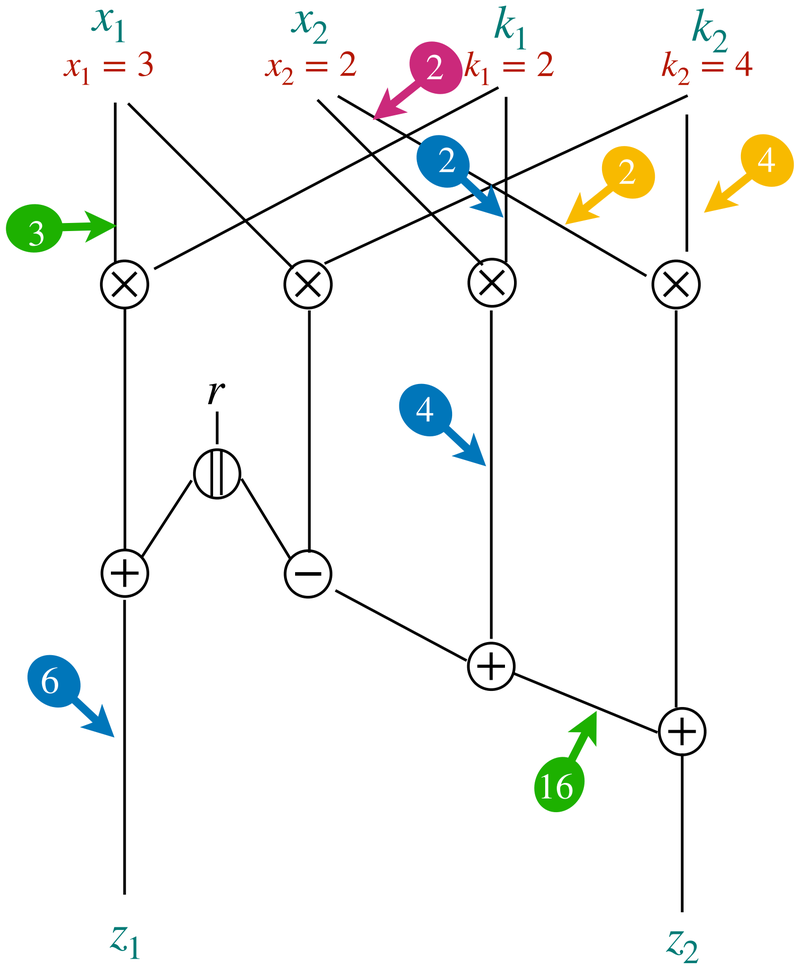
1. Draw $\mathscr{W}$
2. Simulate the corresponding $\mathscr{L}$

The distribution of some $\mathscr{W}$ depend on the secret, they cannot be simulated.

$\text{Sim}() \overset{\epsilon}{\approx} \mathscr{L}$

ϵ is the probability of drawing a $\mathscr{W}$ that cannot be simulated. It depends on p and on the circuit.

Random Probing Security



Reality

$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

$\ldots \approx 2^{19}$ possible sets $\mathscr{W}$

Simulation

1. Draw $\mathscr{W}$
2. Simulate the corresponding $\mathscr{L}$

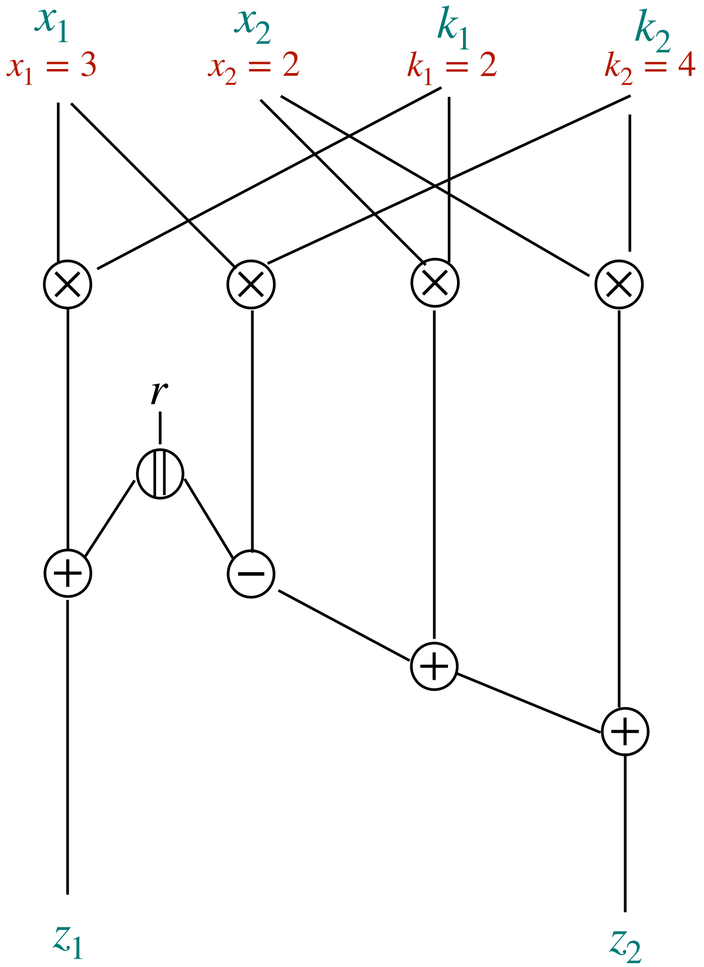
The distribution of some $\mathscr{W}$ depend on the secret, they cannot be simulated.

$\text{Sim}() \overset{\epsilon}{\approx} \mathscr{L}$

ϵ is the probability of drawing a $\mathscr{W}$ that cannot be simulated. It depends on p and on the circuit.

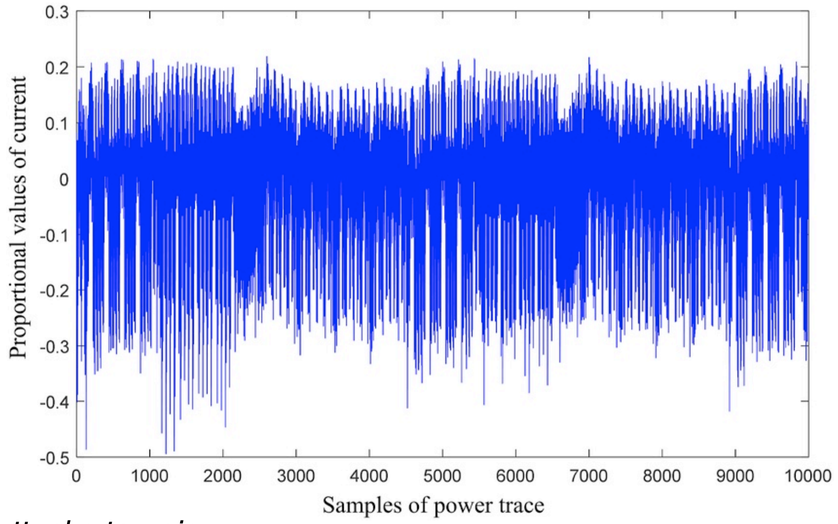
$\epsilon = 2^{-128} \implies p \geq$ some bound

Noisy leakage model



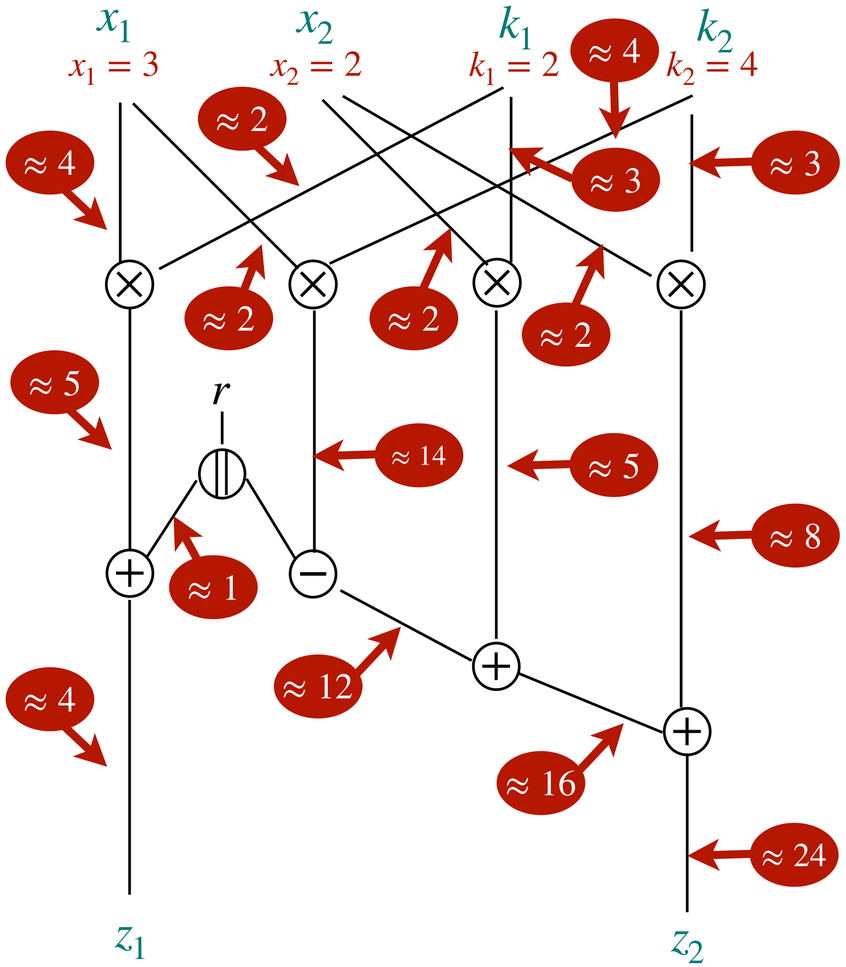
Attacker model

The attacker is given the value of each wire with a certain noise. More precisely, the attacker gets $||_i \left(f(v_i) + \eta\right)$ for each wire v_i, noise η_i and a leakage function f.



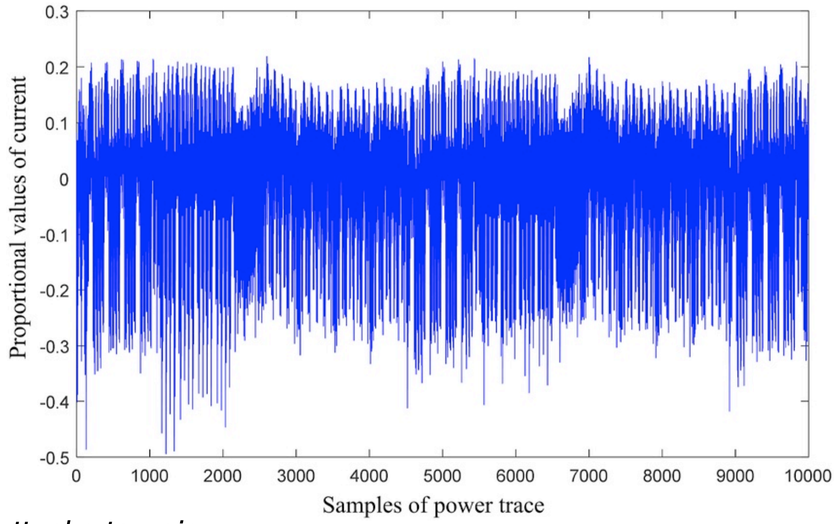
[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

Noisy leakage model



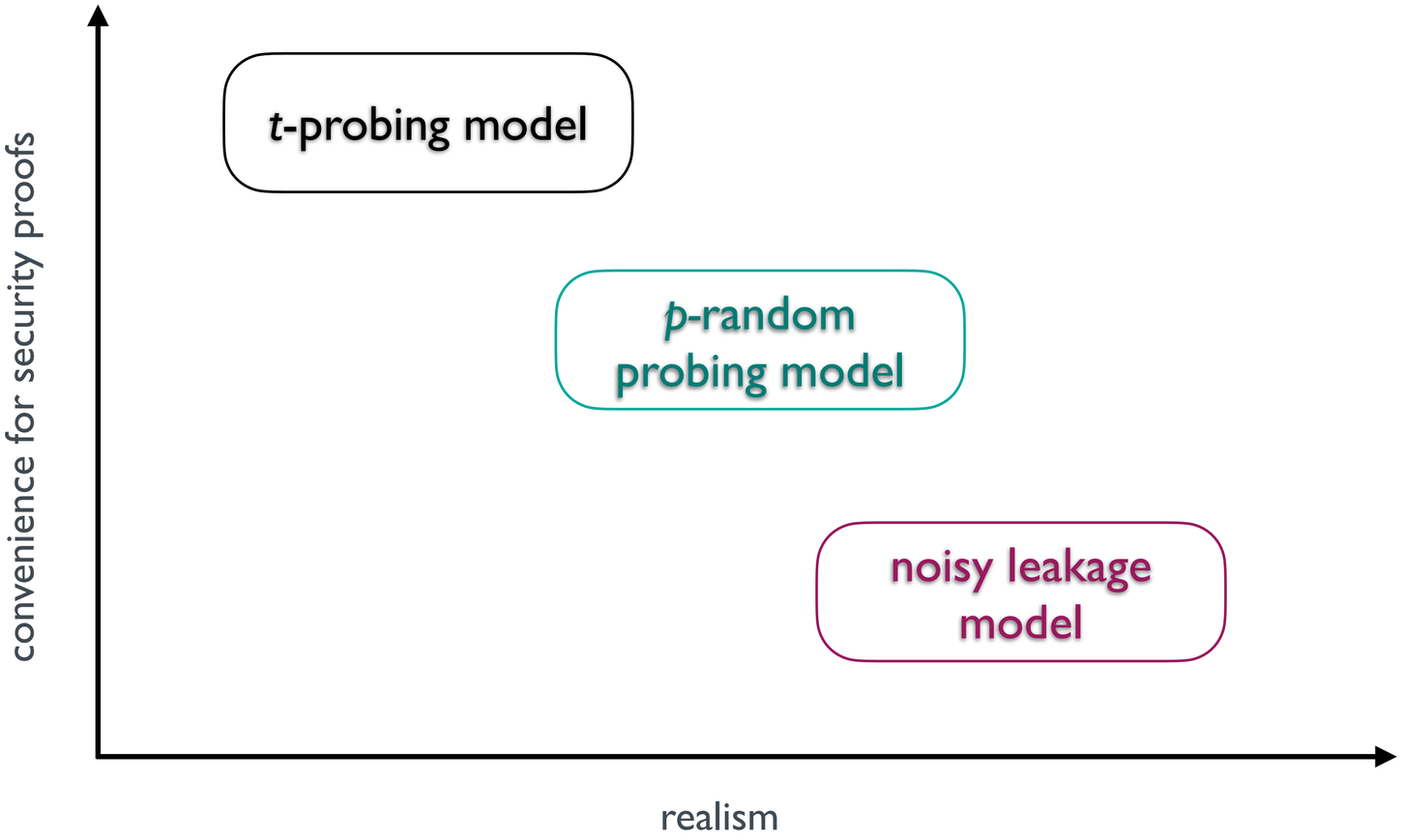
Attacker model

The attacker is given the value of each wire with a certain noise. More precisely, the attacker gets $||_i \left(f(v_i) + \eta\right)$ for each wire v_i, noise η_i and a leakage function f.

