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

Journées du GDR-Sécurité, Caen, 24/06/2025

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

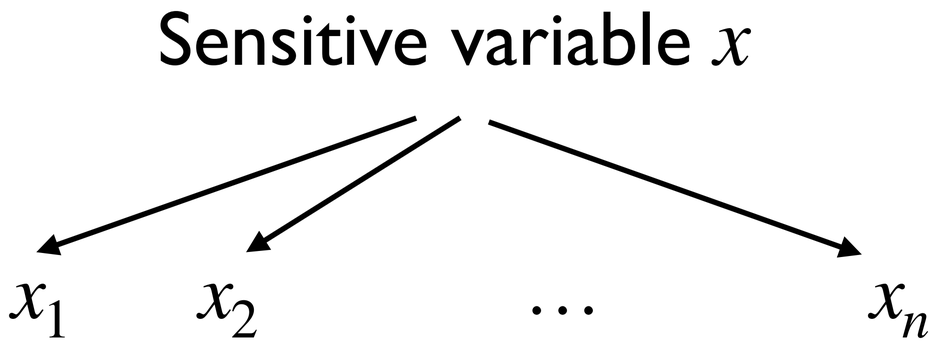
3) Random-probing Raccoon

1) The relevance of the random probing model

2) Scaling up is a big challenge

3) Random-probing Raccoon

Masking



Each strict subset of $(x_i)_{1 \leq i \leq n}$ 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

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

Masking in Practice

- Masking linear operations

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

$$z \leftarrow x + y$$

Masking in Practice

- Masking linear operations

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

$$\mathbf{z} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

$$z \leftarrow x + y$$

Masking in Practice

- Masking linear operations

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

$$\mathbf{z} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

$$z \leftarrow x + y$$

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

$$x = x_1 + x_2$$
$$y = y_1 + y_2$$

Masking in Practice

- Masking linear operations

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

$$\mathbf{z} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

$$z \leftarrow x + y$$

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

$$x = x_1 + x_2 \qquad z_1 \leftarrow x_1 y_1 + x_1 y_2$$
$$y = y_1 + y_2 \qquad z_2 \leftarrow x_2 y_2 + x_2 y_1$$

Masking in Practice

- Masking linear operations

$$x = x_1 + x_2 + \ldots + x_n$$
$$y = y_1 + y_2 + \ldots + y_n$$

$$\mathbf{z} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

$$z \leftarrow x + y$$

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

$$x = x_1 + x_2 \qquad z_1 \leftarrow x_1 y_1 + r + x_1 y_2$$
$$y = y_1 + y_2 \qquad z_2 \leftarrow x_2 y_2 - r + x_2 y_1$$