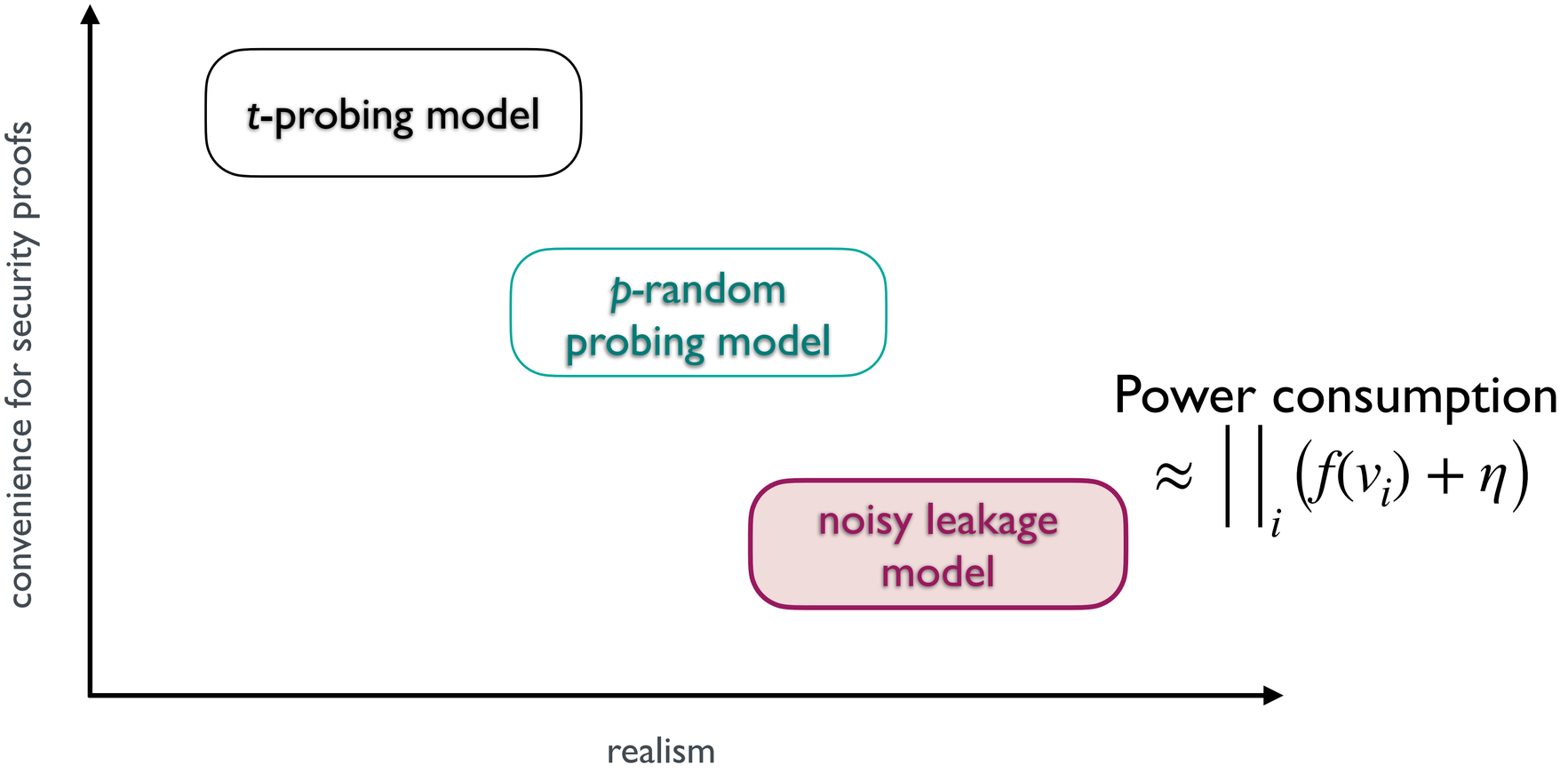


[DDF14] A. Duc, S. Dziembowski, S. Faust. *Unifying leakage models: From probing attacks to noisy leakage*. EUROCRYPT 2014

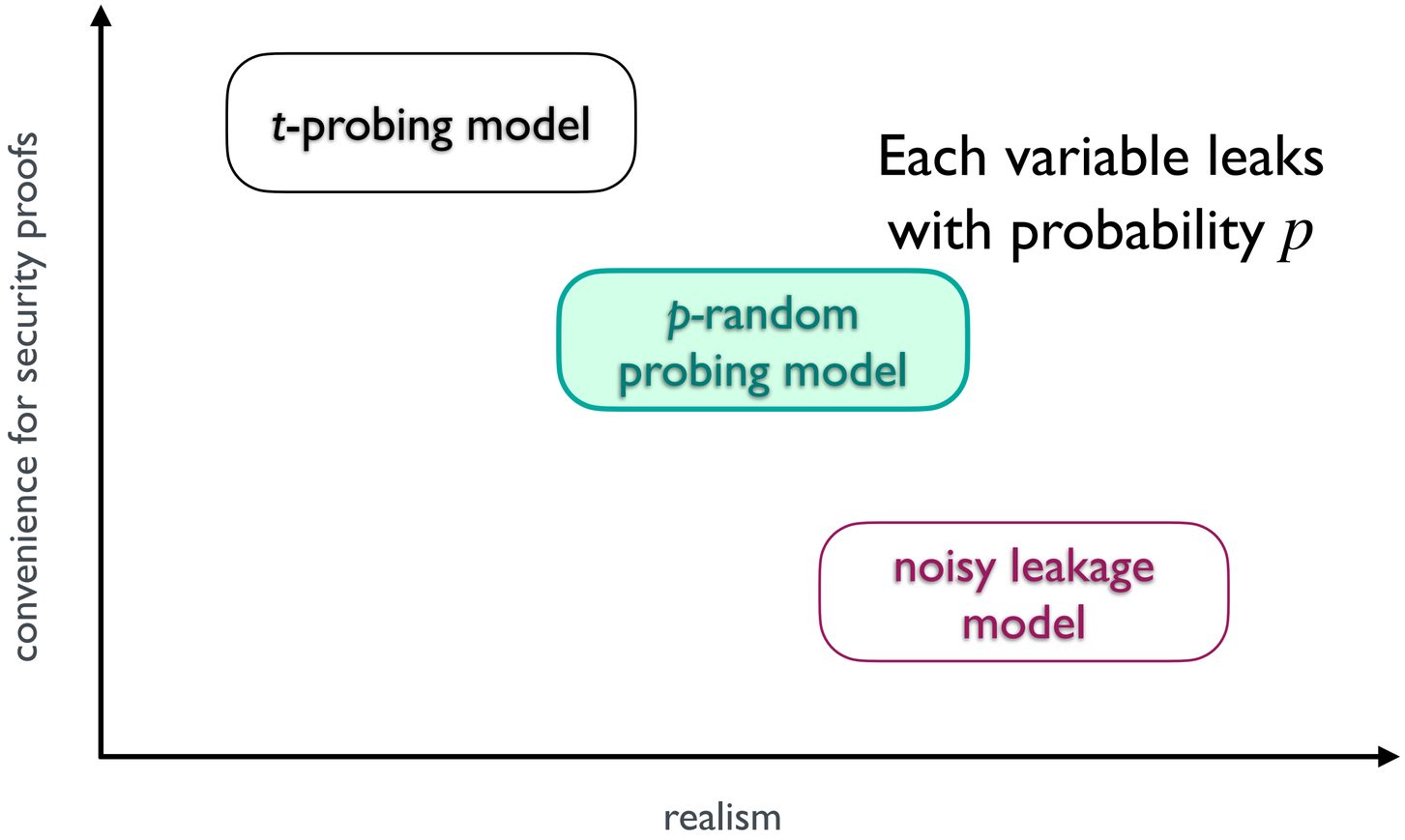
Leakage Models



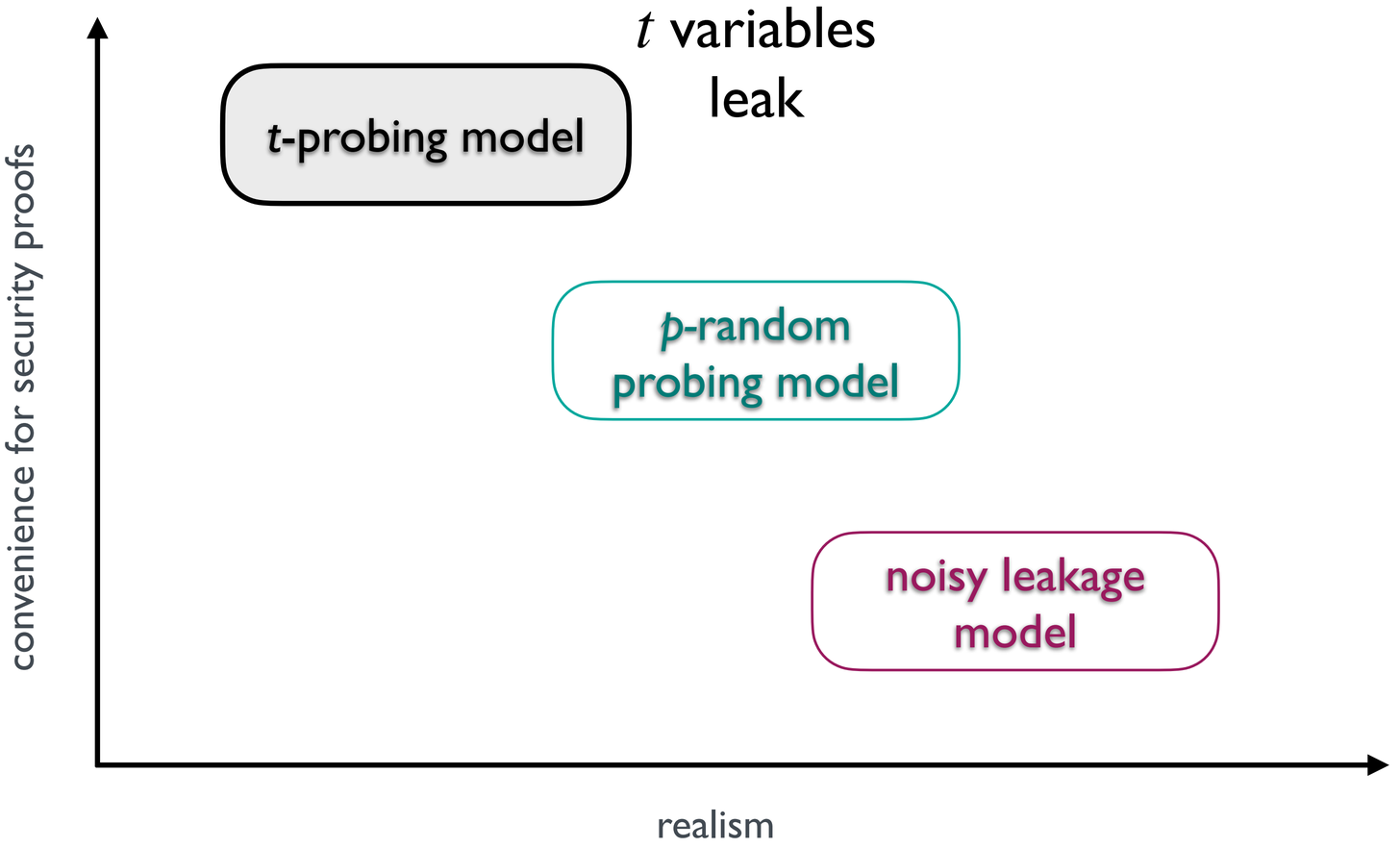
Leakage Models



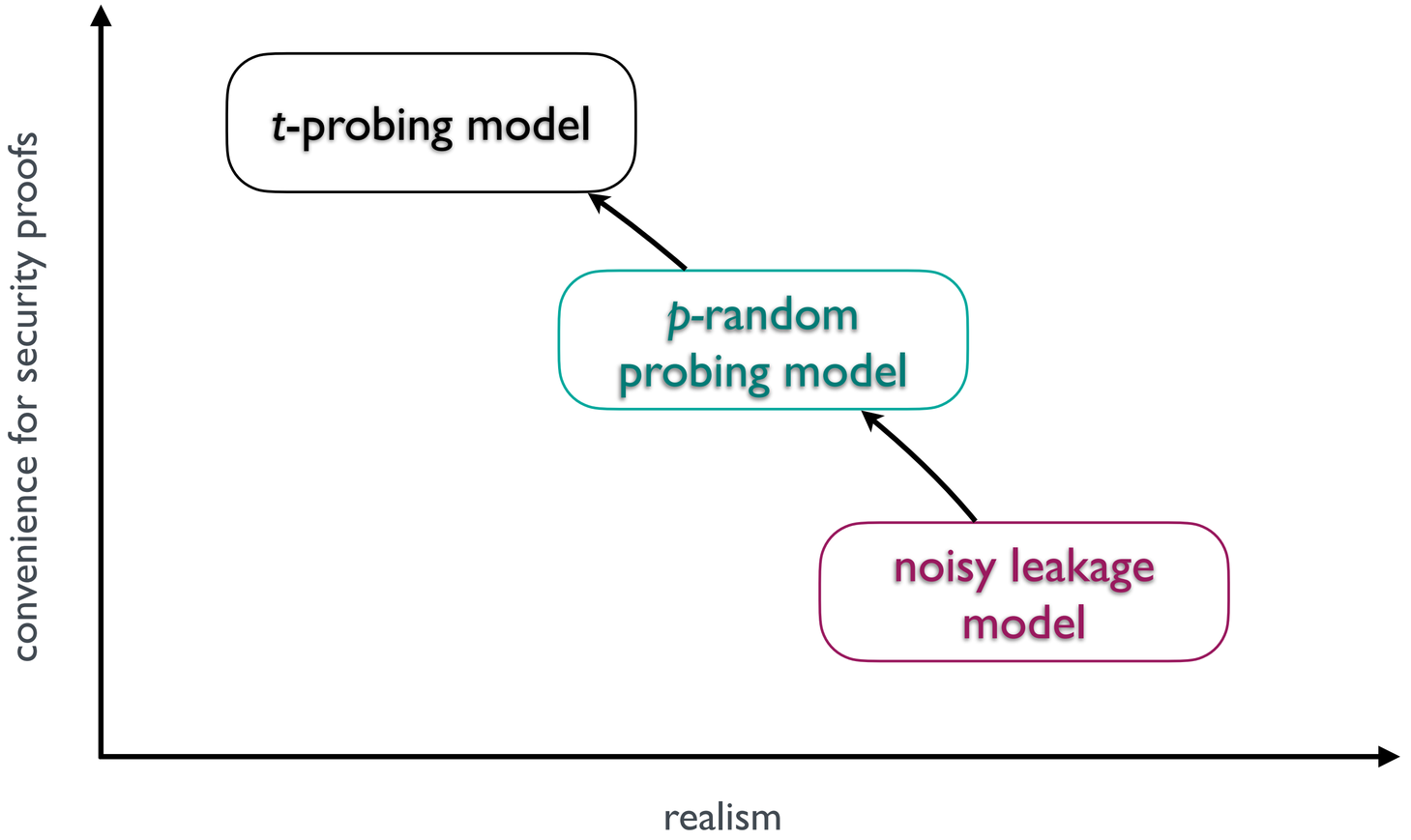
Leakage Models



Leakage Models



Leakage Models



[DDF14] *Unifying Leakage Models: From Probing Attacks to Noisy Leakage*. Alexandre Duc, Stefan Dziembowski, Sebastian Faust. Eurocrypt 2014

☑ Perfect simulation, can be immediately plugged into the black-box security.

➡ Extra information can be handled (e.g. [dPKPR24] , [BBEF+19])

☑ Comprehensive toolbox for proofs: many gadgets and composition techniques.

☐ Loose reduction to the noisy leakage model

➡ Even with a perfectly identified leakage of a chip, the required masking order is prohibitively high.

➡ Masked implementations may not be practically secure.

[dPKPR24] R. del Pino, S. Katsumata, T. Prest and M. Rossi *Raccoon: A Masking-Friendly Signature Proven in the Probing Model.* CRYPTO 2024

[BBEF+19] G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, B. Grégoire, M. Rossi and Mehdi Tibouchi. *Masking the GLP Lattice-Based Signature Scheme at Any Order.* EUROCRYPT 2019

☑ Perfect simulation, can be immediately plugged into the black-box security.

➡ Extra information can be handled (e.g. [dPKPR24] , [BBEF+19])

☑ Comprehensive toolbox for proofs: many gadgets and composition techniques.

☐ Loose reduction to the noisy leakage model

➡ Even with a perfectly identified leakage of a chip, the required masking order is prohibitively high.