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

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

$r \leftarrow \$$

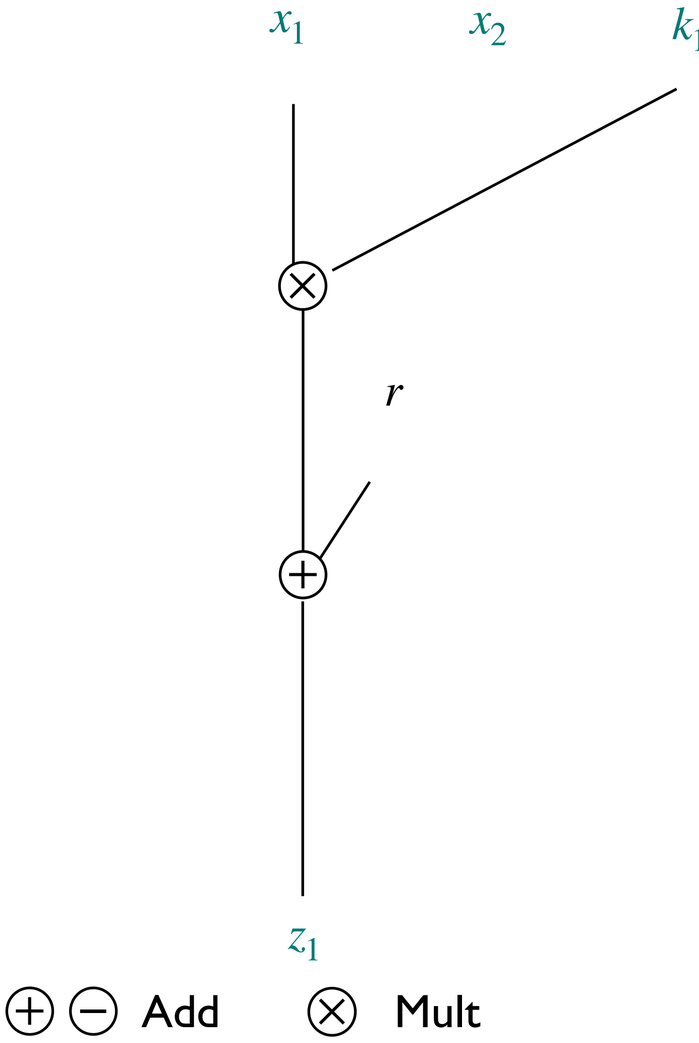
$z_1 \leftarrow x_1 k_1 + r$

$r' \leftarrow x_1 k_2 - r$

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

$z_2 \leftarrow r'' + x_2 k_2$

From a gadget to a circuit

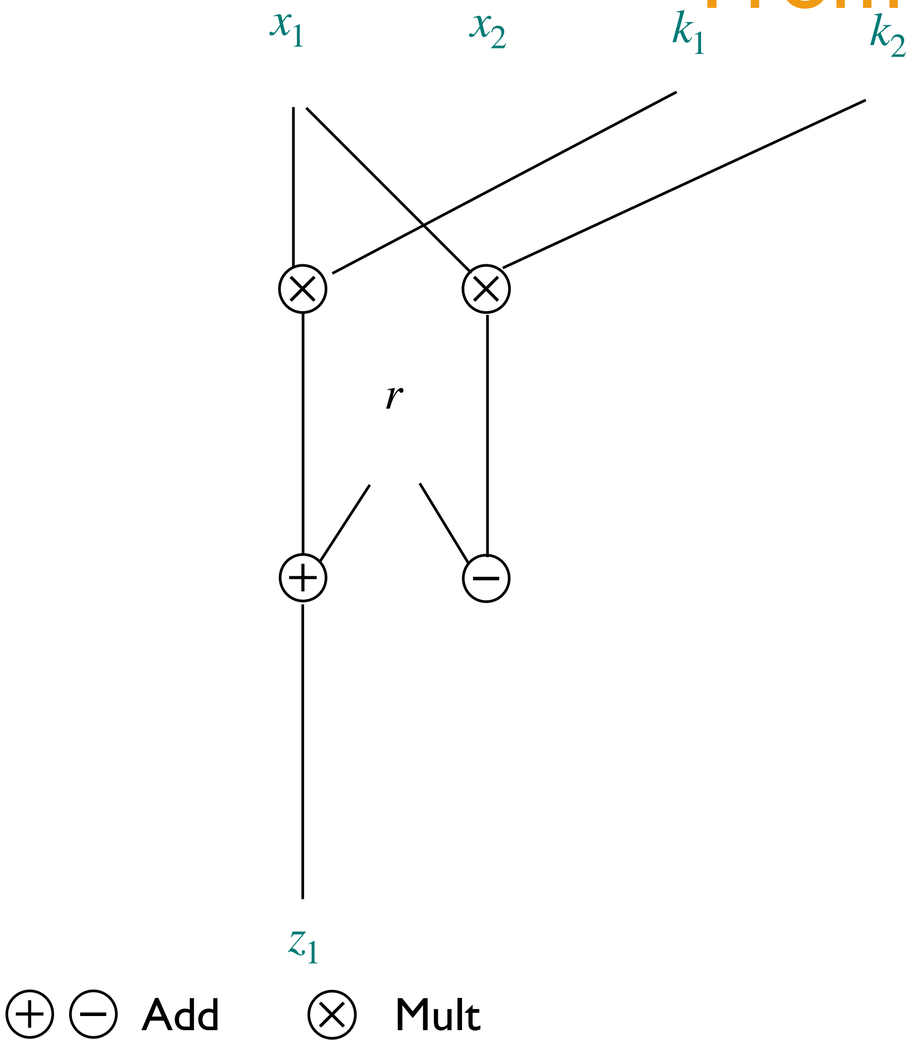


A Multiplication gadget

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

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

From a gadget to a circuit

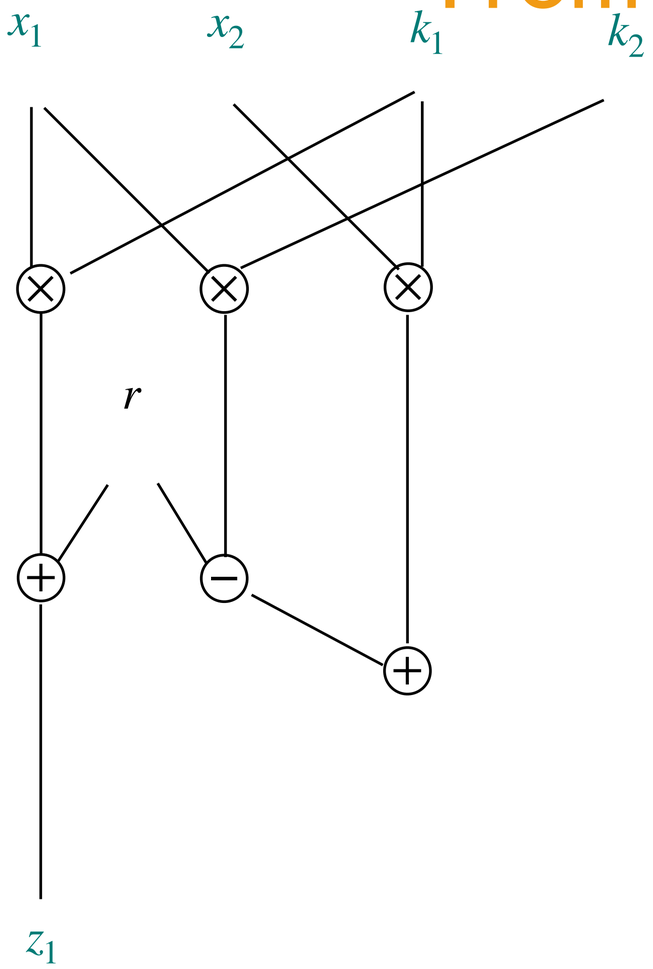


A Multiplication gadget

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

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

From a gadget to a circuit



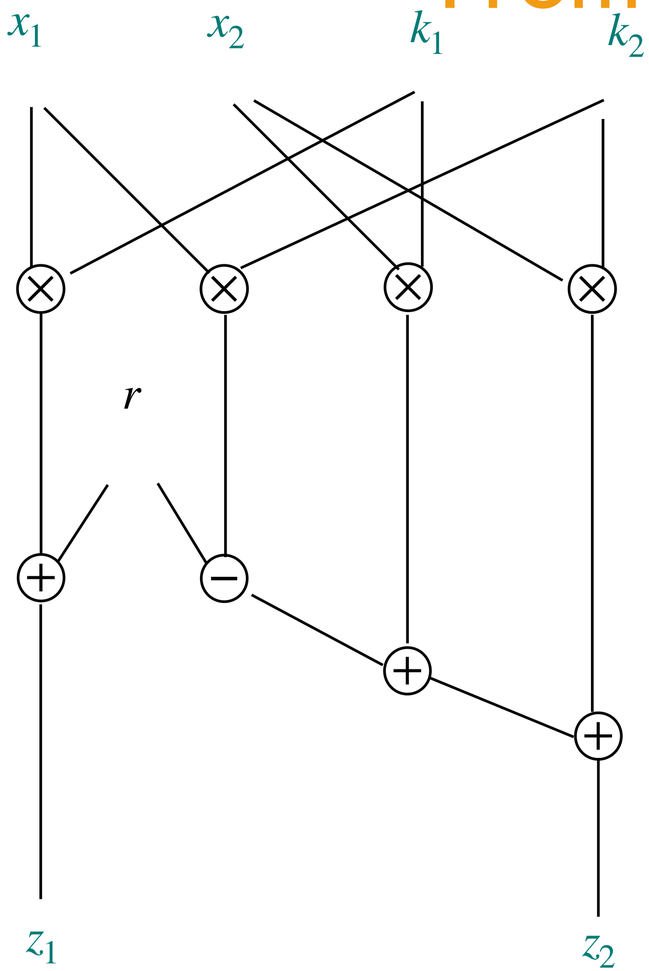
⊕ ⊖ Add ⊗ Mult

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



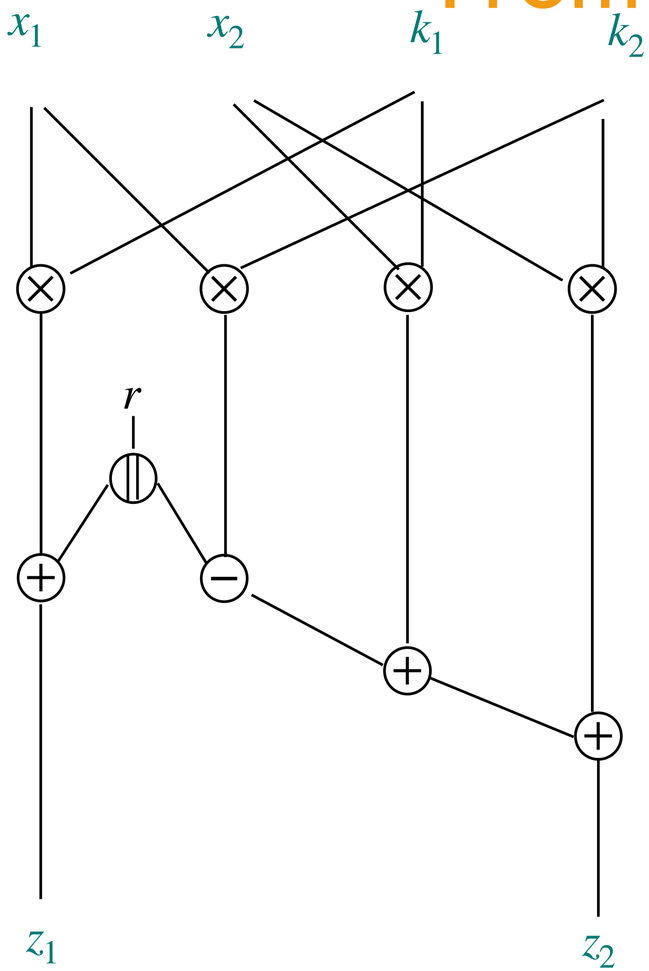
⊕ ⊖ Add ⊗ Mult

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

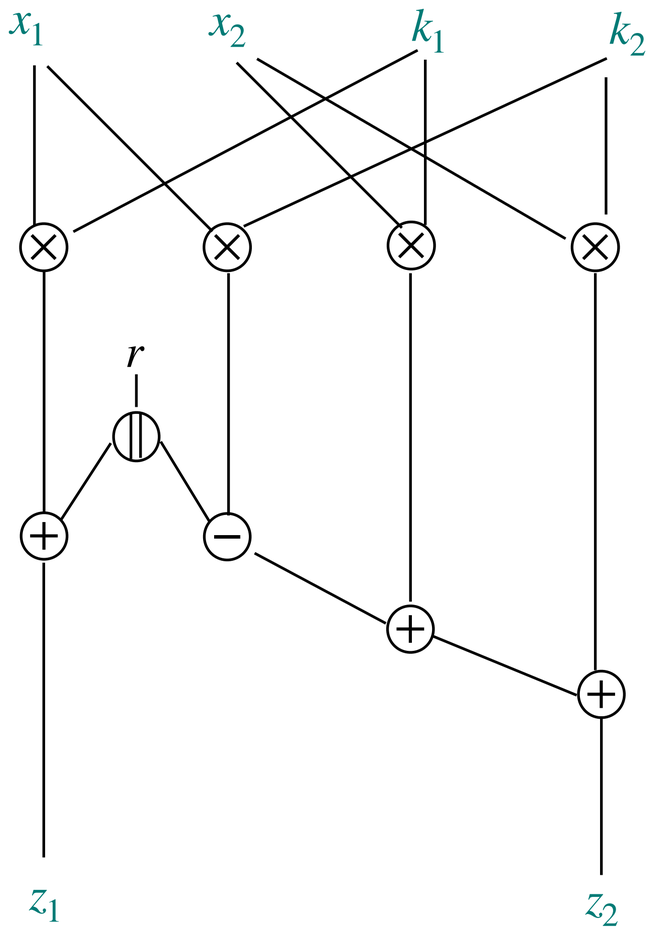
A Multiplication gadget

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

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

Leakage Models

Attacker view



A Multiplication gadget

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

$$r \leftarrow \$$$

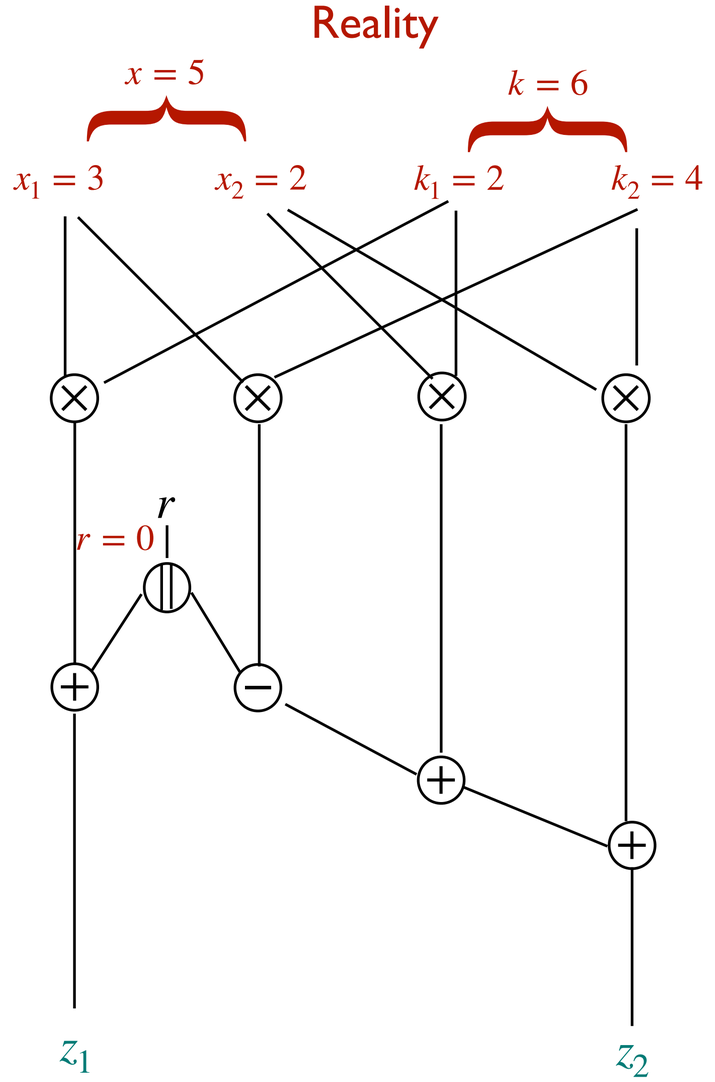
$$z_1 \leftarrow x_1 k_1 + r$$

$$r' \leftarrow x_1 k_2 - r$$

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

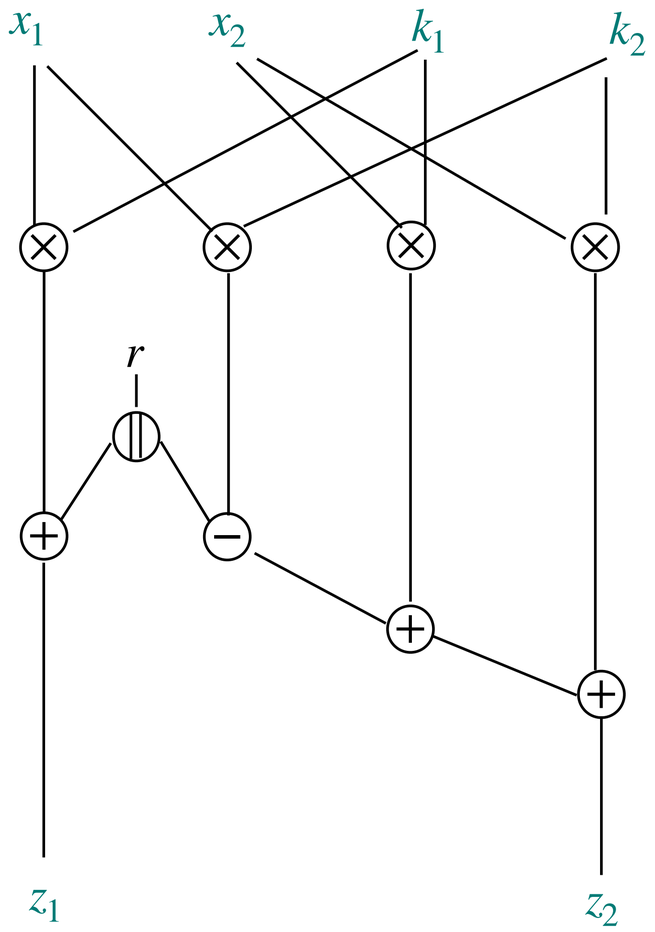
$$z_2 \leftarrow r'' + x_2 k_2$$

Reality



Leakage Models

Attacker view

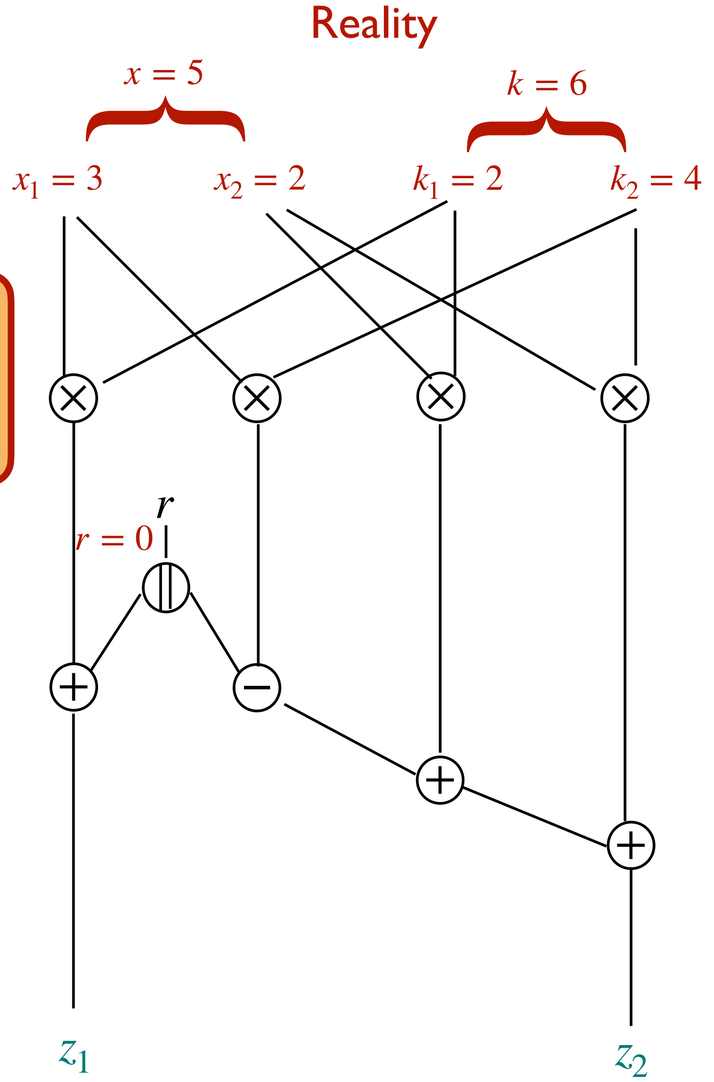


Attacker model

The attacker ← circuit + leakage

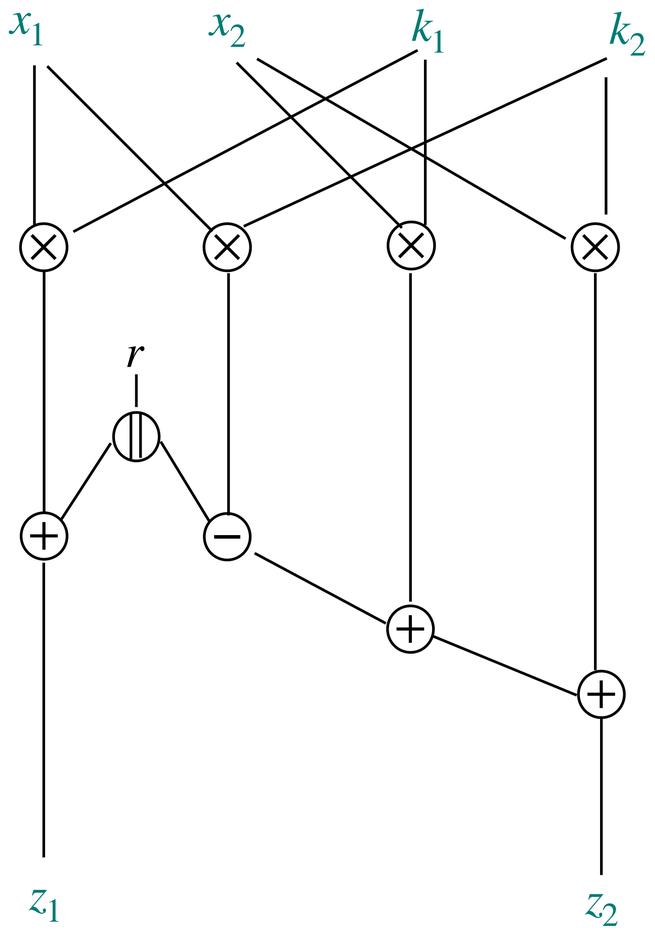
He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