➡ Masked implementations may not be practically secure.

☑ Close link with the physical leakage of a chip.

☑ Existing studies for specific gadgets/ operations.

☐ No theoretical framework for proofs and composition (except with leak free gadgets)

[dPKPR24] R. del Pino, S. Katsumata, T. Prest and M. Rossi *Raccoon: A Masking-Friendly Signature Proven in the Probing Model.* CRYPTO 2024

[BBEF+19] G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, B. Grégoire, M. Rossi and Mehdi Tibouchi. *Masking the GLP Lattice-Based Signature Scheme at Any Order.* EUROCRYPT 2019

[PR13] E. Prouff M. Rivain. *Masking against Side-Channel Attacks: a Formal Security Proof* EUROCRYPT 2013

[KSB24] D. Kamel, F.-X. Standaert and O. Bronchain, *Information Theoretic Evaluation of Raccoon's Side-Channel Leakage.* CiC 2024

t-probing model

- ☑ Perfect simulation, can be immediately plugged into the black-box security.
  - ➡ Extra information can be handled (e.g. [dPKPR24] , [BBEF+19])
- ☑ Comprehensive toolbox for proofs: many gadgets and composition techniques.
- ☐ Loose reduction to the noisy leakage model
  - ➡ Even with a perfectly identified leakage of a chip, the required masking order is prohibitively high.
  - ➡ Masked implementations may not be practically secure.

p-random probing model

- ☑ The success of the simulation depends on the probability p.
  - ➡ semi-direct link with the physical leakage of a chip.
  - ➡ Masked implementations are provably secure up to a certain leakage probability p. For concrete chips, p can lie between 2^{-15} to 2^{-7} ([BCMRRST25])
- ☐ Relatively high entrance price for understanding the proofs.
- ☐ The toolbox remains to be designed.
  - ➡ Could be plugged into the black-box security.
  - ➡ Not a lot of gadgets
  - ➡ The composition techniques have not converged yet.

noisy leakage model

- ☑ Close link with the physical leakage of a chip.
- ☑ Existing studies for specific gadgets/ operations.
- ☐ No theoretical framework for proofs and composition (except with leak free gadgets)

[BCMRRST25] S. Belaïd, G. Cassiers, C. Mutschler, M. Rivain, T. Roche, F.-X Standaert and A. R. Taleb. A Methodology to Achieve Provable Side-Channel Security in Real-World Implementations. CiC 2025

[PR13] E. Prouff M. Rivain. *Masking against Side-Channel Attacks: a Formal Security Proof* EUROCRYPT 2013

[KSB24] D. Kamel, F.-X. Standaert and O. Bronchain, *Information Theoretic Evaluation of Raccoon's Side-Channel Leakage.* CiC 2024

t-probing model

- ☑ Perfect simulation, can be immediately plugged into the black-box security.
  - ➡ Extra information can be handled (e.g. [dPKPR24] , [BBEF+19])
- ☑ Comprehensive toolbox for proofs: many gadgets and composition techniques.
- ☐ Loose reduction to the noisy leakage model
  - ➡ Even with a perfectly identified leakage of a chip, the required masking order is prohibitively high.
  - ➡ Masked implementations may not be practically secure.

p-random probing model

- ☑ The success of the simulation depends on the probability p.
  - ➡ semi-direct link with the physical leakage of a chip.
  - ➡ Masked implementations are provably secure up to a certain leakage probability p. For concrete chips, p can lie between 2^{-15} to 2^{-7} ([BCMRRST25])
- ☐ Relatively high entrance price for understanding the proofs.
- ☐ The toolbox remains to be designed.
  - ➡ Could be plugged into the black-box security.
  - ➡ Not a lot of gadgets
  - ➡ The composition techniques have not converged yet.

noisy leakage model

- ☑ Close link with the physical leakage of a chip.
- ☑ Existing studies for specific gadgets/ operations.
- ☐ No theoretical framework for proofs and composition (except with leak free gadgets)

[BCMRRST25] S. Belaïd, G. Cassiers, C. Mutschler, M. Rivain, T. Roche, F.-X Standaert and A. R. Taleb. A Methodology to Achieve Provable Side-Channel Security in Real-World Implementations. CiC 2025

[PR13] E. Prouff M. Rivain. *Masking against Side-Channel Attacks: a Formal Security Proof* EUROCRYPT 2013

[KSB24] D. Kamel, F.-X. Standaert and O. Bronchain, *Information Theoretic Evaluation of Raccoon's Side-Channel Leakage.* CiC 2024

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

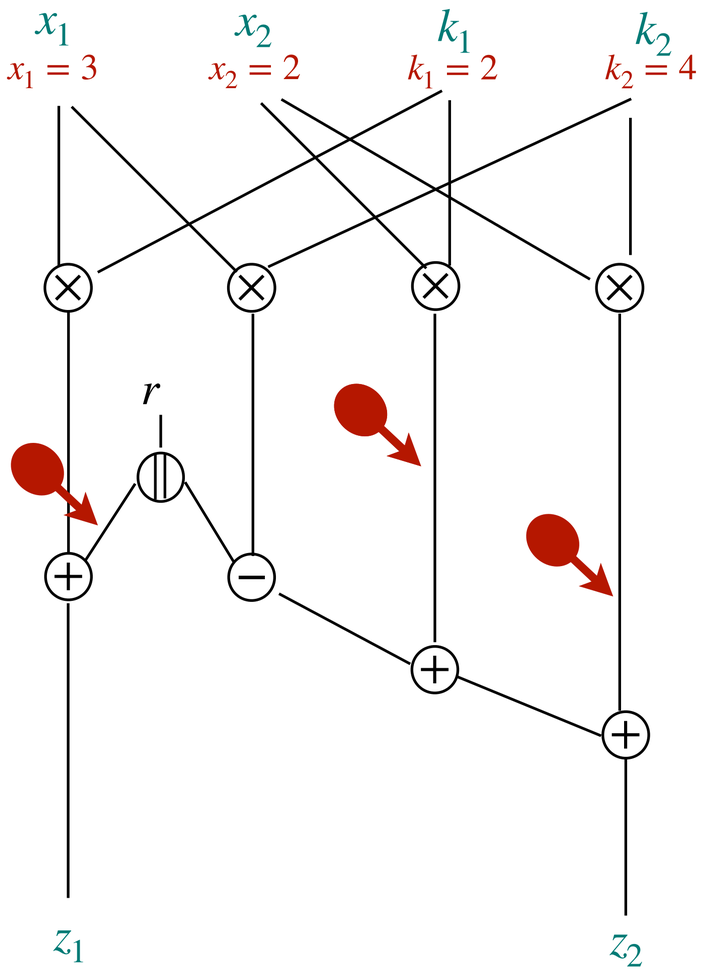
4) Ideas for future works

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

4) Ideas for future works



Random Probing Composability

(p, ϵ, t)-random-probing composability

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

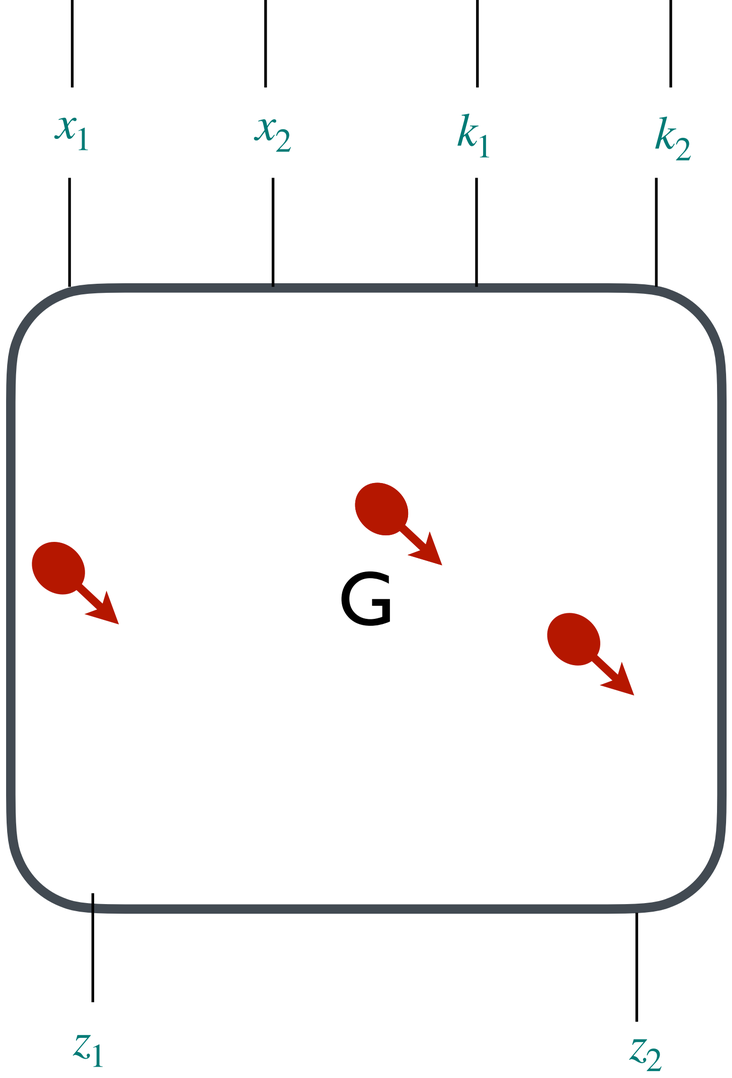
$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{W}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\mathsf{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\epsilon}{\approx} \mathscr{L}$$

This is simplified, I ignored the leaking outputs.

$\mathbb{P}$(« more than t shares of each $[|x|]$ and $[|k|]$ are required to simulate $\mathscr{L}$) $\leq \varepsilon$

[C:BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020



Random Probing Composability

(p, ϵ, t)-random-probing composability

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

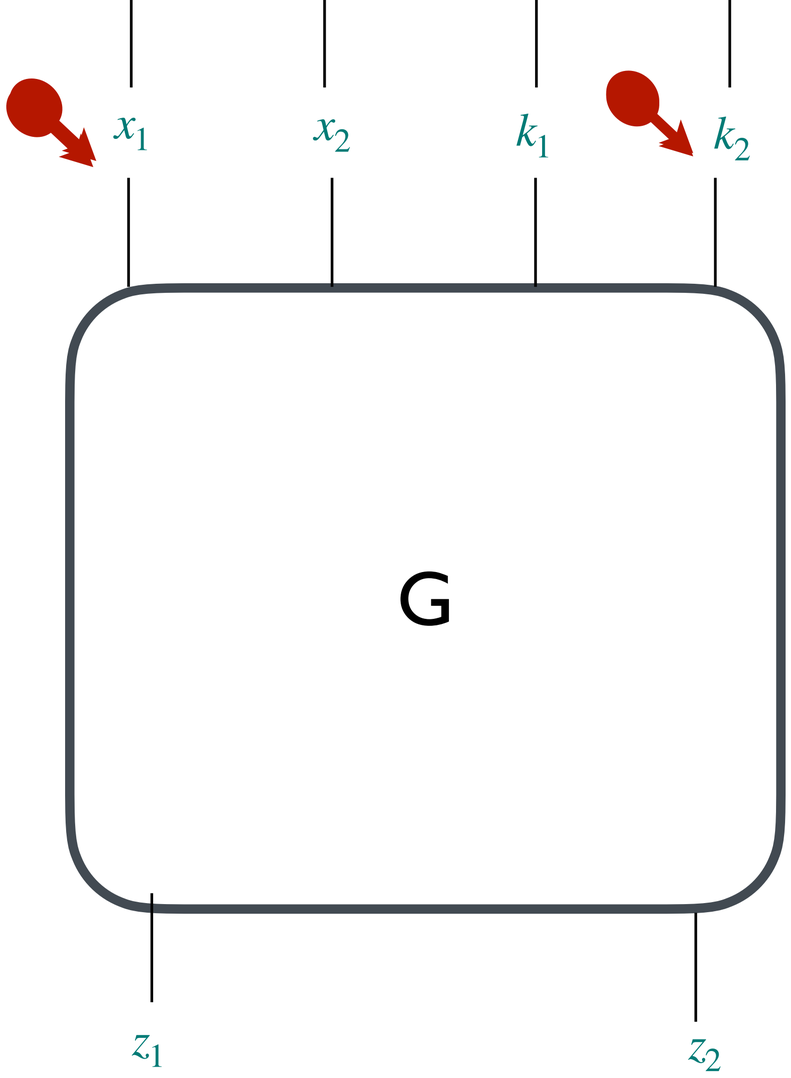
$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{W}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\mathsf{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

$$\text{and out} \overset{\epsilon}{\approx} \mathscr{L}$$

This is simplified, I ignored the leaking outputs.

$\mathbb{P}$(« more than t shares of each $[|x|]$ and $[|k|]$ are required to simulate $\mathscr{L}$) $\leq \varepsilon$

[C:BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020



Random Probing Composability

(p, ϵ, t)-random-probing composability

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{W}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\mathsf{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

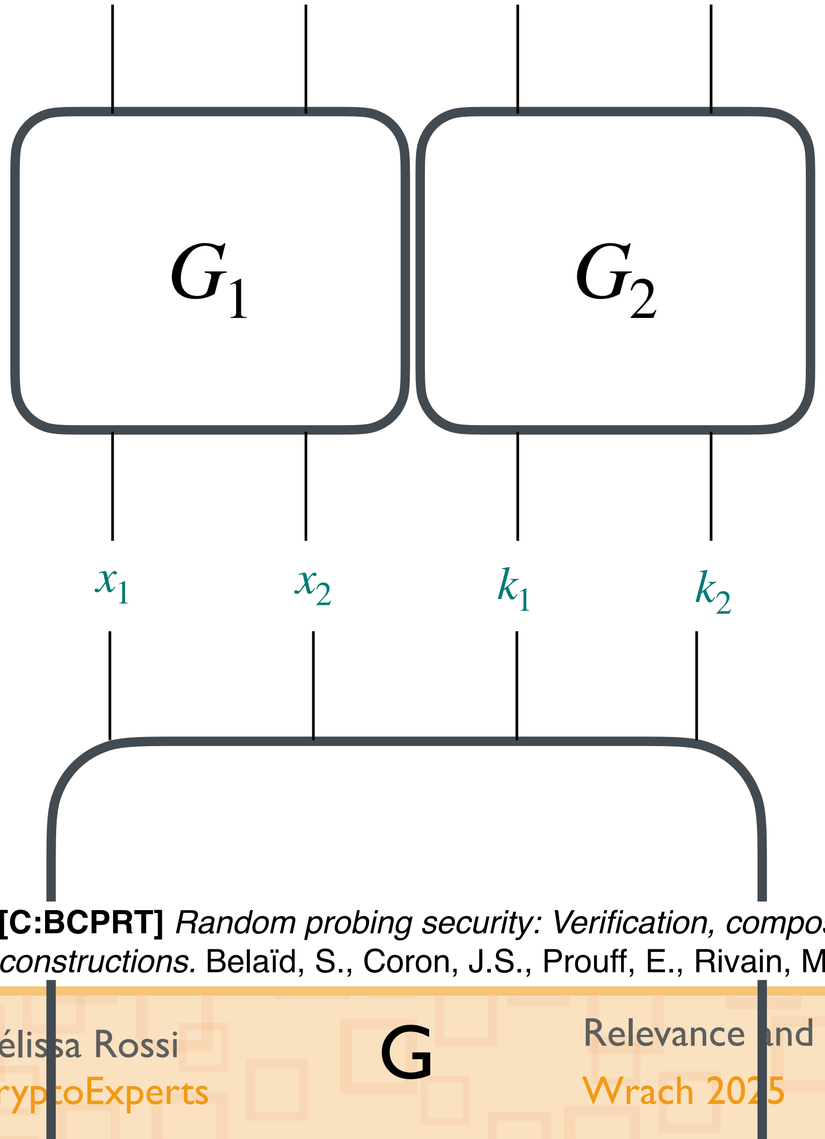
$$\text{and out} \overset{\epsilon}{\approx} \mathscr{L}$$

This is simplified, I ignored the leaking outputs.