Reality

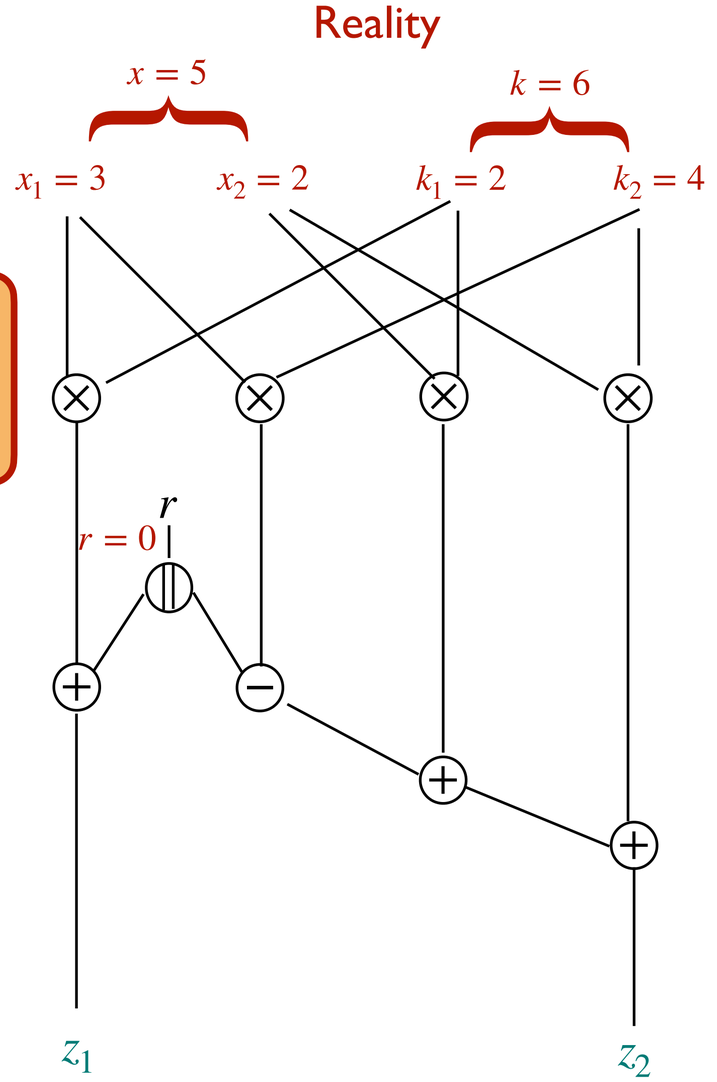


Leakage Models

Attacker view



Reality

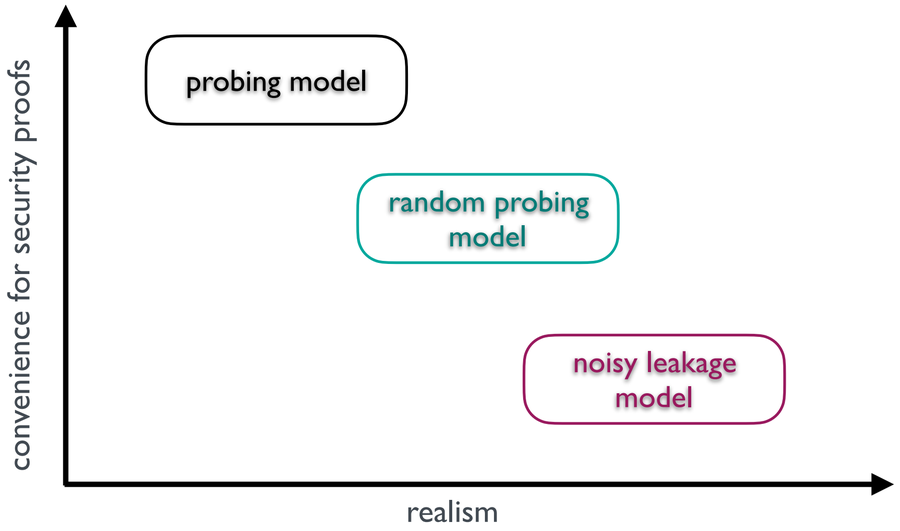


Attacker model

The attacker ← circuit + leakage

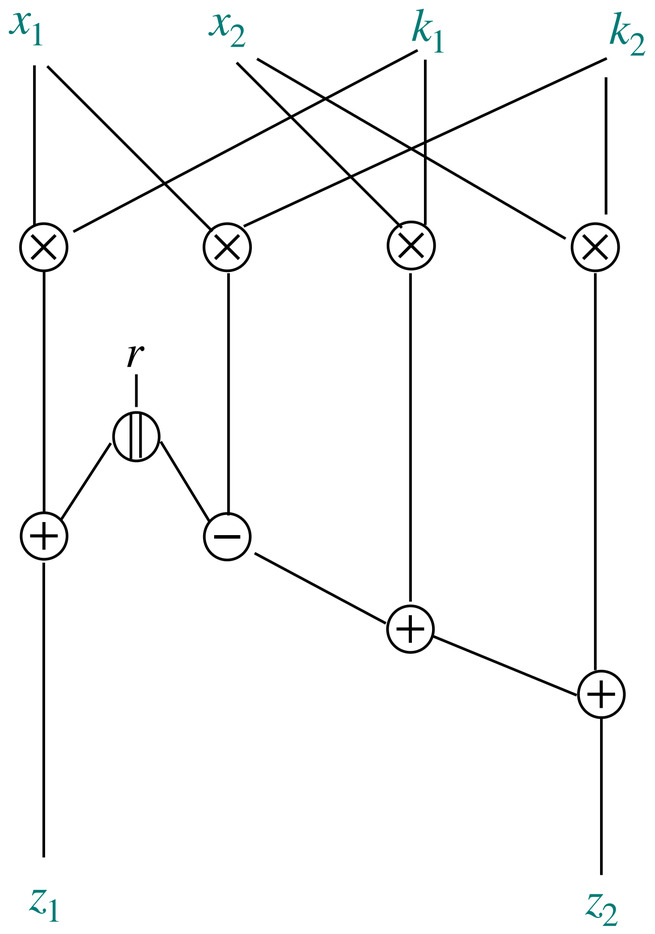
He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

3 flavours

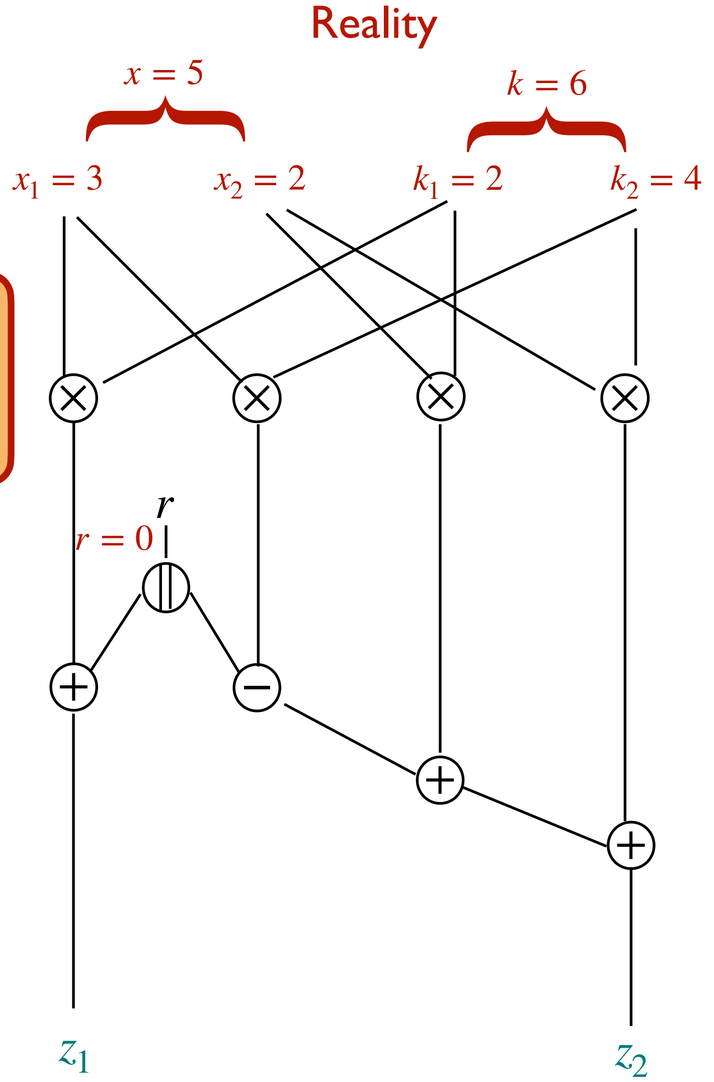


Leakage Models

Attacker view



Reality

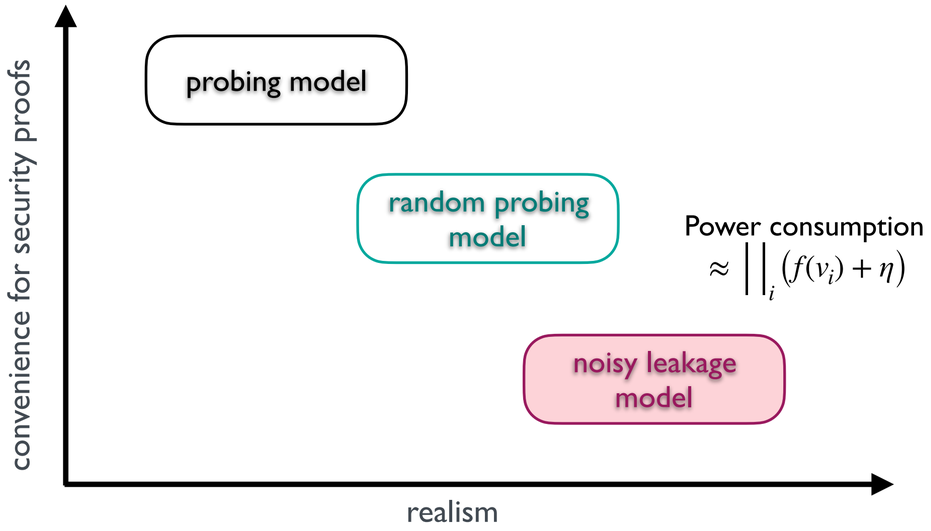


Attacker model

The attacker ← circuit + leakage

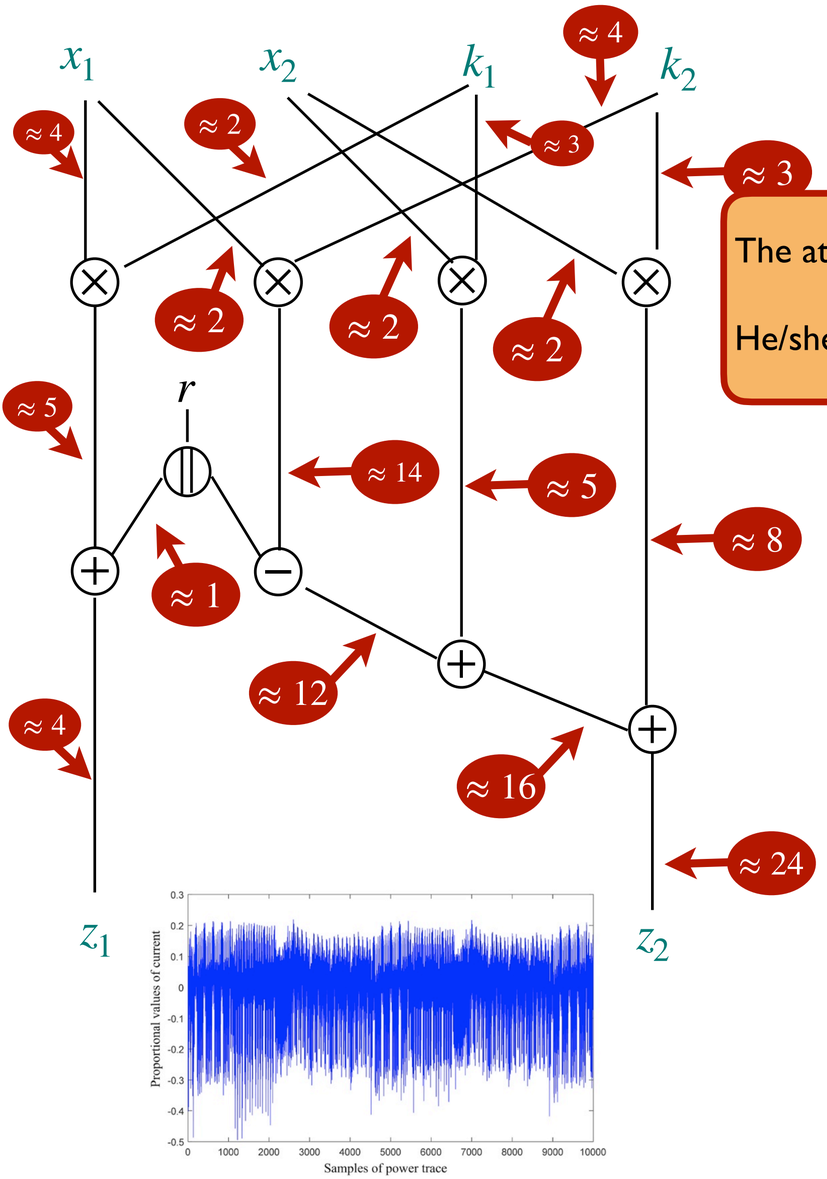
He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