$\mathbb{P}$(« more than t shares of each $[|x|]$ and $[|k|]$ are required to simulate $\mathscr{L}$) $\leq \varepsilon$

[C:BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability



(p, ϵ, t)-random-probing composability

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Let $(x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1})$ be uniform encodings of inputs x and k.
Let $\mathscr{L} \leftarrow \mathsf{AssignWires}(\mathscr{W}, (x_1, \ldots, x_{t+1}), (k_1, \ldots, k_{t+1}))$ be the values taken by the probes.

There exists a 2-stage simulator such that

$$I_x, I_k \leftarrow \mathsf{SimIn}(\mathscr{W}) \text{ with } |I_x| \leq t \text{ and } |I_k| \leq t$$
$$\mathsf{out} \leftarrow \mathsf{SimOut}(x_{|I_x}, k_{|I_k})$$

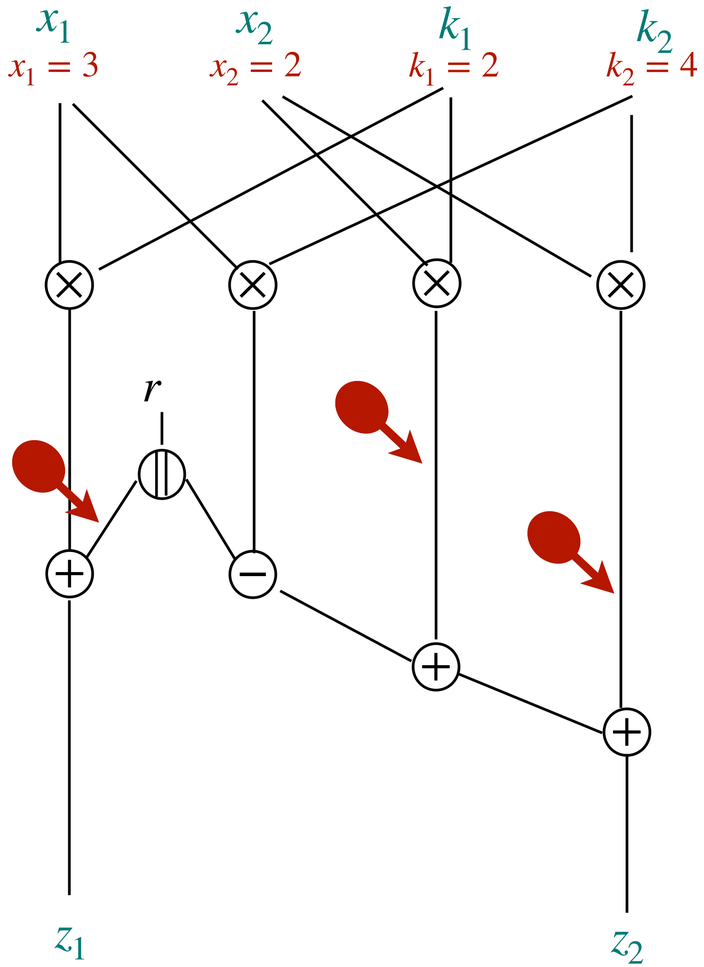
$$\text{and out} \overset{\epsilon}{\approx} \mathscr{L}$$

This is simplified, I ignored the leaking outputs.

$\mathbb{P}$(« more than t shares of each $[|x|]$ and $[|k|]$ are required to simulate $\mathscr{L}$) $\leq \varepsilon$

[C:BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability

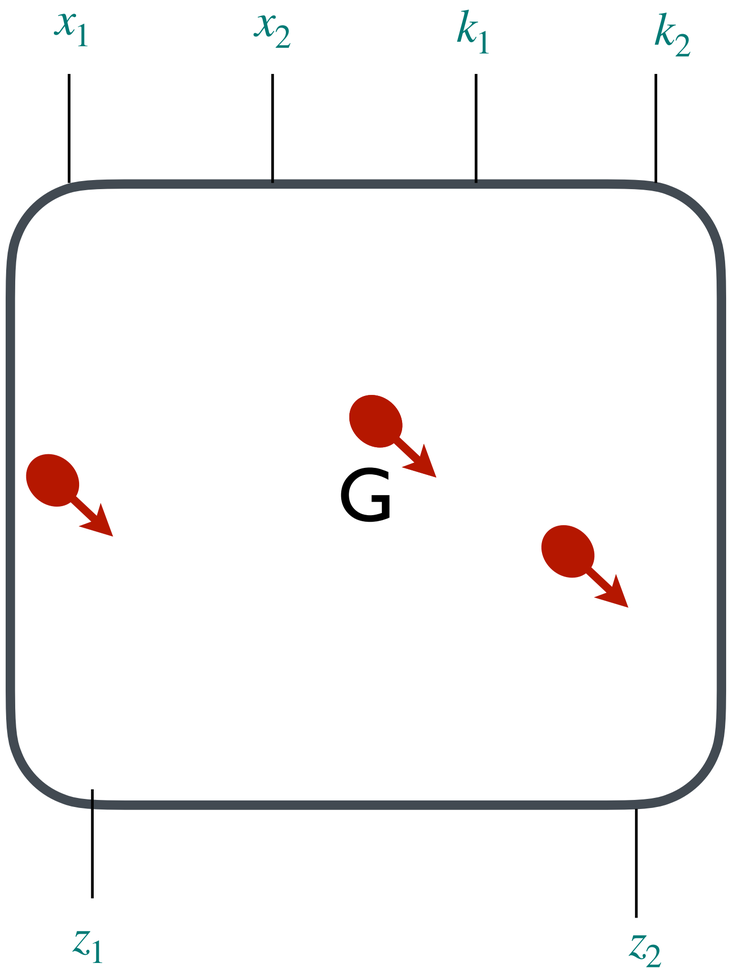


RPC:

Propagation of the simulation from inside to gadget to its inputs

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability

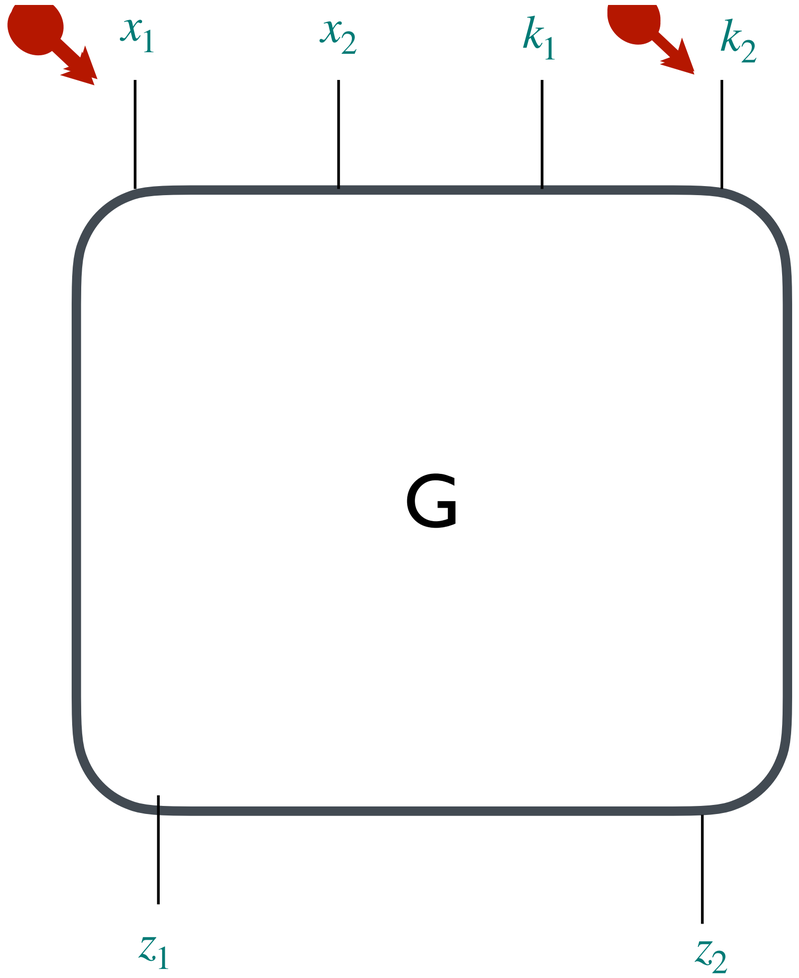


RPC:

Propagation of the simulation from inside to gadget to its inputs

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability

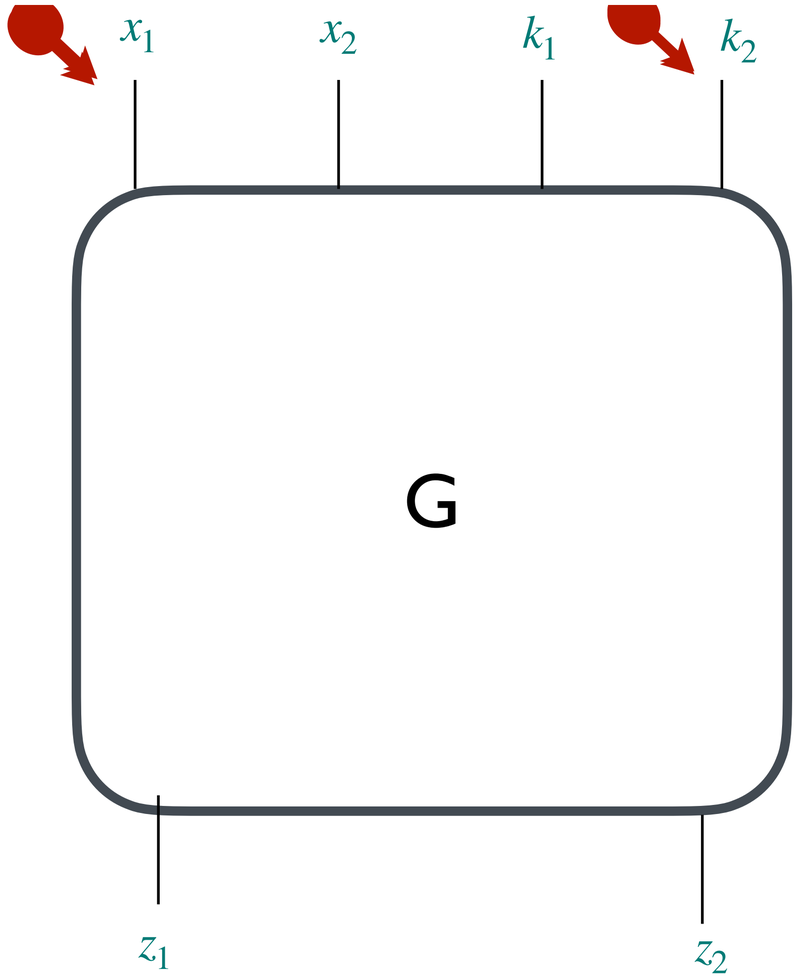


RPC:

Propagation of the simulation from inside to gadget to its inputs

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability



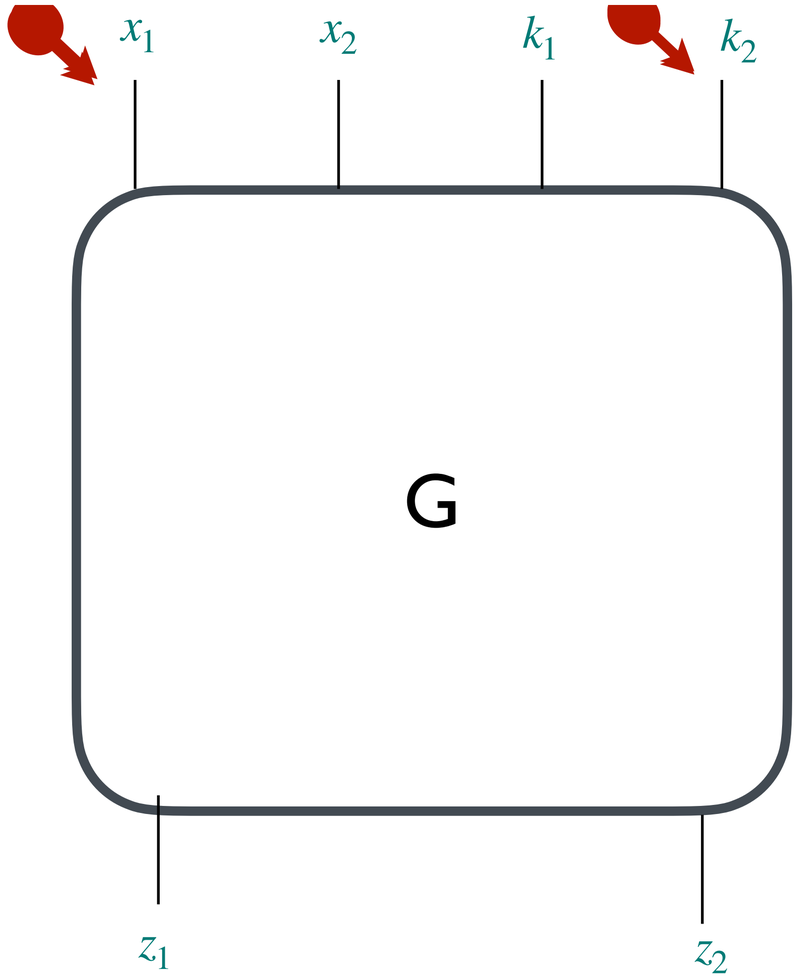
RPC:

Propagation of the simulation from inside to gadget to its inputs

Except with probability ϵ!

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability



RPC:

Propagation of the simulation from inside to gadget to its inputs

Except with probability ϵ!

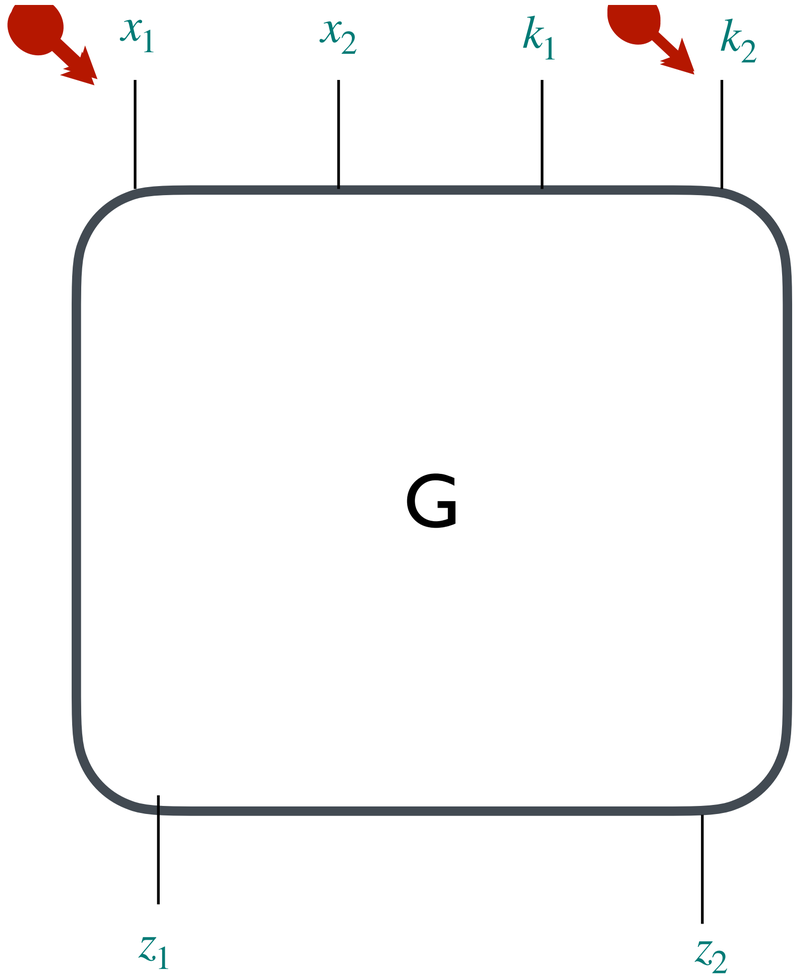
Composition

A circuit composed of (t, p, ϵ_i)-RPC gadgets will itself be (t, p, ϵ)-RPC with

$$\epsilon = 1 - \Pi_{i=0}^{k}(1 - \epsilon_i).$$

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability



RPC:

Propagation of the simulation from inside to gadget to its inputs

Except with probability ϵ!

Composition

A circuit composed of (t, p, ϵ_i)-RPC gadgets will itself be (t, p, ϵ)-RPC with

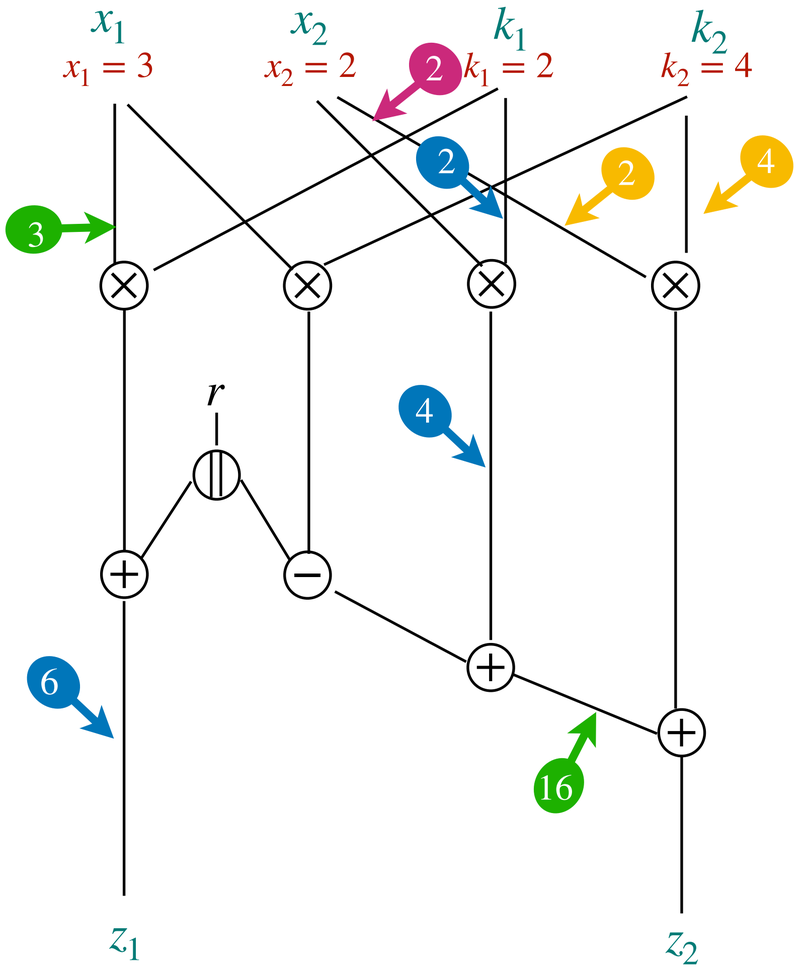
$$\epsilon = 1 - \Pi_{i=0}^{k}(1 - \epsilon_i).$$

The more gadgets, the higher the simulation is likely to fail.

[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.*
Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

Random Probing Composability

How to estimate the simulation failure probability ϵ ?



$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

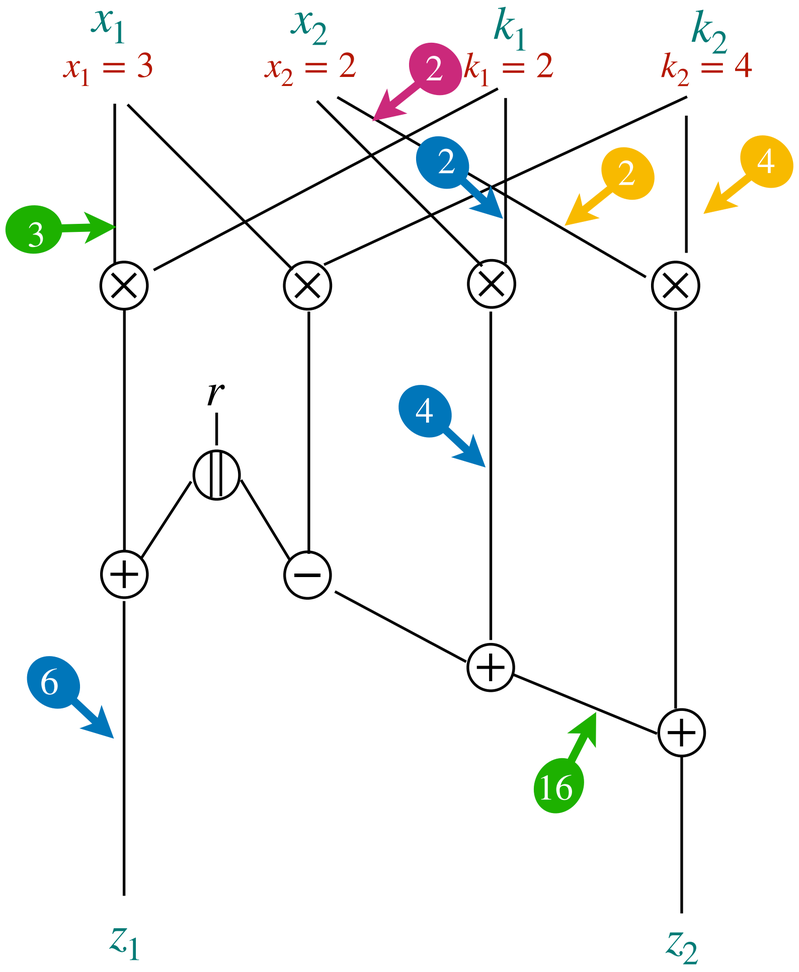
$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

Random Probing Composability

How to estimate the simulation failure probability ϵ ?



$\mathscr{W} = \emptyset$ with proba $(1-p)^{19}$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^{18}$

$\mathscr{L} = \{2\}$

$\mathscr{W} = \{x_1, r''\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{3,16\}$

$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

$\mathscr{L} = \{2,4\}$

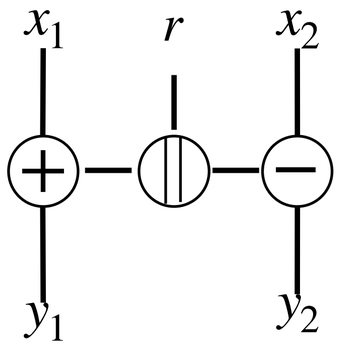
$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

$\mathscr{L} = \{6,4,2\}$

… $\approx 2^{19}$ possible sets $\mathscr{W}$

We need to
1) Identify all the sets $\mathscr{W}$ that cannot be simulated
2) Compute the probability to draw one of them.

How to concretely compute the ϵ ?



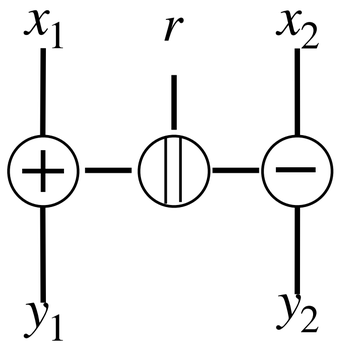
SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

How to concretely compute the ϵ ?



SimpleRefresh

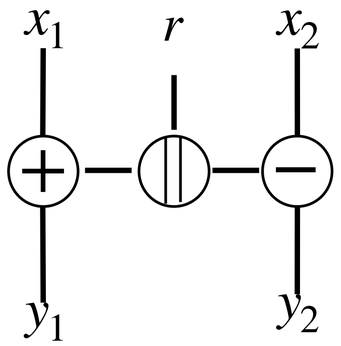
$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1 - (1-p)^3$)
And 2 output wires:
y_1
y_2

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

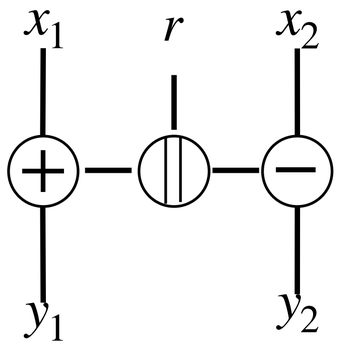
$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

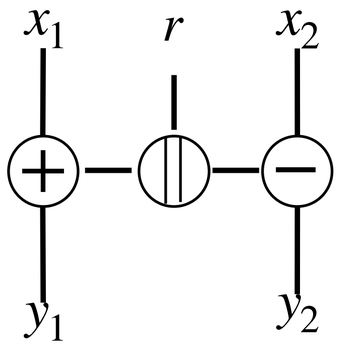
Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

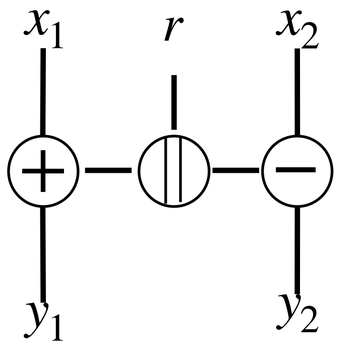
Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

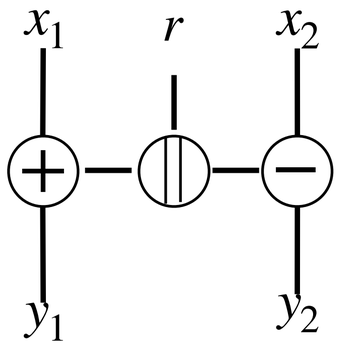
$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \emptyset$ with proba $(1-p)^5$

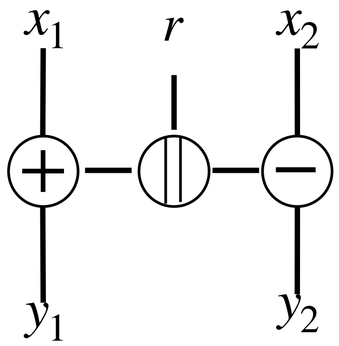
$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

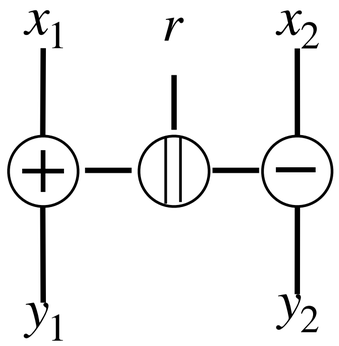
$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

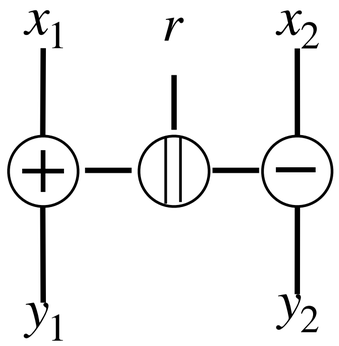
$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

Only 5 internal wires for
3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

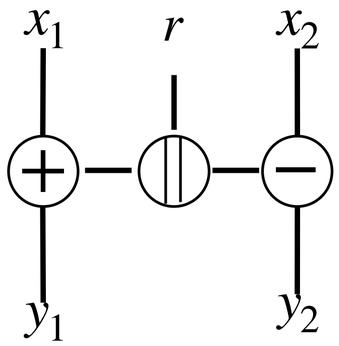
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

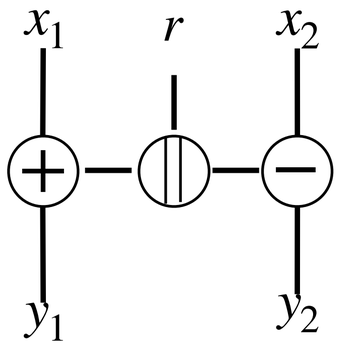
Probe Distribution Table (Crypto 2021)

	$\varnothing$
$\varnothing$	$(1-p)^2$
x_1	$p(1-p)$
x_2	$p(1-p)$
(x_1, x_2)	p^2

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

- $\mathscr{W} = \varnothing$ with proba $(1-p)^5$
- $\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$
- $\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$
- $\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$
- $\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$
- $\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$
- $\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$
- $\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$
$\varnothing$	$(1-p)^2$
x_1	$p(1-p)$
x_2	$p(1-p)$
(x_1, x_2)	p^2

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

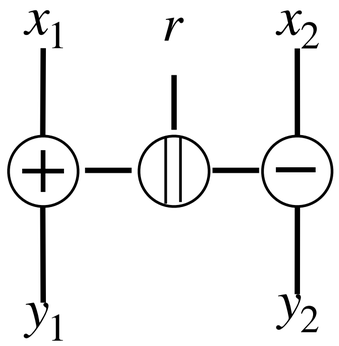
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$
$\varnothing$	$(1-p)^2$
x_1	$p(1-p)$
x_2	$p(1-p)$
(x_1, x_2)	p^2

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

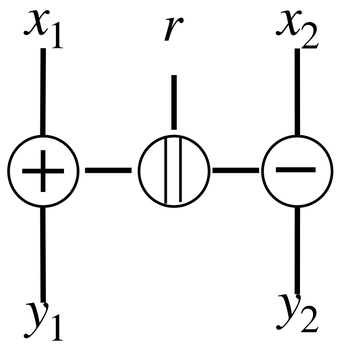
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$
$\varnothing$	$(1-p)^2$
x_1	$p(1-p)$
x_2	$p(1-p)$
(x_1, x_2)	p^2

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

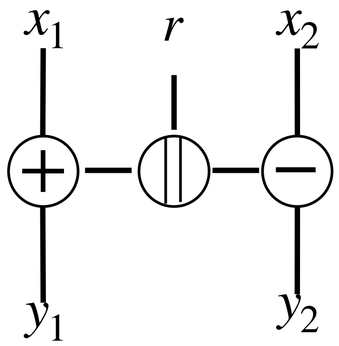
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$	y_1	y_2	(y_1, y_2)
$\varnothing$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

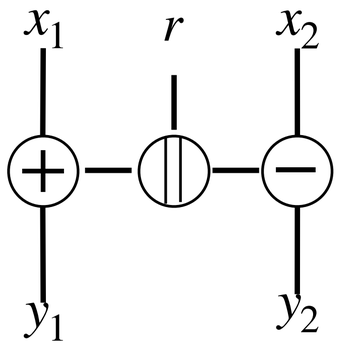
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \emptyset$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

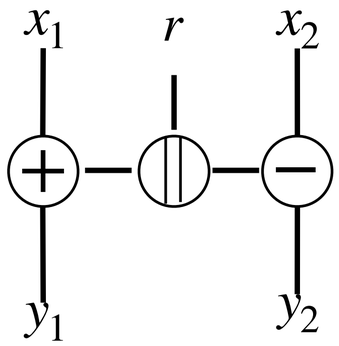
$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

Probe Distribution Table (Crypto 2021)

	$\emptyset$	y_1	y_2	(y_1, y_2)
$\emptyset$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$	y_1	y_2	(y_1, y_2)
$\varnothing$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

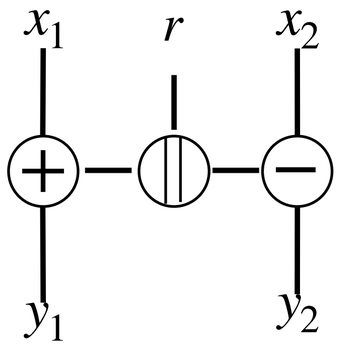
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathcal{W} = \emptyset$ with proba $(1-p)^5$

$\mathcal{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathcal{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathcal{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathcal{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathcal{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

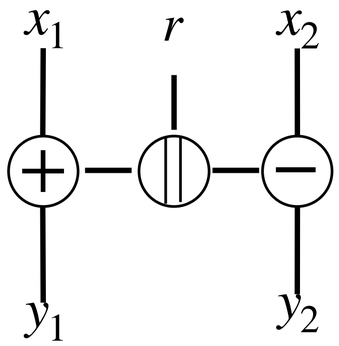
$\mathcal{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathcal{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

Probe Distribution Table (Crypto 2021)

	$\emptyset$	y_1	y_2	(y_1, y_2)
$\emptyset$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$	y_1	y_2	(y_1, y_2)
$\varnothing$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \varnothing$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

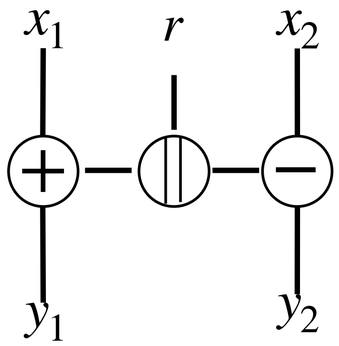
$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

How to concretely compute the ϵ ?