3 flavours



Leakage Models

Attacker view

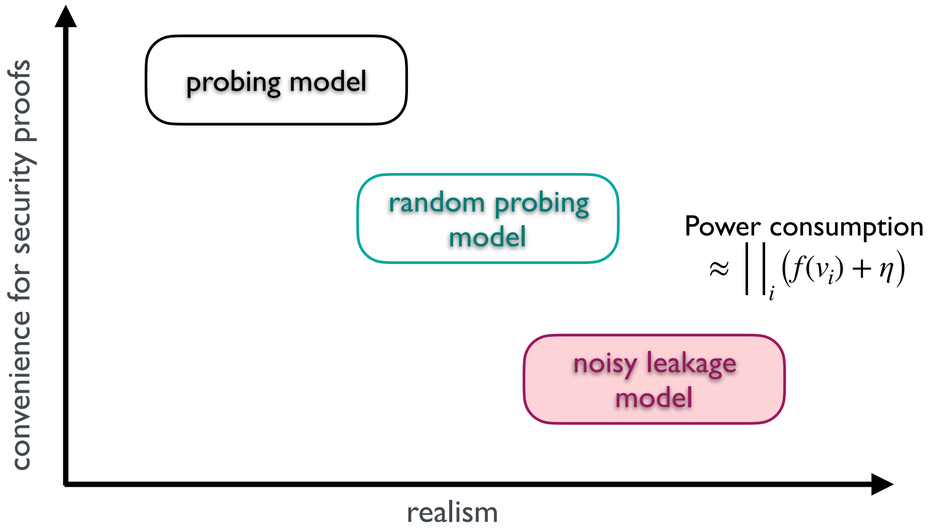


Attacker model

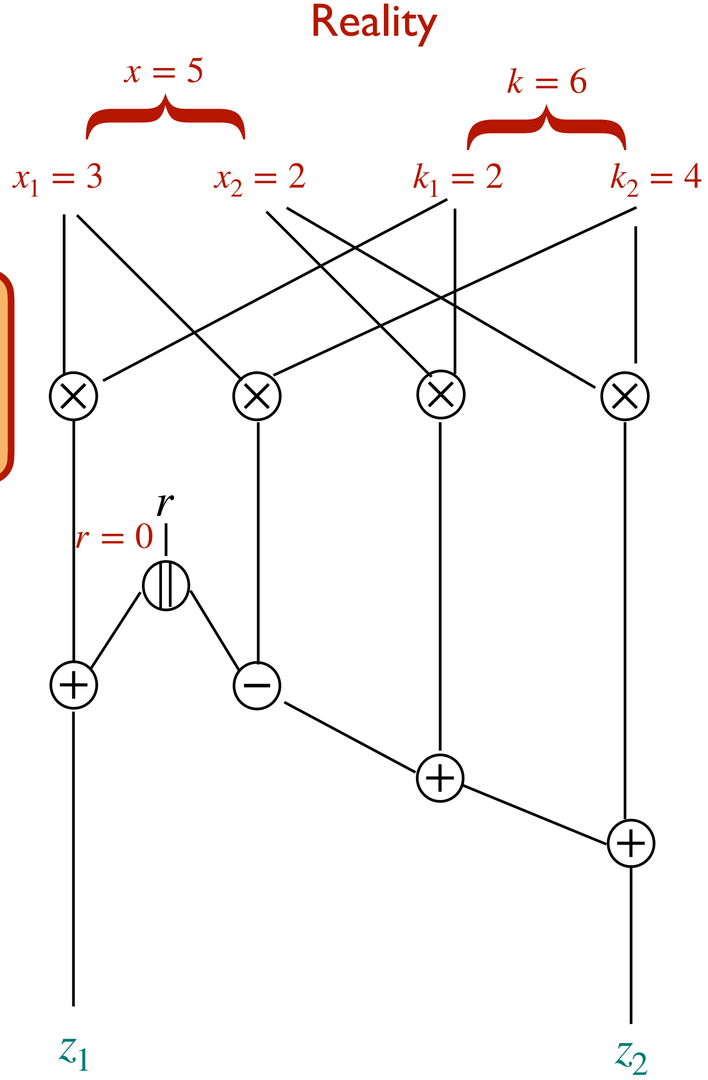
The attacker ← circuit + leakage

He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

3 flavours

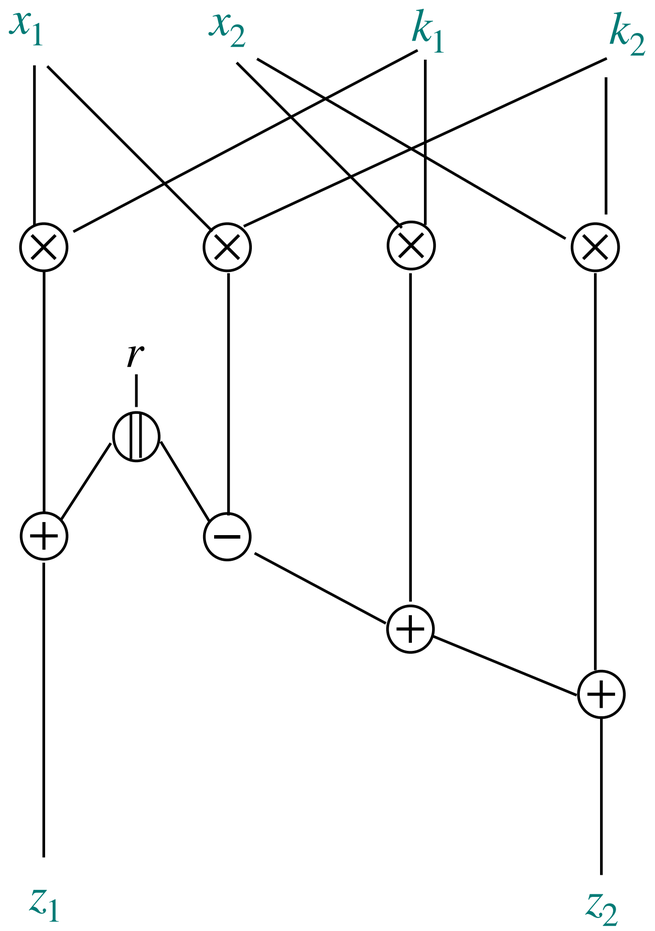


Reality

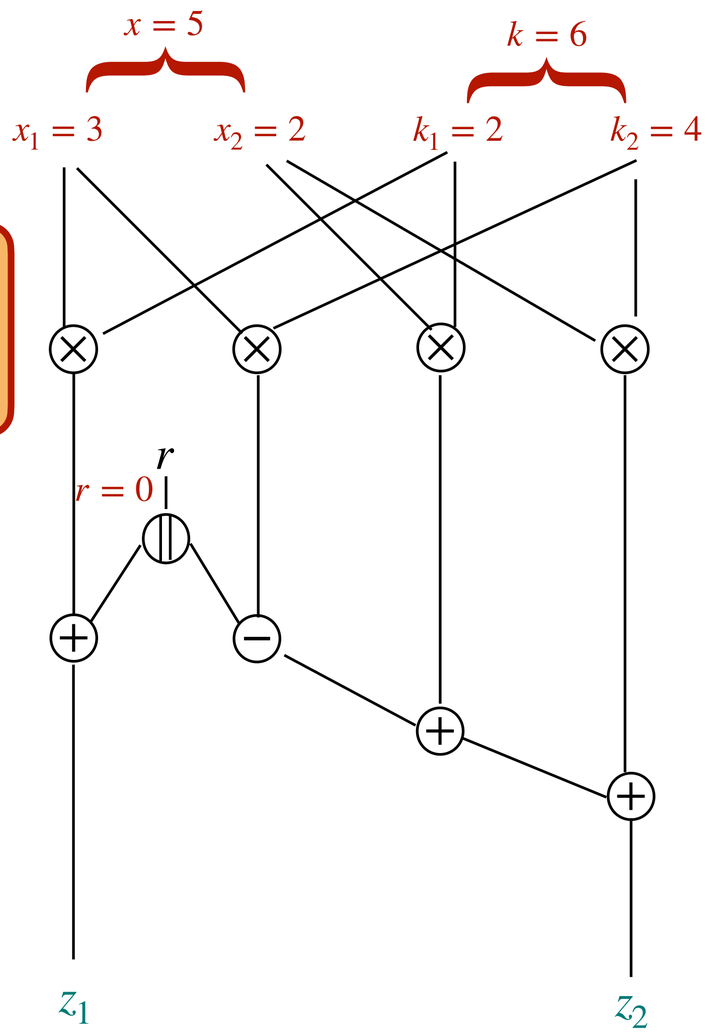


Leakage Models

Attacker view



Reality

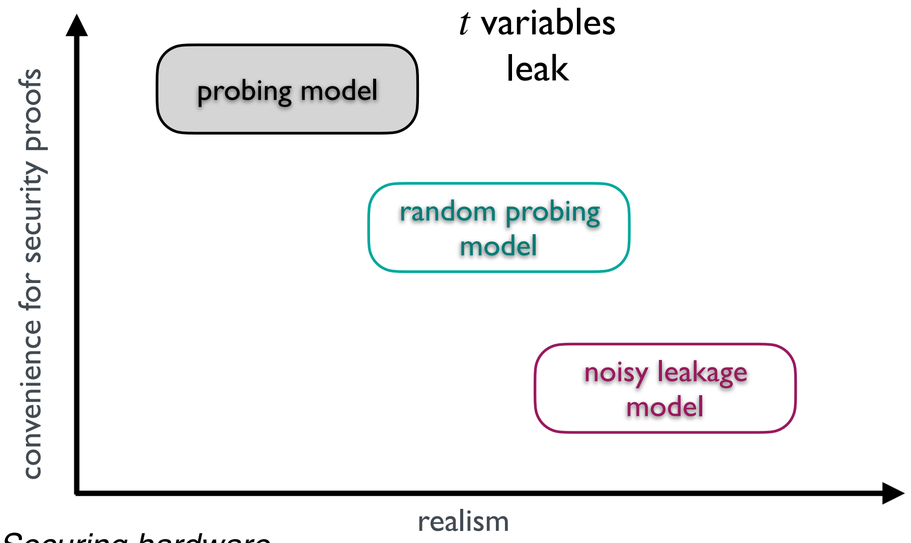


Attacker model

The attacker ← circuit + leakage

He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

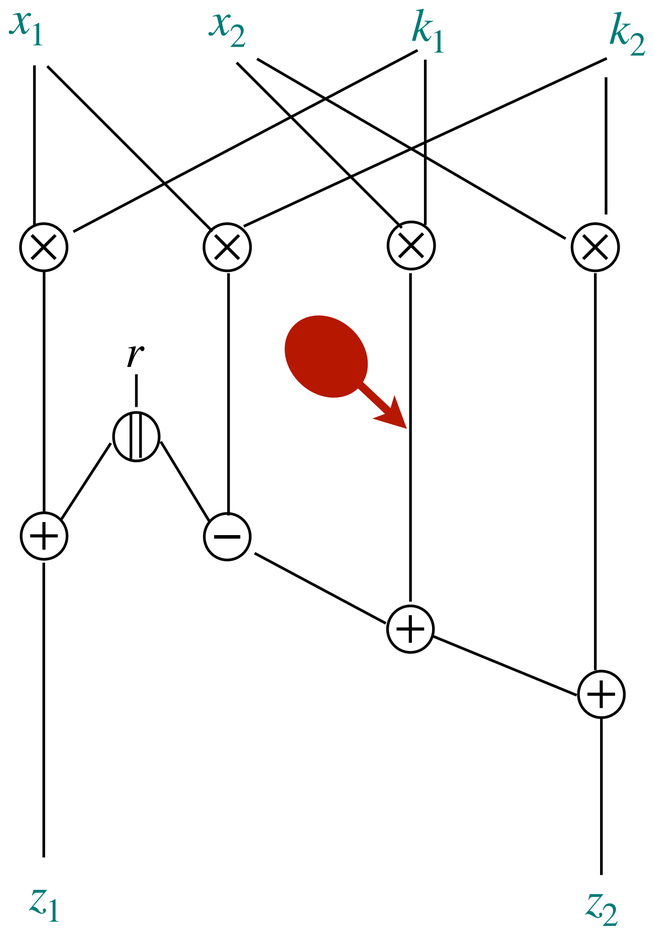
3 flavours



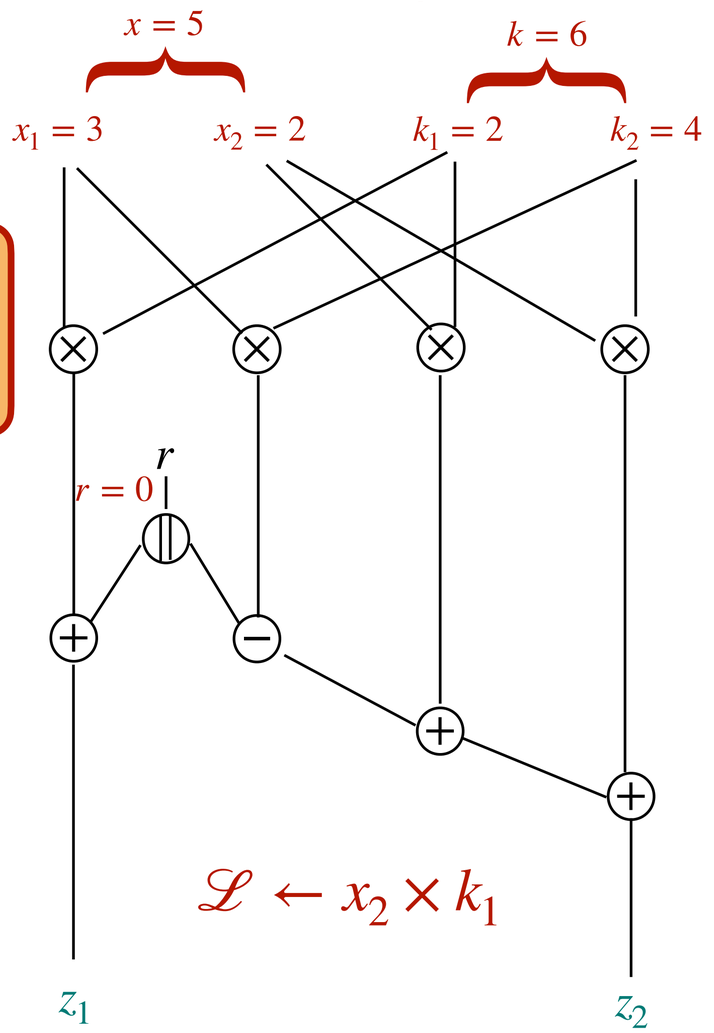
[ISW03] Y. Ishai, A. Sahai, and D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003

Leakage Models

Attacker view



Reality

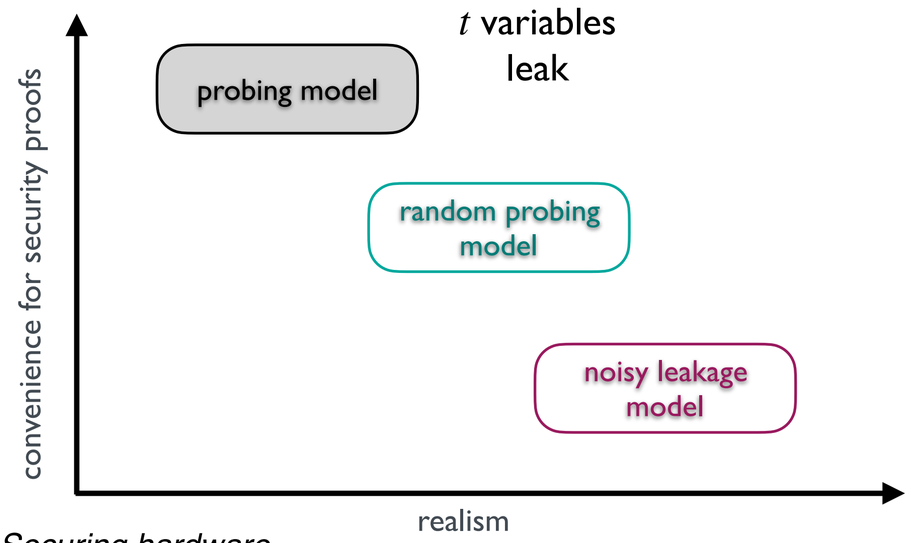


Attacker model

The attacker ← circuit + leakage

He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

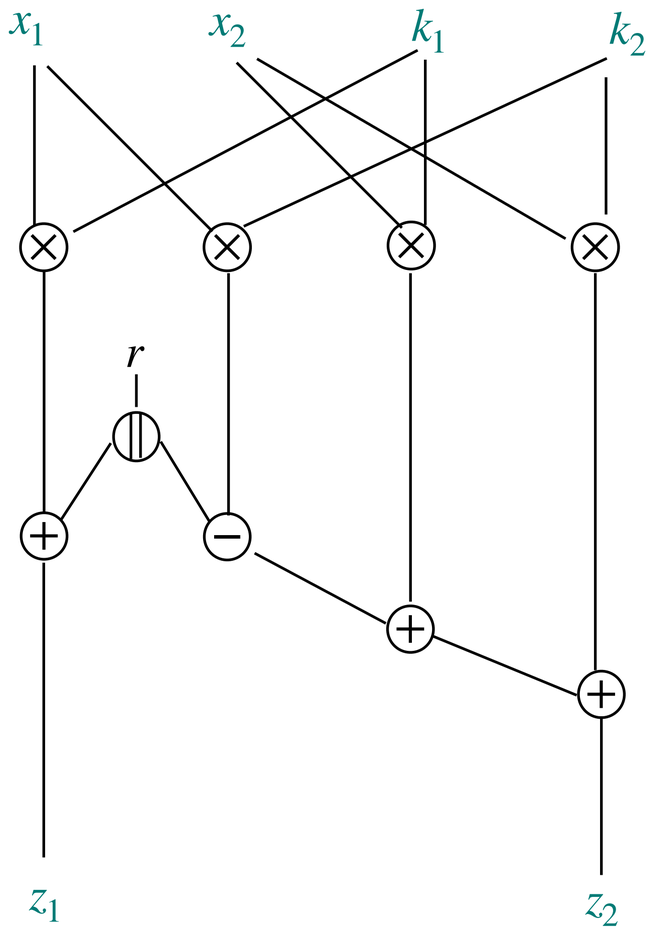
3 flavours



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

Random probing model

Attacker view

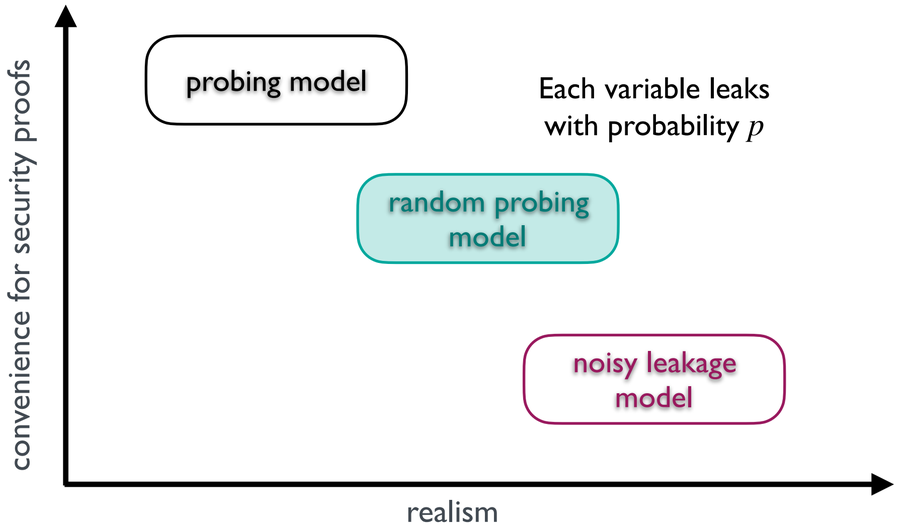


Attacker model

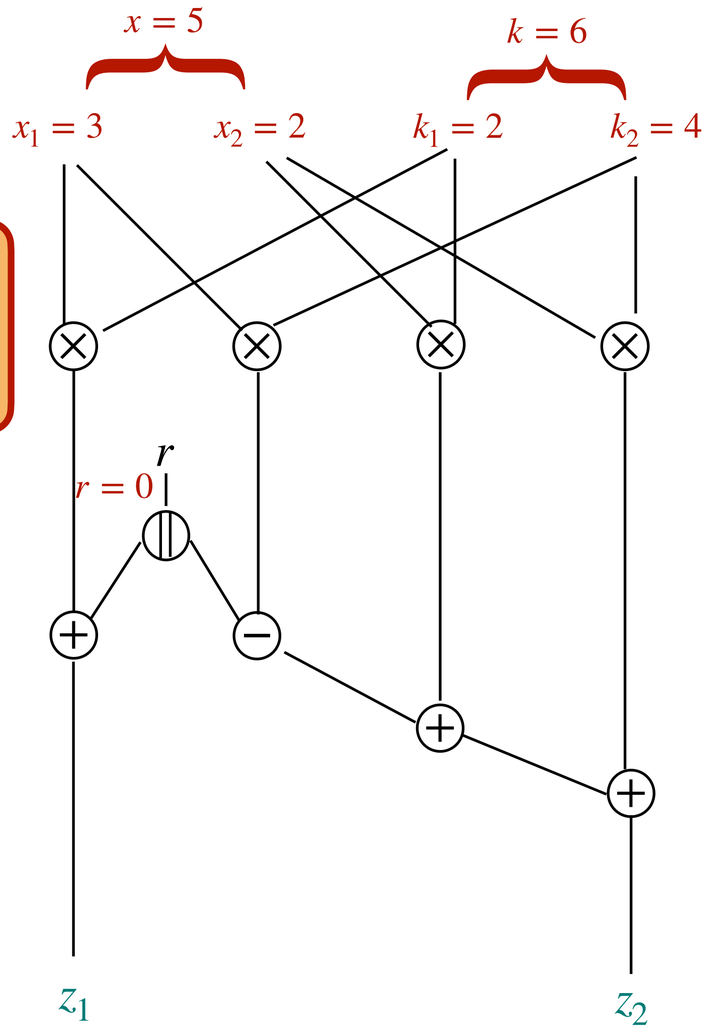
The attacker ← circuit + leakage

He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

3 flavours

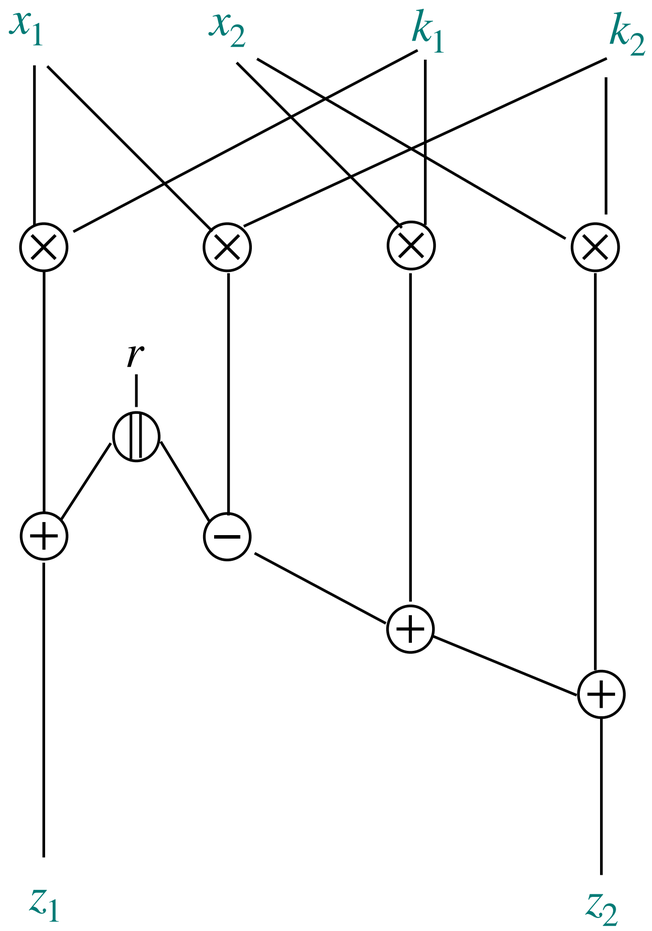


Reality

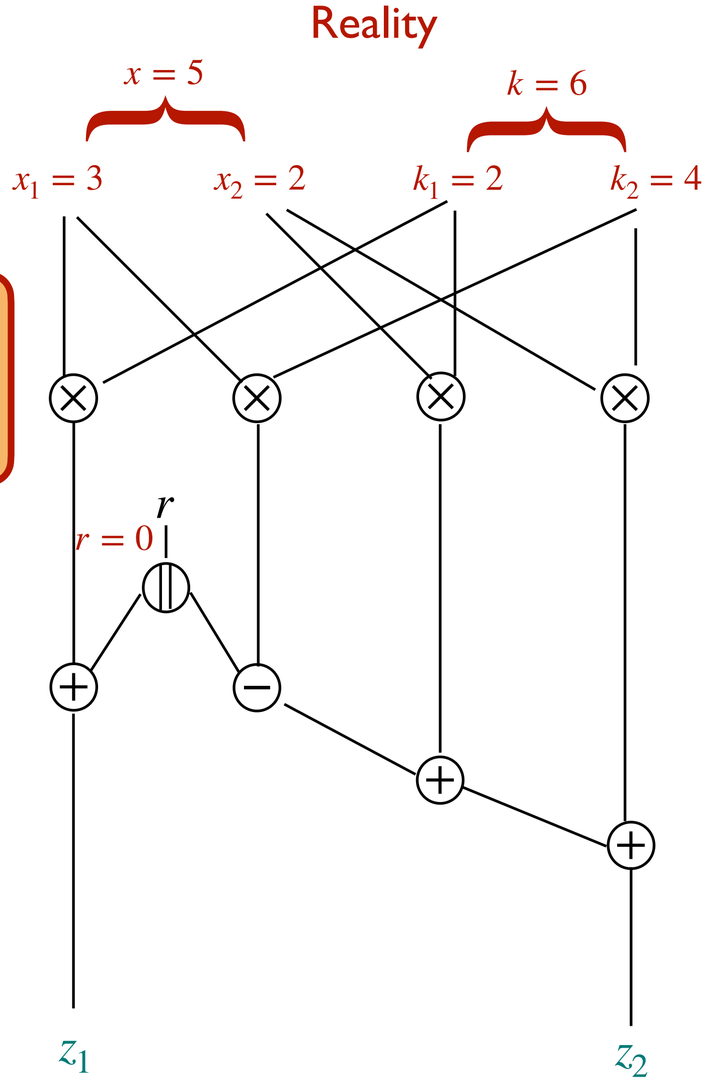


Random probing model

Attacker view



Reality

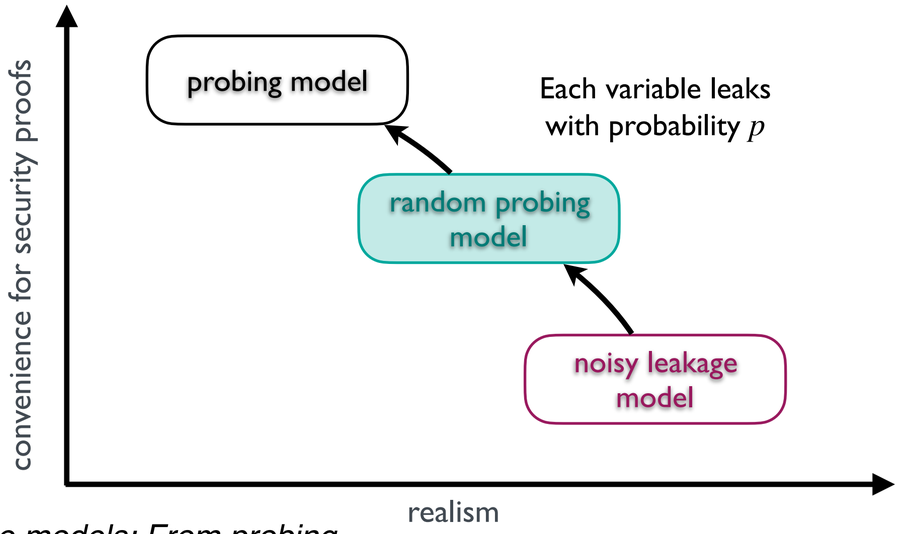


Attacker model

The attacker ← circuit + leakage

He/she must not recover any information about $x = \sum x_i$ and $k = \sum k_i$.