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Only 5 internal wires for 3 internal variables:
x_1 (proba. p)
x_2 (proba.p)
r (proba. $1-(1-p)^3$)
And 2 output wires:
y_1
y_2

$\mathscr{W} = \emptyset$ with proba $(1-p)^5$

$\mathscr{W} = \{x_1\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{x_2\}$ with proba $p(1-p)^4$

$\mathscr{W} = \{r\}$ with proba $\left(1-(1-p)^3\right)(1-p)^2$

$\mathscr{W} = \{x_1, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_2, r\}$ with proba $p(1-p)\left(1-(1-p)^3\right)$

$\mathscr{W} = \{x_1, x_2\}$ with proba $p^2(1-p)^3$

$\mathscr{W} = \{x_1, x_2, r\}$ with proba $p^2\left(1-(1-p)^3\right)$

Probe Distribution Table (Crypto 2021)

	$\emptyset$	y_1	y_2	(y_1, y_2)
$\emptyset$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

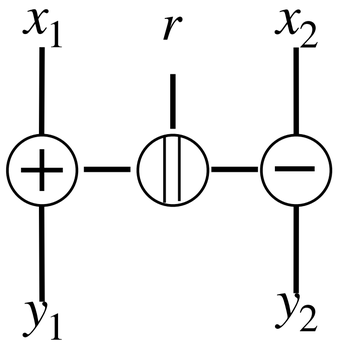
Cases that cannot be simulated (we need all the shares of x)

(p, ϵ, t)-random-probing composability

$\rightarrow$ $t = 1$
$\epsilon = p - p(1-p)^4$

This technique does not scale well
specially for large gadgets with many shares (PQC)

Using cardinal tables



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Cardinal Distribution Table

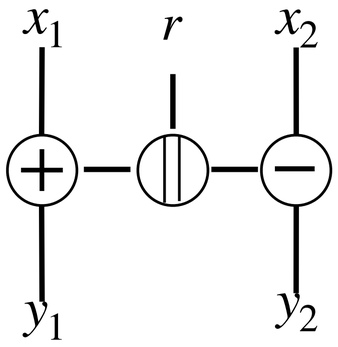
t_{in} \ t_{out}	0	1	2
0	$(1-p)^2$	$(1-p)^5$	0
1	$2p(1-p)$	$2(1-p)\cdot(1-(1-p)^4)$	0
2	p^2	$p-p(1-p)^4$	1

Probe Distribution Table (Crypto 2021)

	$\emptyset$	y_1	y_2	(y_1, y_2)
$\emptyset$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

[BRR25] S. Belaïd, M. Rivain and M. Rossi, *New Techniques for Random Probing Security and Application to Raccoon Signature Scheme* published in Eurocrypt 2025

Using cardinal tables



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\emptyset$	y_1	y_2	(y_1, y_2)
$\emptyset$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

Sum

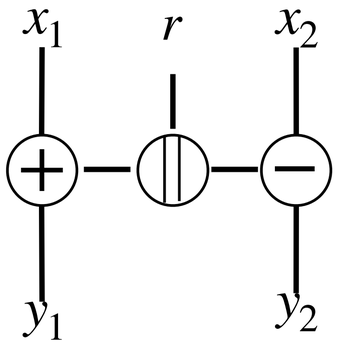
Max

Cardinal Distribution Table

t_{in} \ t_{out}	0	1	2
0	$(1-p)^2$	$(1-p)^5$	0
1	$2p(1-p)$	$2(1-p)\cdot(1-(1-p)^4)$	0
2	p^2	$p-p(1-p)^4$	1

[BRR25] S. Belaïd, M. Rivain and M. Rossi, *New Techniques for Random Probing Security and Application to Raccoon Signature Scheme* published in Eurocrypt 2025

Using cardinal tables



SimpleRefresh

$y_1 + y_2 = x_1 + x_2$

$r \leftarrow \$$
$y_1 \leftarrow x_1 + r$
$y_2 \leftarrow x_2 - r$

Probe Distribution Table (Crypto 2021)

	$\varnothing$	y_1	y_2	(y_1, y_2)
$\varnothing$	$(1-p)^2$	$(1-p)^5$	$(1-p)^5$	0
x_1	$p(1-p)$	$(1-p)\cdot(1-(1-p)^4)$	$p(1-p)^4$	0
x_2	$p(1-p)$	$p(1-p)^4$	$(1-p)\cdot(1-(1-p)^4)$	0
(x_1, x_2)	p^2	$p-p(1-p)^4$	$p-p(1-p)^4$	1

Sum

Max

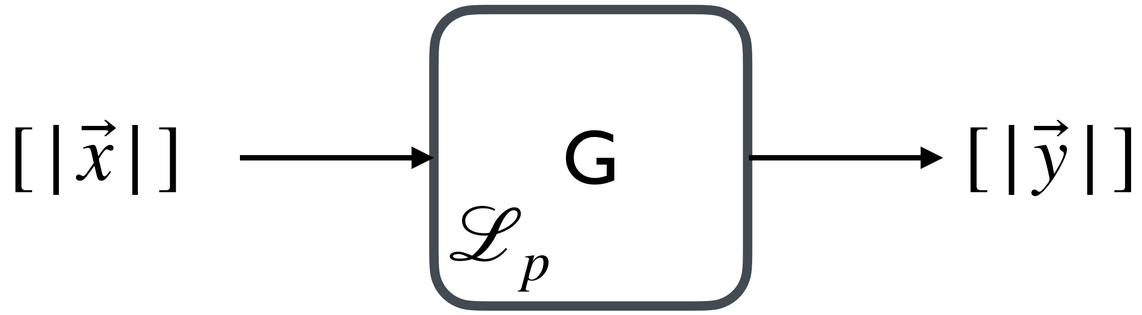
Cardinal Distribution Table

t_{in} \ t_{out}	0	1	2
0	$(1-p)^2$	$(1-p)^5$	0
1	$2p(1-p)$	$2(1-p)\cdot(1-(1-p)^4)$	0
2	p^2	$p-p(1-p)^4$	1

Probability envelopes: $\mathscr{E}_{t_{\text{out}}}(t_{\text{in}})$

[BRR25] S. Belaïd, M. Rivain and M. Rossi, *New Techniques for Random Probing Security and Application to Raccoon Signature Scheme* published in Eurocrypt 2025

Extensions of the Random Probing Security

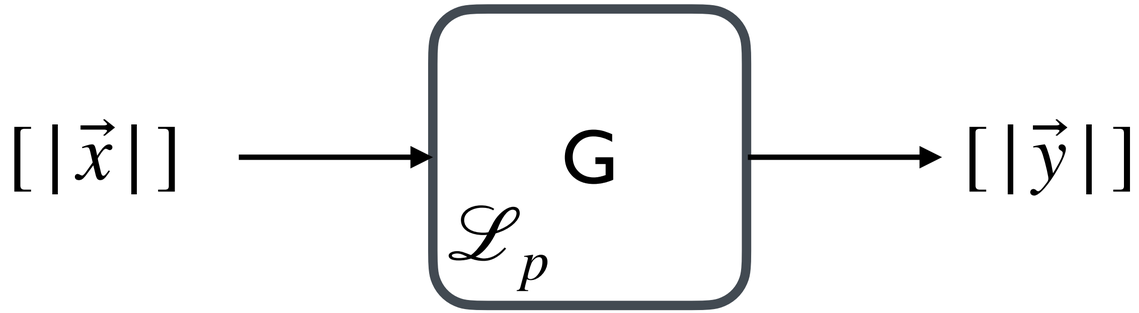


(p, t, ε)-RPC

$\mathbb{P}$(« more than t shares of each $[|x^i|]$ are required to simulate $\mathscr{L}_p$ and t output shares of each $[|y^i|]$ ») $\leq \varepsilon$

Extensions of the Random Probing Security

Cardinal Random Probing Composability (cardinal-RPC)



$(p, \overrightarrow{\mathscr{E}})$-cardinal RPC

$\mathbb{P}$(« exactly t_i^{in} shares of each $[|x^i|]$ are required to simulate $\mathscr{L}_p$ and t_j^{out} output shares of each $[|y^j|]$ ») $\leq \mathscr{E}_{\overrightarrow{t^{out}}}(\overrightarrow{t^{in}})$

$\forall (t_1^{in}, \ldots, t_\ell^{in}) \in [0,n]^\ell, \forall (t_1^{out}, \ldots, t_m^{out}) \in [0,n]^m,$

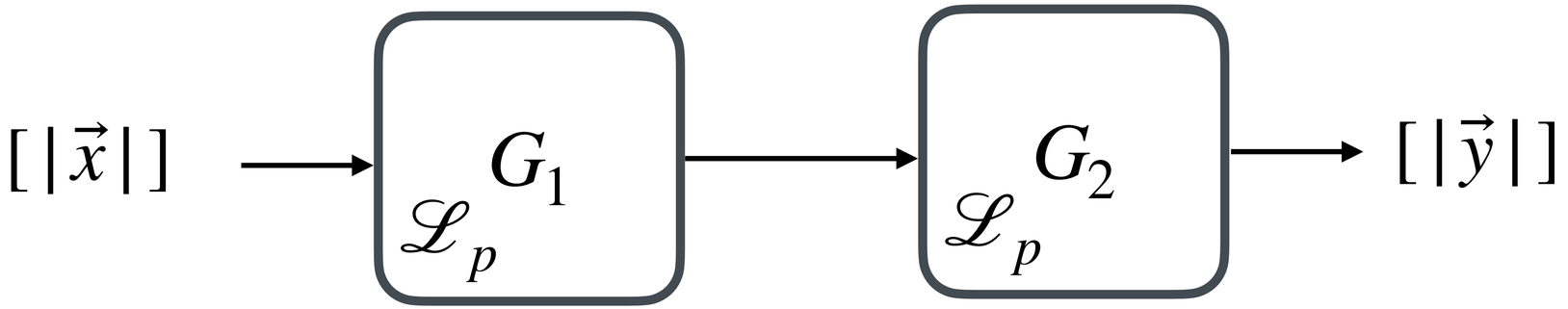
Extensions of the Random Probing Security

Cardinal Random Probing Composability
(cardinal-RPC)

Random Probing Security with Auxiliary Inputs and public Outputs
(RPS-AI-O)

Extensions of the Random Probing Security

Cardinal Random Probing Composability (cardinal-RPC)



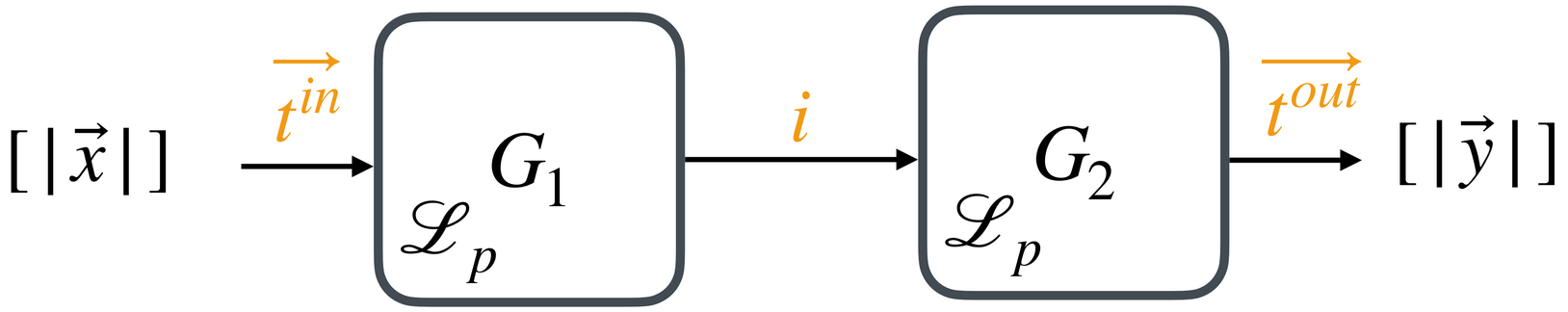
(p, t, ε)-RPC

(p, t, ε_3)-RPC advantage of the composition:

$$\varepsilon_3 \leq 1 - (1 - \varepsilon_1)(1 - \varepsilon_2)$$

Extensions of the Random Probing Security

Cardinal Random Probing Composability (cardinal-RPC)



$(p, \overrightarrow{\mathscr{E}})$-cardinal RPC

$(p, \overrightarrow{\mathscr{E}^3})$-cardinal RPC advantage of the composition:

$$\mathscr{E}^3_{\overrightarrow{t^{out}}}(\overrightarrow{t^{in}}) = \sum_{i=0}^{n} \mathscr{E}^1_i(\overrightarrow{t^{in}}) \cdot \mathscr{E}^2_{\overrightarrow{t^{out}}}(i)$$

Current state of the art

☑ Existing elementary gadgets proved (Cardinal or threshold)-RPC
- ➡ Addition
- ➡ Multiplication
- ➡ Copy
- ➡ Refresh

☑ Composition achievable by combining the enveloppes.

☑ Other composition techniques

[BFO23] Berti, F., Faust, S., Orlt, M. *Provable secure parallel gadgets*. TCHES 2023

[JMB24] V. Jahandideh, B. Mennink and L. Batina
An Algebraic Approach for Evaluating RandomProbing Security With Application to AES. TCHES 2024

Current state of the art

- ☑ Existing elementary gadgets proved (Cardinal or threshold)-RPC
  - ➡ Addition
  - ➡ Multiplication
  - ➡ Copy
  - ➡ Refresh
- ☑ Composition achievable by combining the enveloppes.
- ☑ Other composition techniques

Exciting work still lies ahead !