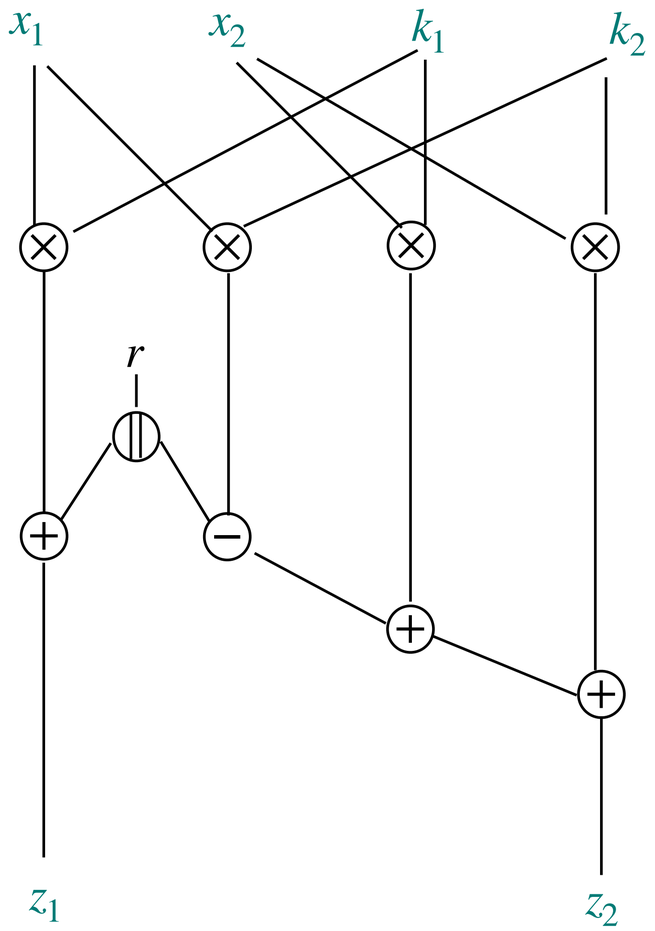
3 flavours



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

Random probing model

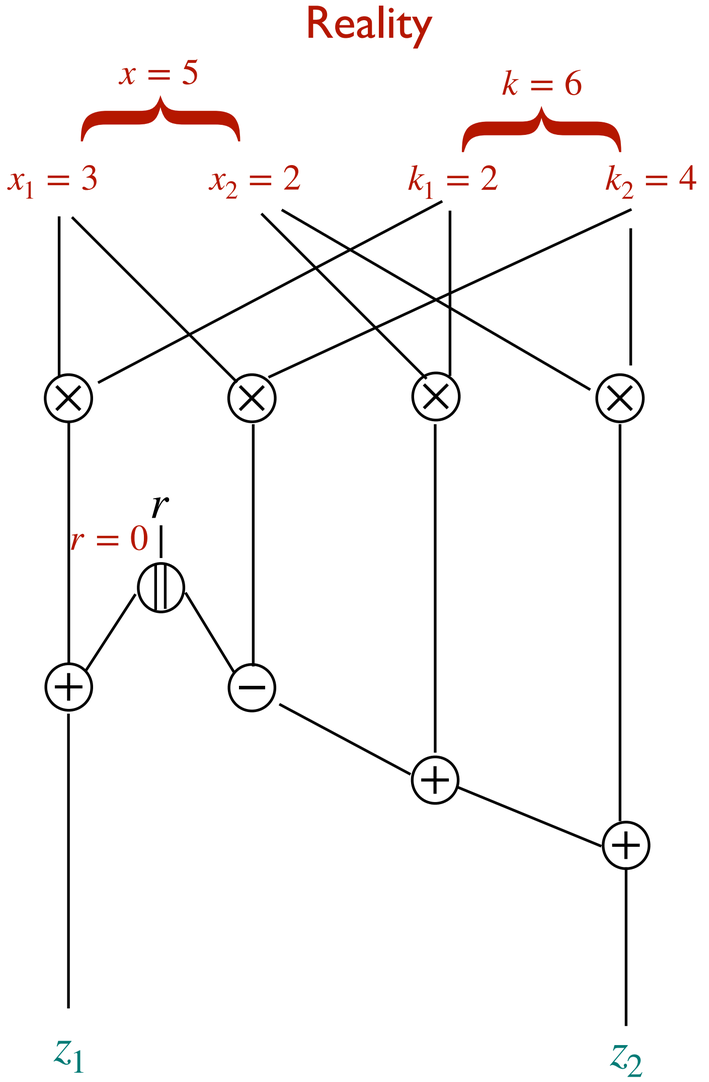
Attacker view



Attacker model

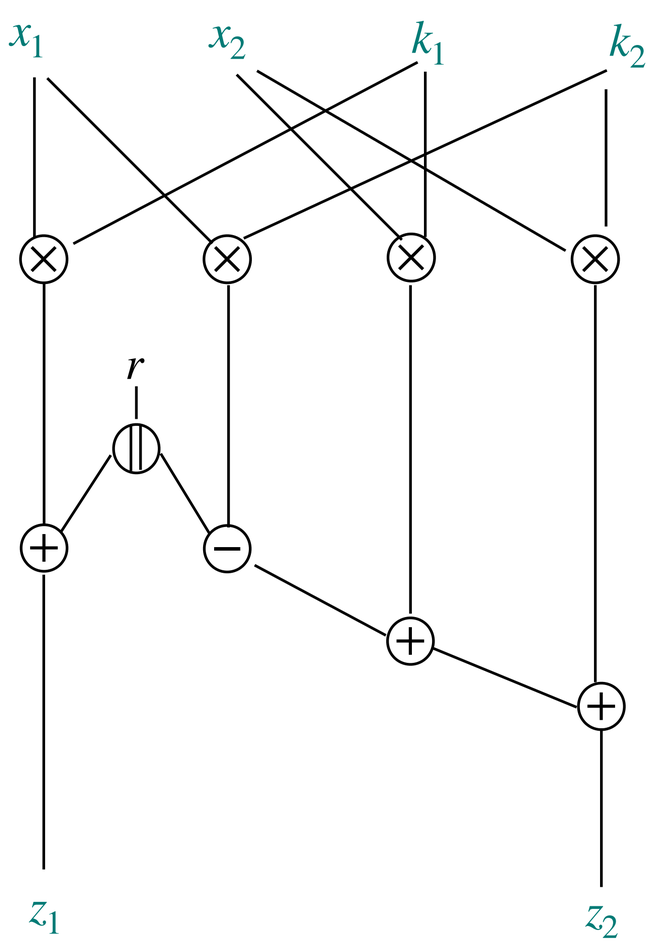
The attacker is given the value of each wire with probability p.

Reality



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

Random probing model

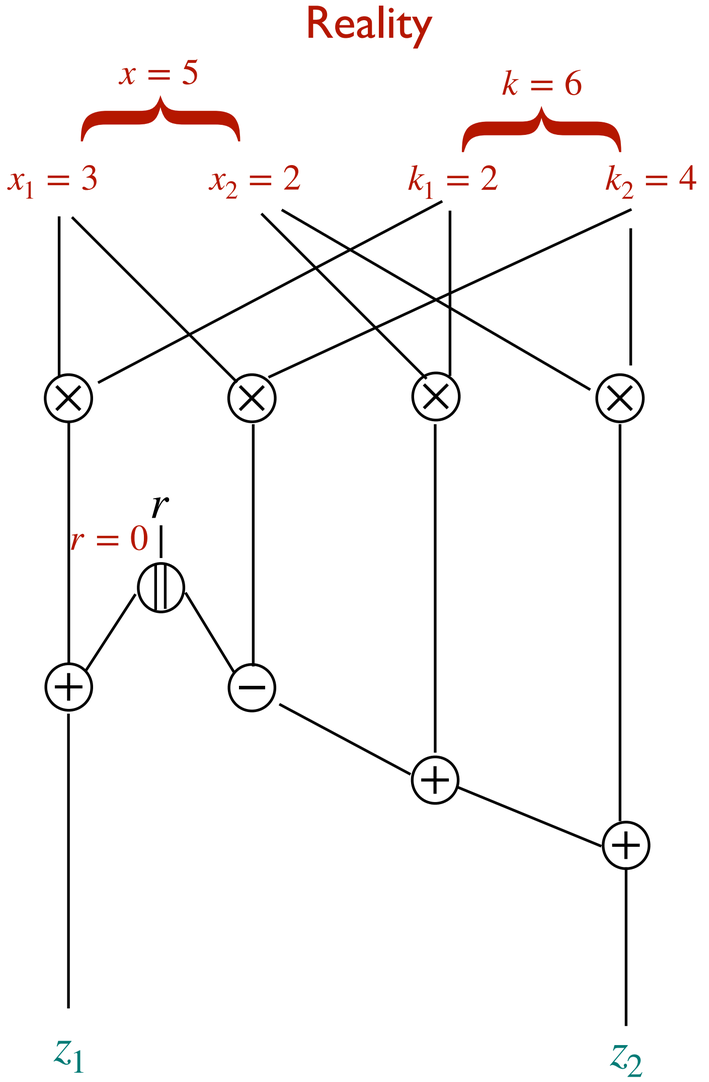


Attacker model

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

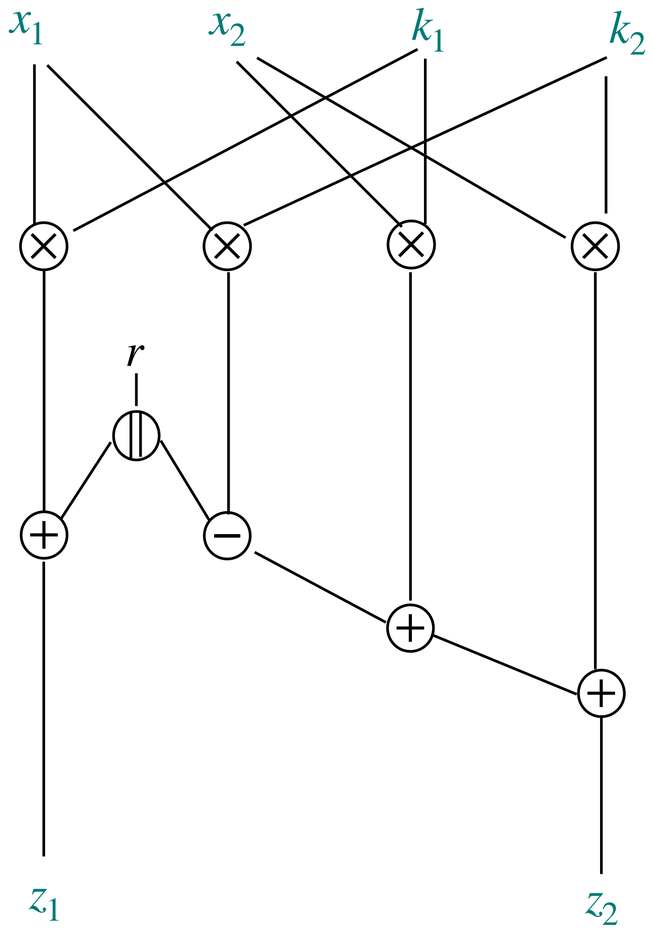
(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.