- ☐ More advanced gadgets
  - ➡ Mask conversions, comparisons
  - ➡ Sampling with specific distributions
- ☐ Optimized composition for tighter bounds
- ☐ Formal verification
- ☐ Efficient protected implementations

[BFO23] Berti, F., Faust, S., Orlt, M. *Provable secure parallel gadgets*. TCHES 2023

[JMB24] V. Jahandideh, B. Mennink and L. Batina
An Algebraic Approach for Evaluating RandomProbing Security With Application to AES. TCHES 2024

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

4) Ideas for future works

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

4) Ideas for future works

Raccoon Signature Scheme

KeyGen

1. Generate a large matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$
2. $[\,|s|\,] = (0,\ldots,0)$
3. Add noise to $[\,|s|\,]$
4. Compute $[\,|t|\,] = \mathbf{A} \cdot [\,|s|\,]$
5. Add noise to $[\,|t|\,]$
6. Decode $[\,|t|\,]$ to t
7. The verification key is $(\mathbf{A}, t)$
8. The signing key is $[\,|s|\,]$

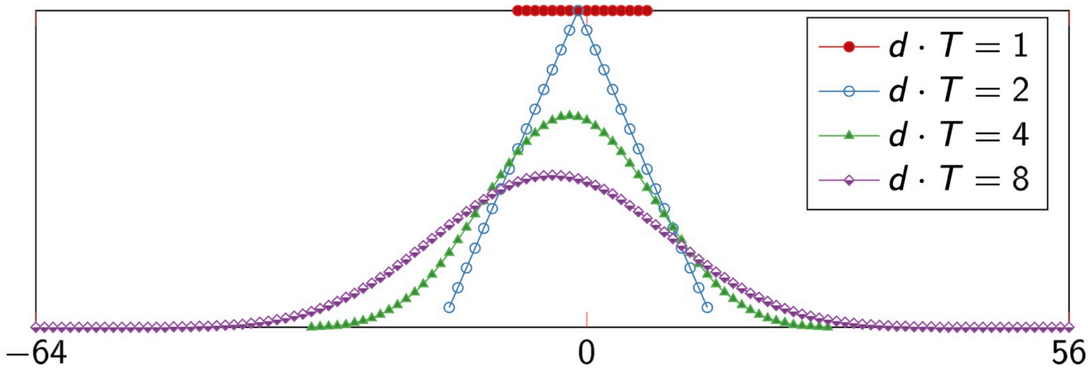
Signature

1. $[\,|r|\,] = \text{Refresh}(0,\ldots,0)$
2. Add noise to $[\,|r|\,]$
3. Compute the commitment $[\,|w|\,] = \mathbf{A} \cdot [\,|r|\,]$
4. Add noise to $[\,|w|\,]$
5. Decode $[\,|w|\,]$ to w
6. Compute the challenge $c = \text{H}(w, \text{msg}, \text{vk})$
7. Compute the response $[\,|z|\,] = [\,|s|\,] \cdot c + [\,|r|\,]$
8. Decode $[\,|z|\,]$ to z
   No Rejection Sampling
9. The signature is sig = (c, z)

« Add noise to »

For $i \in [0,T]$:

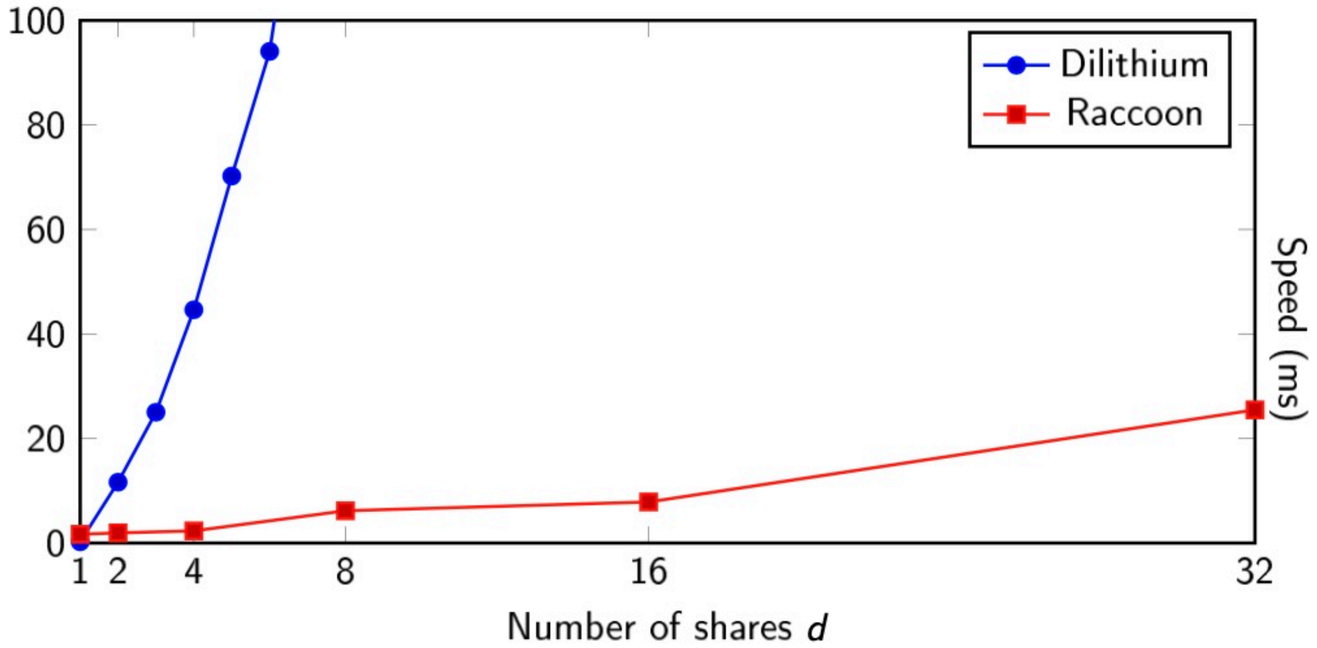
1. Sample d small uniforms randoms $[\,|r|\,] = (r_0, \ldots, r_{d-1})$
2. $[\,|x|\,] = [\,|x|\,] + [\,|r|\,]$
3. Refresh $[\,|x|\,]$



Distribution of the random that is added

[dPKPR24] R. del Pino, S. Katsumata, T. Prest and M. Rossi
Raccoon: A Masking-Friendly Signature Proven in the Probing Model. CRYPTO 2024

Objective of Raccoon



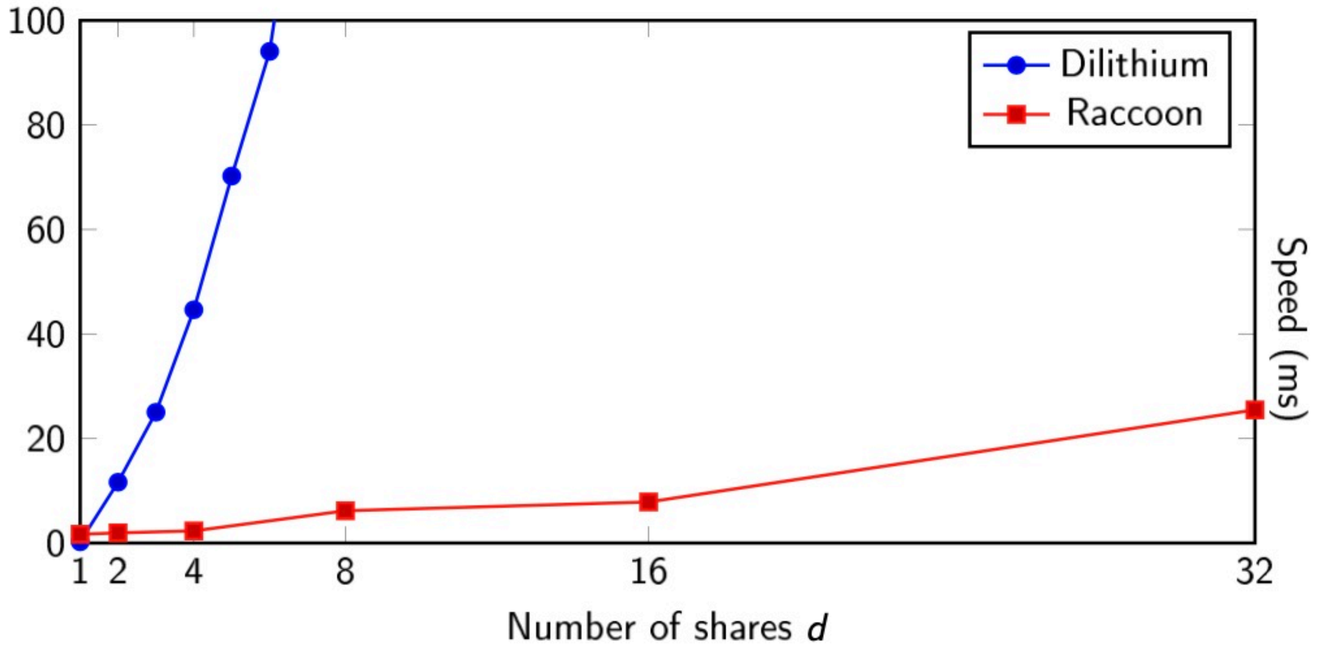
Raccoon 128-16

q	549824583172097
n	512
k	5
l	4
d	16
T	2

- ➡ Quasi-linear in the masking order
- ➡ Proof in the $(d-1)$-probing model
- ➡ Same assumptions as Dilithium/ML-DSA
- ➡ Simpler
- ➡ Same key size for the verification key
- ➡ When masked, orders of magnitude faster

Signatures $4\times$ larger

Objective of Raccoon



Raccoon 128-16

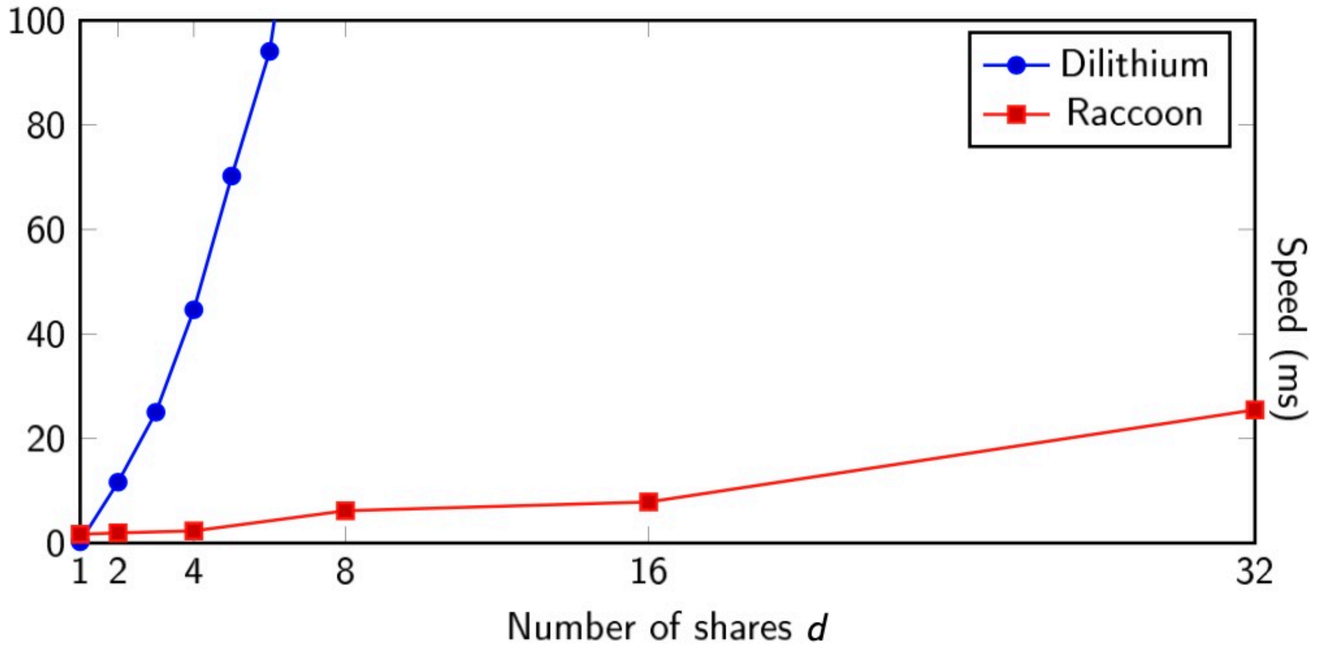
q	5498245831720 97
n	512
k	5
l	4
d	16
T	2

- ➡ Quasi-linear in the masking order
- ➡ Proof in the $(d-1)$-probing model
- ➡ Same assumptions as Dilithium/ML-DSA
- ➡ Simpler
- ➡ Same key size for the verification key
- ➡ When masked, orders of magnitude faster

Signatures $4\times$ larger

Not selected for NIST additional post-quantum signatures

Objective of Raccoon



Raccoon 128-16

q	5498245831720 97
n	512
k	5
l	4
d	16
T	2

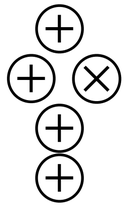
- ➡ Quasi-linear in the masking order
- ➡ Proof in the $(d-1)$-probing model
- ➡ Same assumptions as Dilithium/ML-DSA
- ➡ Simpler
- ➡ Same key size for the verification key
- ➡ When masked, orders of magnitude faster

Signatures $4\times$ larger

Not selected for NIST additional post-quantum signatures

But thresholdizable

Functionally

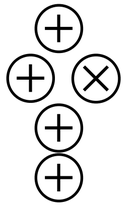


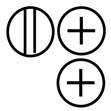
KeyGen

1. Generate a large matrix $\mathbf{A} \in \mathscr{R}_q^{k \times \ell}$
2. $[\![s]\!] = (0,\ldots,0)$
3. Add noise to $[\![s]\!]$
4. Compute $[\![t]\!] = \mathbf{A} \cdot [\![s]\!]$
5. Add noise to $[\![t]\!]$
6. Decode $[\![t]\!]$ to t
7. The verification key is $(\mathbf{A}, t)$
8. The signing key is $[\![s]\!]$

« Add noise to »

Add $d \cdot T$ small uniform randoms





Signature

1. $[\![r]\!] = \sout{\text{Refresh}}(0,\ldots,0)$
2. Add noise to $[\![r]\!]$
3. Compute the commitment $[\![w]\!] = \mathbf{A} \cdot [\![r]\!]$
4. Add noise to $[\![w]\!]$
5. Decode $[\![w]\!]$ to w
6. Compute the challenge $c = \mathsf{H}(w, \mathsf{msg}, \mathsf{vk})$
7. Compute the response $[\![z]\!] = [\![s]\!] \cdot c + [\![r]\!]$
8. Decode $[\![z]\!]$ to z
   No Rejection Sampling
9. The signature is $\mathsf{sig} = (\mathsf{c}, \mathsf{z})$

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

To be composable, they need to include some refreshes

Refresh  

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

$\oplus$ $\otimes$ $\mathbb{H}$

→

To be composable, they need to include some refreshes

Refresh → $\oplus$ $\otimes$ $\mathbb{H}$

→

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

$\oplus$ $\otimes$ $\mathbb{I}$

→

To be composable, they need to include some refreshes

Refresh → $\oplus$ $\otimes$ $\mathbb{I}$

→

➡ Raccoon's refresh is not (yet?) proved RPC (cardinal or threshold).
We could not prove the **random probing composability of the quasilinear refresh.**

[BCPZ16] A. Battistello, J.-S. Coron, E. Prouff, and R. Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. CHES 2016

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

$\oplus$ $\otimes$ $\mathbb{H}$

→

To be composable, they need to include some refreshes

Refresh → $\oplus$ $\otimes$ $\mathbb{H}$

→

➡ Raccoon's refresh is not (yet?) proved RPC (cardinal or threshold).
We could not prove the **random probing composability of the quasilinear refresh.**

The probability computation rapidly explodes if there are too many dependencies in the intermediate variables.

[BCPZ16] A. Battistello, J.-S. Coron, E. Prouff, and R. Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. CHES 2016

Design rationale of our new refresh

Composable (cardinal or threshold RPC) elementary gates are needed

$\oplus$ $\otimes$ $\mathbb{H}$

→

To be composable, they need to include some refreshes

Refresh → $\oplus$ $\otimes$ $\mathbb{H}$

→

➡ Raccoon's refresh is not (yet?) proved RPC (cardinal or threshold).
We could not prove the **random probing composability of the quasilinear refresh.**

The probability computation rapidly explodes if there are too many dependencies in the intermediate variables.