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

Random probing model



Attacker model

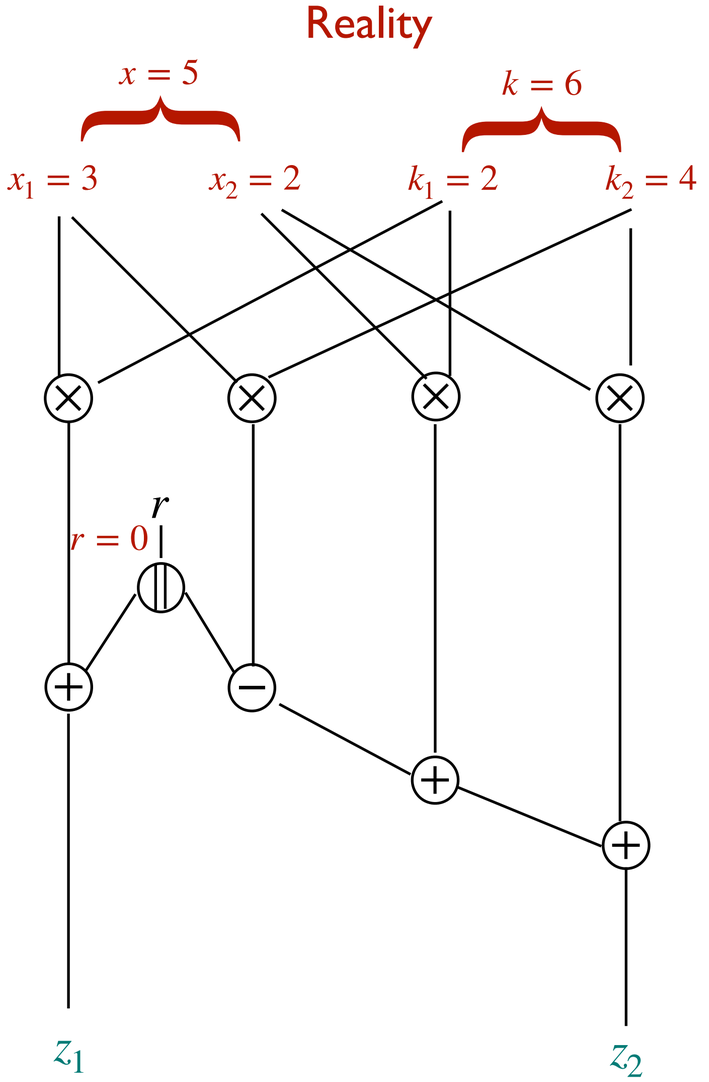
The attacker is given the value of each wire with probability p.

(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

Security Proof

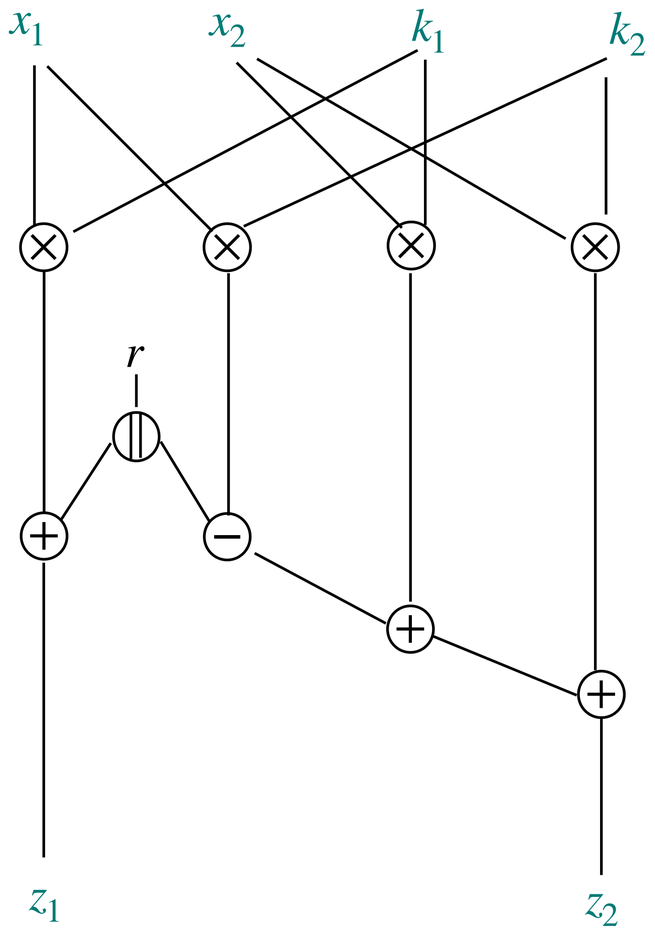
The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.



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

Random probing model

Attacker view



Attacker model

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

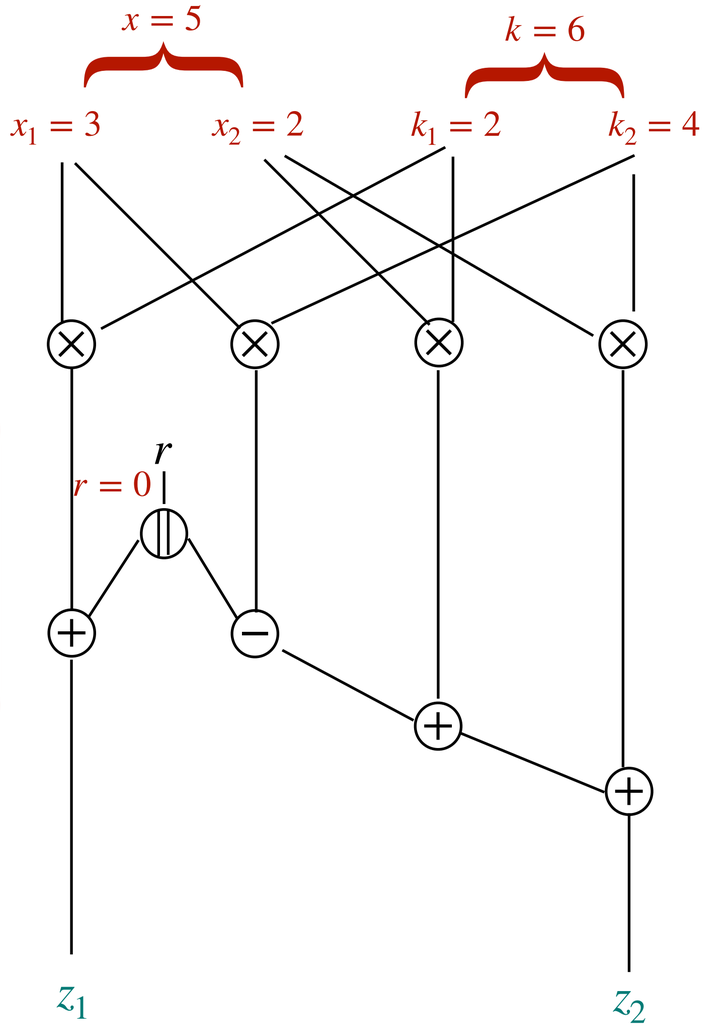
(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

Security Proof

The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.

Reality

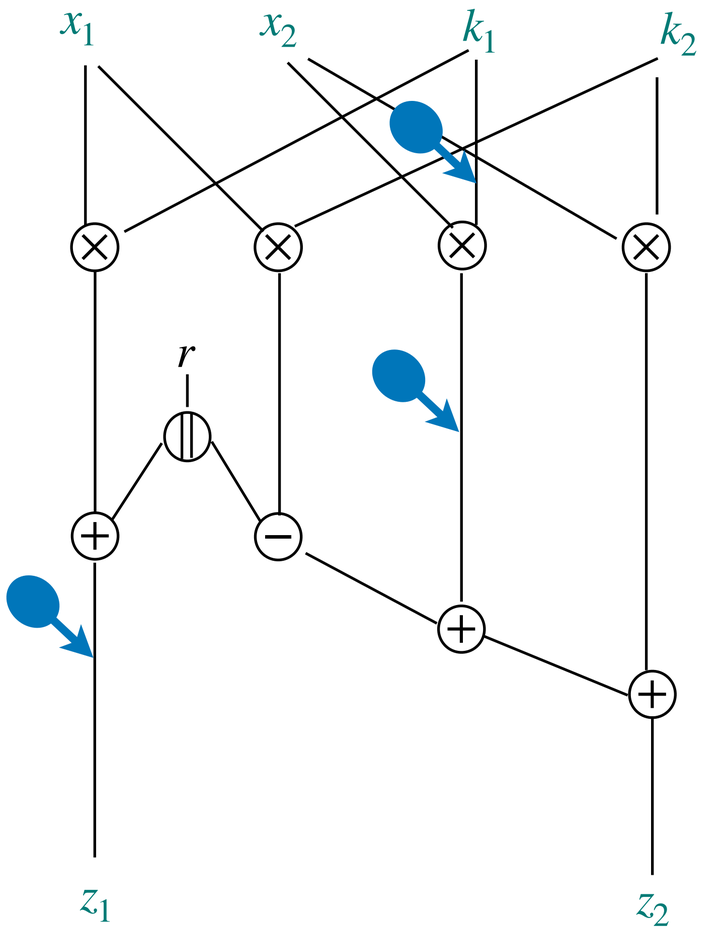


$\mathscr{W} = \varnothing$ with proba $(1-p)^{19}$

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

Random probing model

Attacker view



Attacker model

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

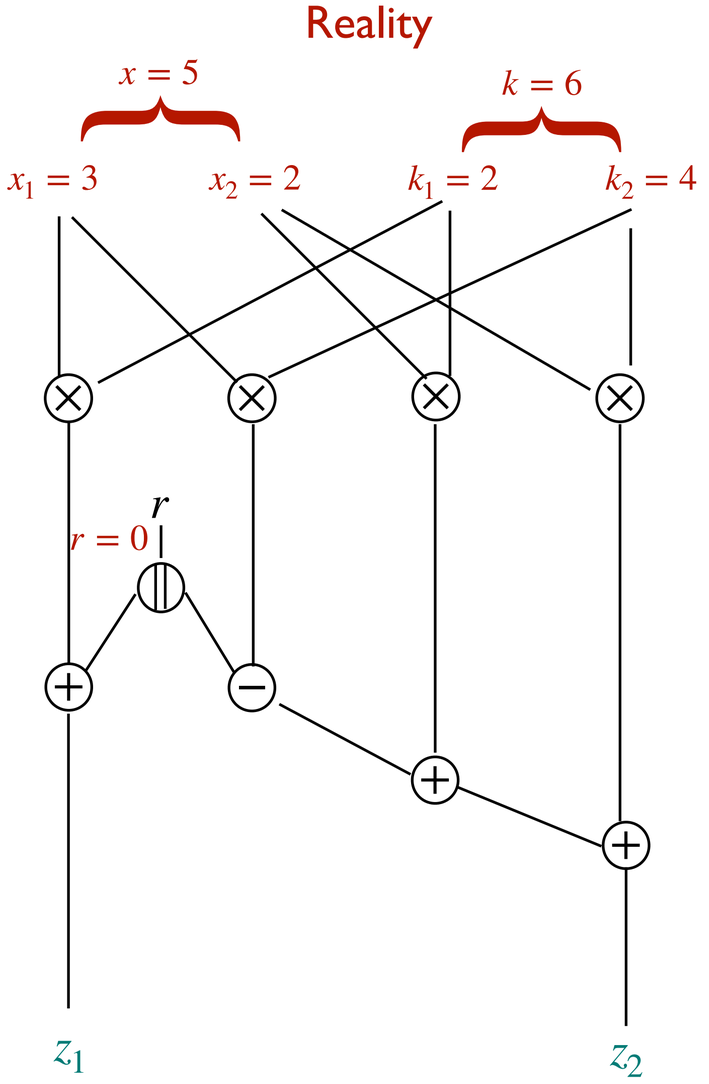
(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

Security Proof

The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.

Reality

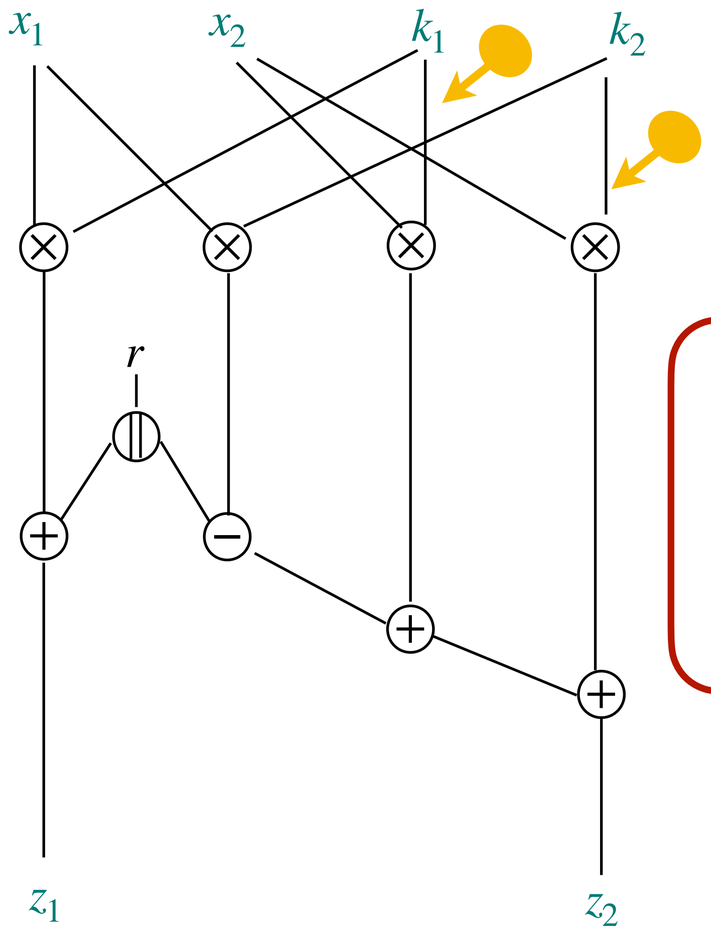


$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$
out $\leftarrow \{\$^1, \$^2 \times \$^3, \$^3\}$

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

Random probing model

Attacker view



Attacker model

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

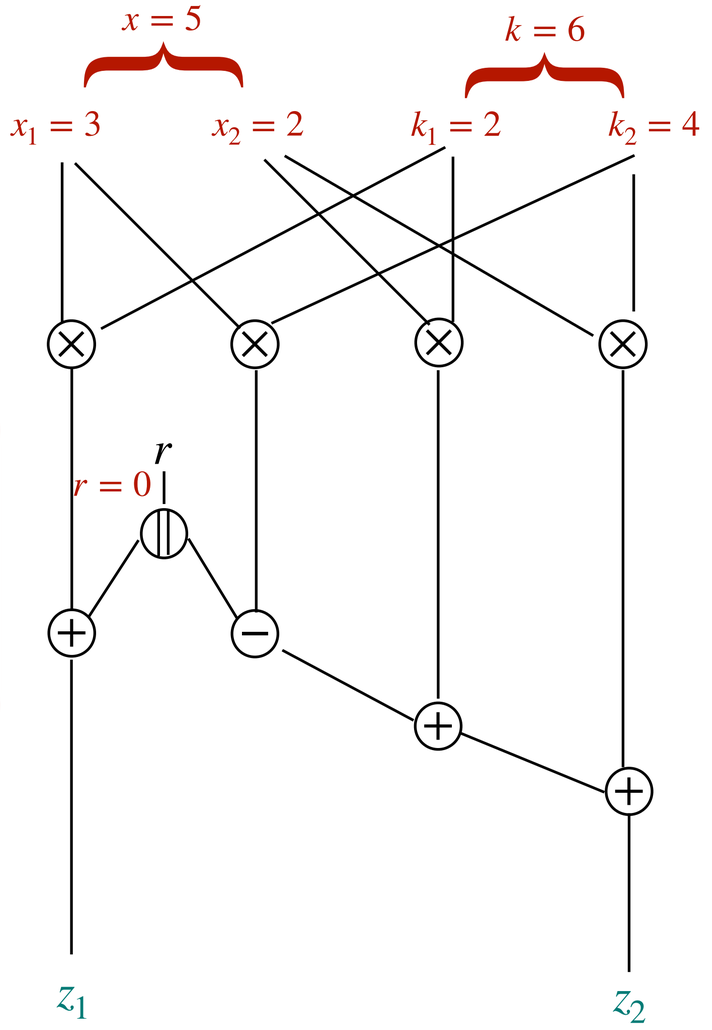
(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

Security Proof

The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.