Random-probing-friendly feature

The gadget should contain as much independent random as possible (including the indexes of the shares that are touched)

[BCPZ16] A. Battistello, J.-S. Coron, E. Prouff, and R. Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. CHES 2016

New Random Probing Composable Refresh

$$[\,|z|\,] = (0,0,0,0,0,0,0,0)$$

New Random Probing Composable Refresh

$$[\,|z|\,] = (0,0,0,0,0,0,0,0)$$

1st iteration

$$r_1 \leftarrow \$, (i_1, j_1) \leftarrow \$ \qquad [(i_1, j_1) = (3,7)]$$

$$[\,|z|\,] = (0,0,r_1,0,0,0,-r_1,0)$$

New Random Probing Composable Refresh

$$[|z|] = (0,0,0,0,0,0,0,0)$$

1st iteration

$$r_1 \leftarrow \$, (i_1, j_1) \leftarrow \$ \qquad [(i_1, j_1) = (3,7)]$$

$$[|z|] = (0,0,r_1,0,0,0,-r_1,0)$$

2nd iteration

$$r_2 \leftarrow \$, (i_2, j_2) \leftarrow \$ \qquad [(i_2, j_2) = (1,8)]$$

$$[|z|] = (r_2,0,r_1,0,0,0,-r_1,-r_2)$$

New Random Probing Composable Refresh

$$[\,|z|\,] = (0,0,0,0,0,0,0,0)$$

1st iteration

$$r_1 \leftarrow \$, (i_1, j_1) \leftarrow \$ \qquad [(i_1, j_1) = (3,7)]$$

$$[\,|z|\,] = (0,0,r_1,0,0,0,-r_1,0)$$

2nd iteration

$$r_2 \leftarrow \$, (i_2, j_2) \leftarrow \$ \qquad [(i_2, j_2) = (1,8)]$$

$$[\,|z|\,] = (r_2,0,r_1,0,0,0,-r_1,-r_2)$$

3rd iteration

$$r_3 \leftarrow \$, (i_3, j_3) \leftarrow \$ \qquad [(i_3, j_3) = (2,3)]$$

$$[\,|z|\,] = (r_2, r_3, r_1 - r_3,0,0,0,-r_1,-r_2)$$

New Random Probing Refresh

Algorithm 1 $\mathsf{RPZeroEnc}^{n,\gamma}$

Output: $[\![z]\!] = (z_1, \ldots, z_n) \in \mathbb{K}^n$ such that $z_1 + \cdots + z_n = 0$

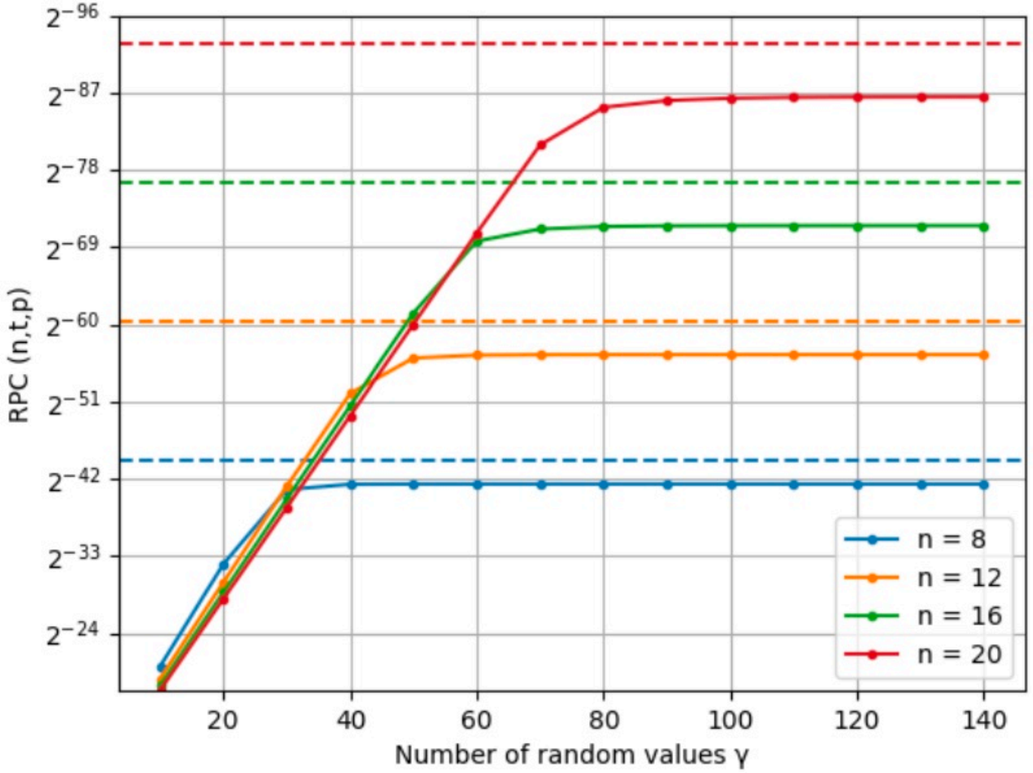
1. $[\![z]\!] = (0, 0, \ldots, 0)$ $\{n$ zeros$\}$
2. **for** $i = 1$ to γ **do**
3. Select two distinct indices $i_1, i_2 \in [1, n]\}$ uniformly at random
4. $r \xleftarrow{\$} \mathbb{K}$
5. $z_{i_1} \leftarrow z_{i_1} + r$
6. $z_{i_2} \leftarrow z_{i_2} - r$
7. **end for**
8. **return** $[\![z]\!]$

Algorithm 2 $\mathsf{RPRefresh}^{\gamma}([\![a]\!])$

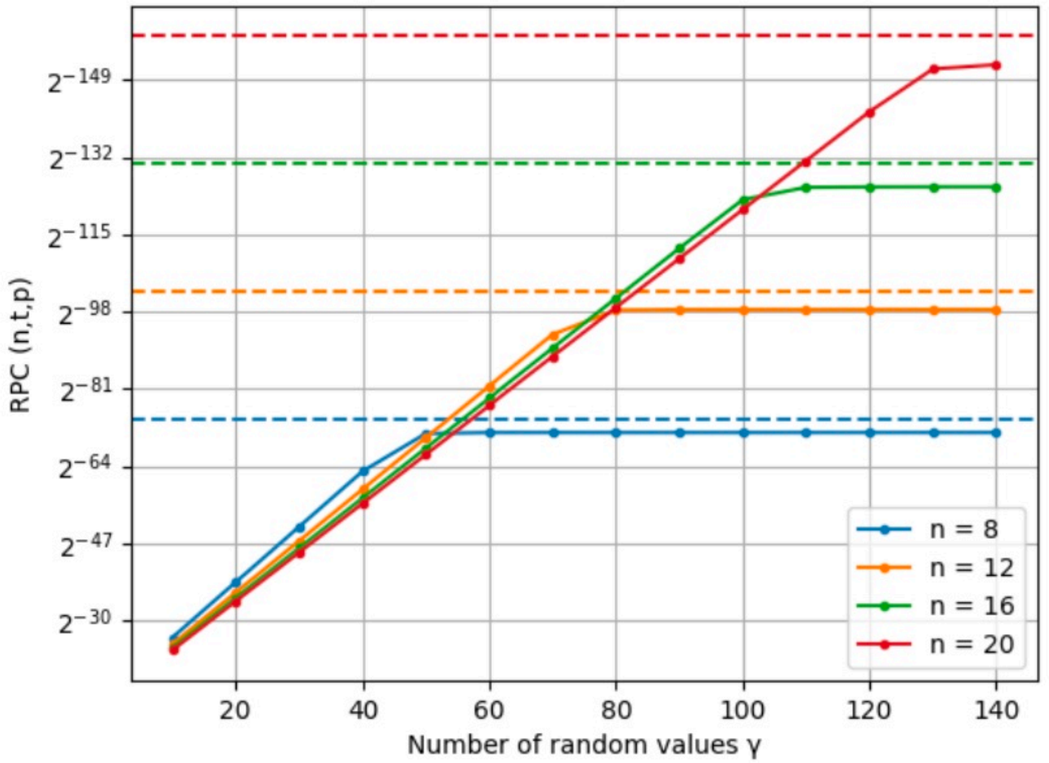
Input: $[\![a]\!] = (a_1, \ldots, a_n) \in \mathbb{K}^n$
Output: $[\![s]\!] = (s_1, \ldots, s_n) \in \mathbb{K}^n$ such that $s_1 + \cdots + s_n = a_1 + \cdots + a_n$

1. $[\![z]\!] \leftarrow \mathsf{RPZeroEnc}^{n,\gamma}()$
2. **for** $i = 1$ to n **do**
3. $s_i \leftarrow a_i + z_i$
4. **end for**
5. **return** $[\![s]\!]$

New Random Probing Composable Refresh



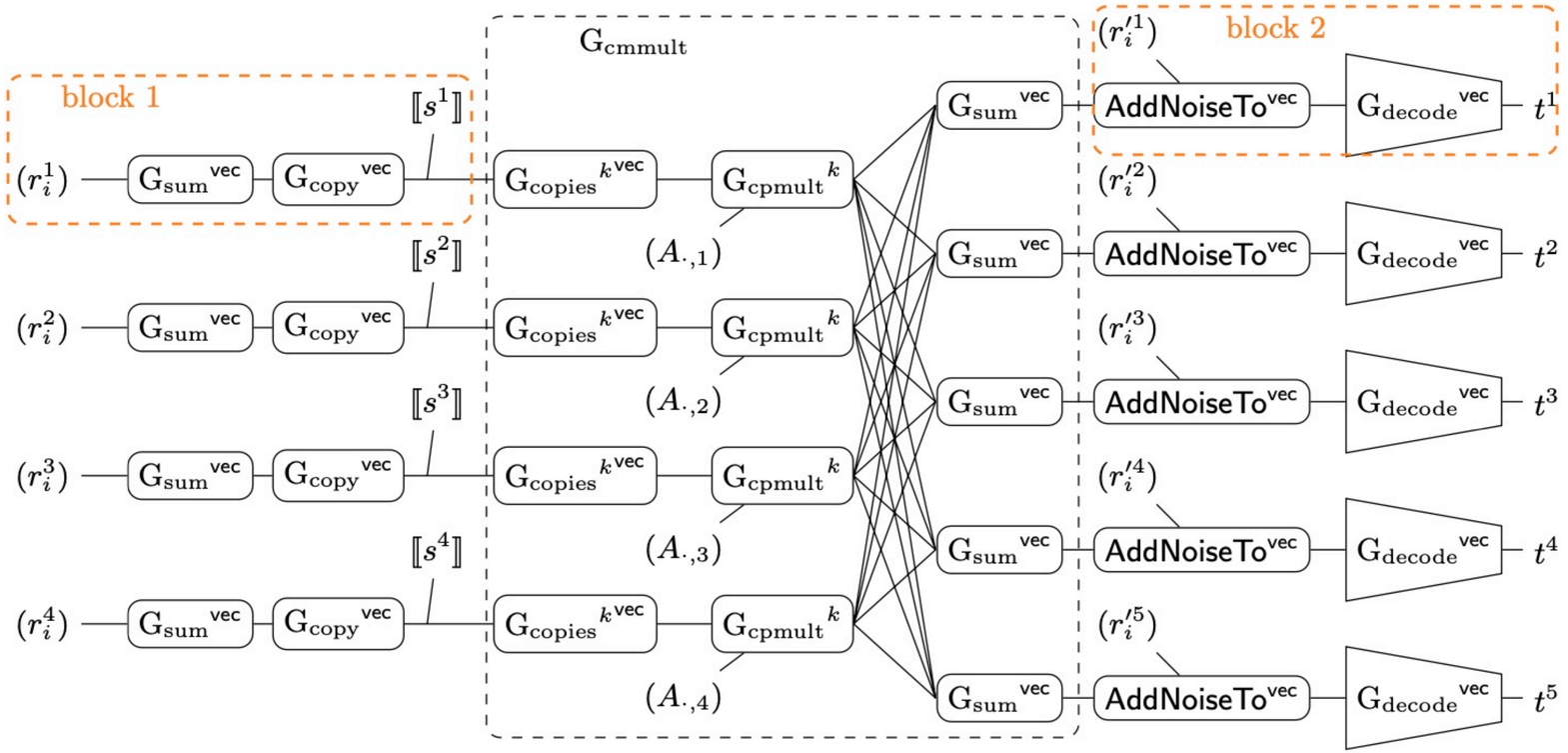
(a) $p = 2^{-10}$



(b) $p = 2^{-16}$

RPC-AI of RPRefresh from cardinal-RPC ($t = n/2$)

Key Generation of Raccoon with random probing secure gadgets



Raccoon 128-16:

- $n = 16$ shares
- 32 (r_i) or (r_i') per vector and polynomial coefficient
- EUF-CMA secure even if 15 values of each (r_i) or (r_i') leak

Random Probing Secure version of Raccoon

	Key Generation		Signature	
	Original	New Gadgets	Original	New Gadgets
# shares	16	16	16	16
# additions	8.49e7	1.82e9	1.02e8	3.44e9
# linear mult.	8.39e7	8.39e7	1.01e8	1.01e8
# randoms	3.60e5	6.57e8	5.57e5	1.42e9
Security RPS/C	1	2^{-132}	1	2^{-130}

Raccoon 128-16:

- $n = 16$ shares
- 32 (r_i) or (r'_i) per vector and polynomial coefficient
- EUF-CMA secure even if 15 values of each (r_i) or (r'_i) leak
- $p = 2^{-24}$

Random Probing Secure version of Raccoon

	Key Generation			Signature		
	Original		New Gadgets	Original		New Gadgets
# shares	16		16	16		16
# additions	8.49e7	× 20	1.82e9	1.02e8	× 30	3.44e9
# linear mult.	8.39e7	× 1	8.39e7	1.01e8	× 1	1.01e8
# randoms	3.60e5	× 2000	6.57e8	5.57e5	× 2500	1.42e9
Security RPS/C	1		2^{-132}	1		2^{-130}

Raccoon 128-16:

- $n = 16$ shares
- 32 (r_i) or (r'_i) per vector and polynomial coefficient
- EUF-CMA secure even if 15 values of each (r_i) or (r'_i) leak
- $p = 2^{-24}$

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

4) Ideas for future works

1) The relevance of the random probing model

2) The challenges: scaling up

3) Random-probing Raccoon

4) Ideas for future works

Completing the toolbox

- ➡ All gadgets can be accessed with the elementary gates but dedicated designs allow for better performance.
- ➡ Proofs are more difficult than with t-probing for large gadgets: a lot of cases to account for.
- ➡ A technique could be to look at small dedicated gadgets form PQC like the quasilinear refresh and the secadd and try to separate it in 'independent steps' and add randomness.

Completing the toolbox

- ➡ All gadgets can be accessed with the elementary gates but dedicated designs allow for better performance.
- ➡ Proofs are more difficult than with t-probing for large gadgets: a lot of cases to account for.
- ➡ A technique could be to look at small dedicated gadgets form PQC like the quasilinear refresh and the secadd and try to separate it in 'independent steps' and add randomness.

Idea behind secadd

$$x + y = x \oplus y \oplus c$$

Where $c^{(0)} = 0$ and $\forall i \geq 1, c^{(i)} = (c^{(i)} \,\&\, x^{(i)}) \oplus (c^{(i)} \,\&\, y^{(i)}) \oplus (x^{(i)} \,\&\, y^{(i)})$

Improving composition

- ➡ Several composition techniques already exists with very different ideas.
- ➡ Compare them, find links and look at the best way of composing random probing gadgets.
- ➡ Current composition techniques lack a lot of tightness.

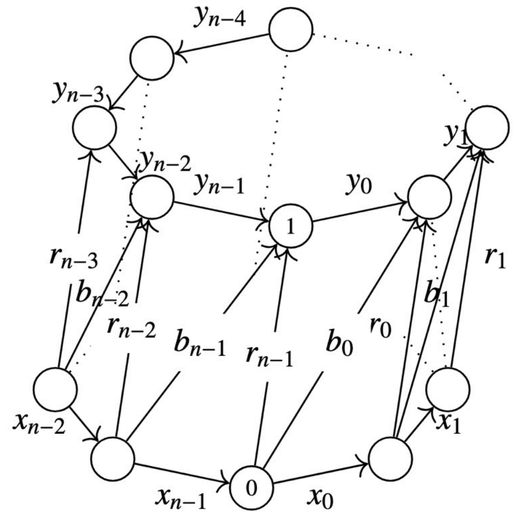


Figure extracted from **[BFO23]**

[BCPRT20] 8. Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R. *Random probing security: Verification, composition, expansion and new constructions.* CRYPTO 2020

[BFO23] Berti, F., Faust, S., Orlt, M. *Provable secure parallel gadgets*. TCHES 2023

[DFZ19] S. Dziembowski, S. Faust, K. Zebrowski
Simple refreshing in the noisy leakage model. ASIACRYPT 2019

[JMB24] V. Jahandideh, B. Mennink and L. Batina
An Algebraic Approach for Evaluating Random Probing Security With Application to AES. TCHES 2024

Thank you