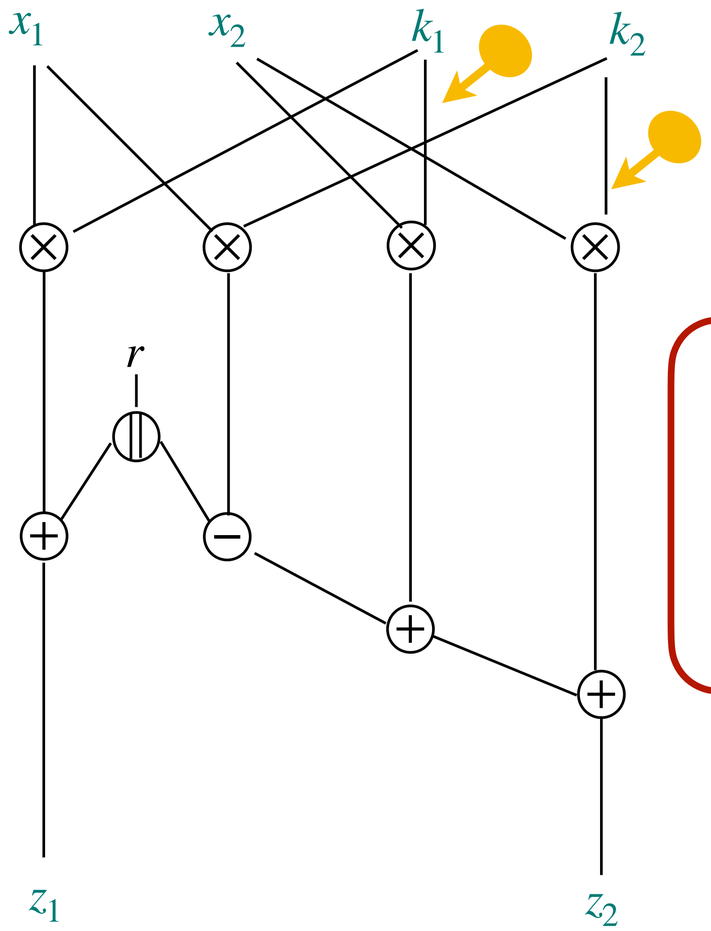
Reality



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

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

Random probing model



Attacker model

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

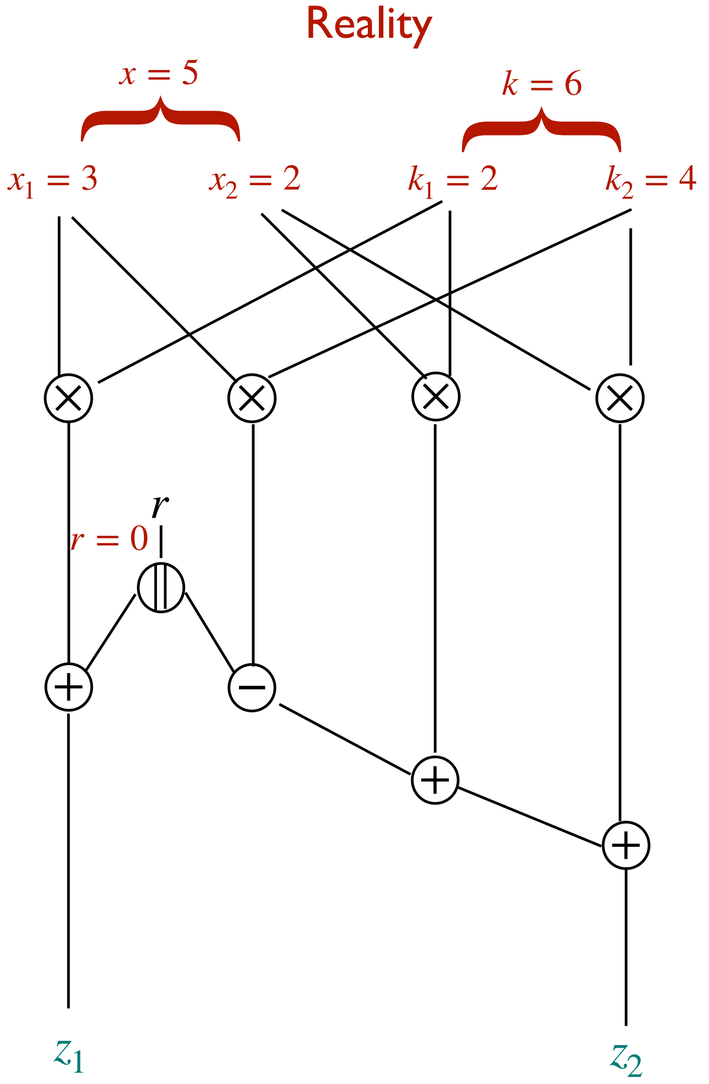
(p, ϵ)-random-probing security

Let $\mathscr{W}$ be a set of wires that are drawn with prob. p.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

…except with probability ϵ.

Security Proof

The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.

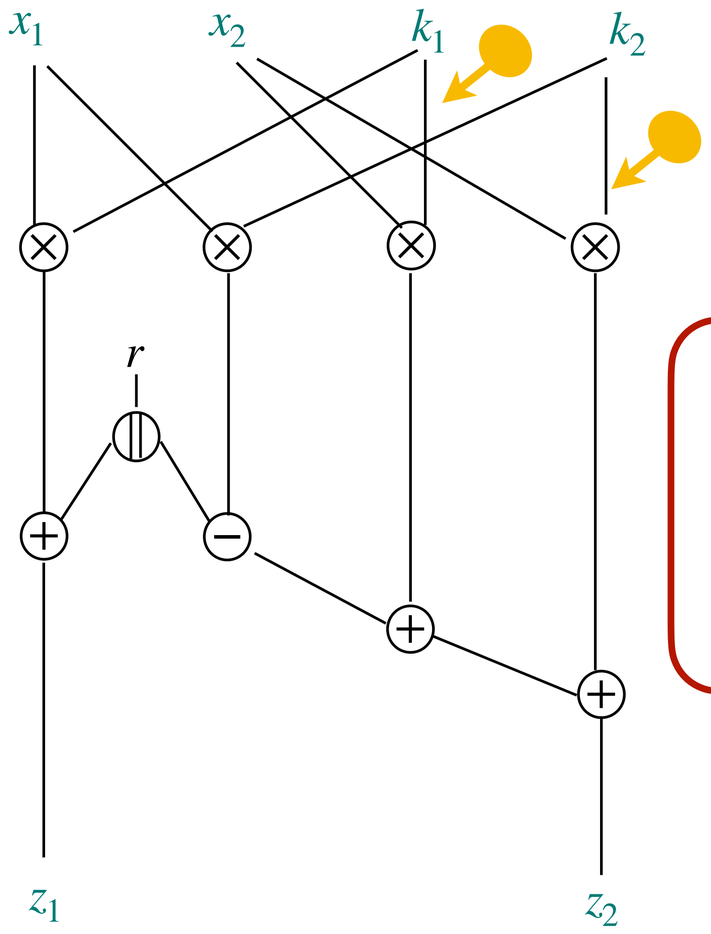


$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$
out $\leftarrow \{\$^*, k - \$^*\}$

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

Random probing model

Attacker view



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.
Given $\mathscr{W}$, the attacker cannot deduce the values of the secrets x and k.

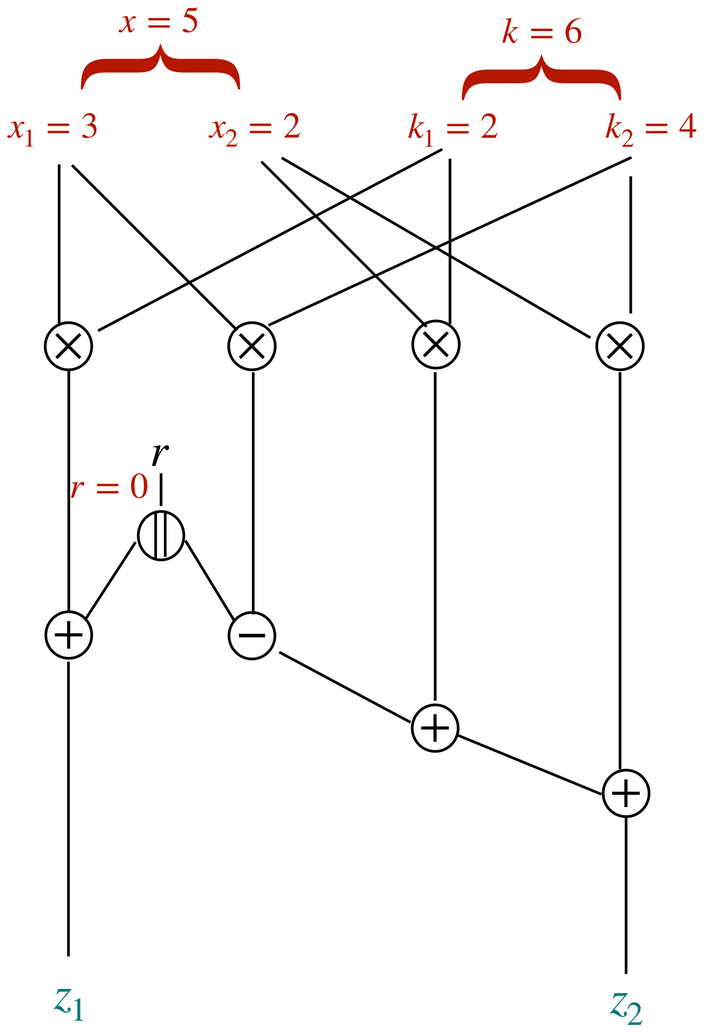
…except with probability ϵ.

Security Proof

The simulator provides *out* that is **simulated** without the secrets: $\mathscr{L} \overset{id}{\approx}_{\epsilon}$ out.

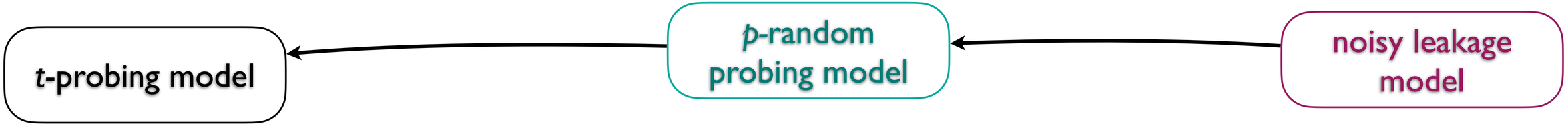
$\epsilon = 2^{-128} \implies p \geq$ some bound

Reality



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

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

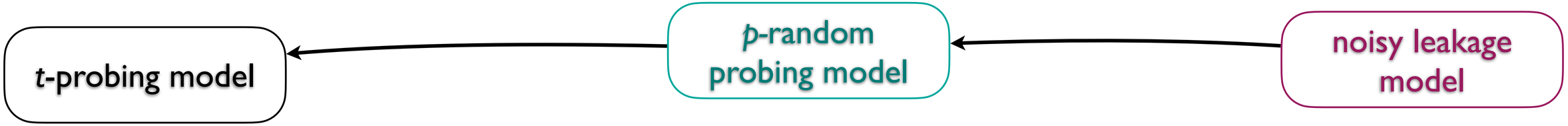
☐ Loose reduction to the noisy leakage model

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

➡ Masked implementations in this model 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 in this model 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 in this model 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 in this model 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) Scaling up is a big challenge

3) Random-probing Raccoon

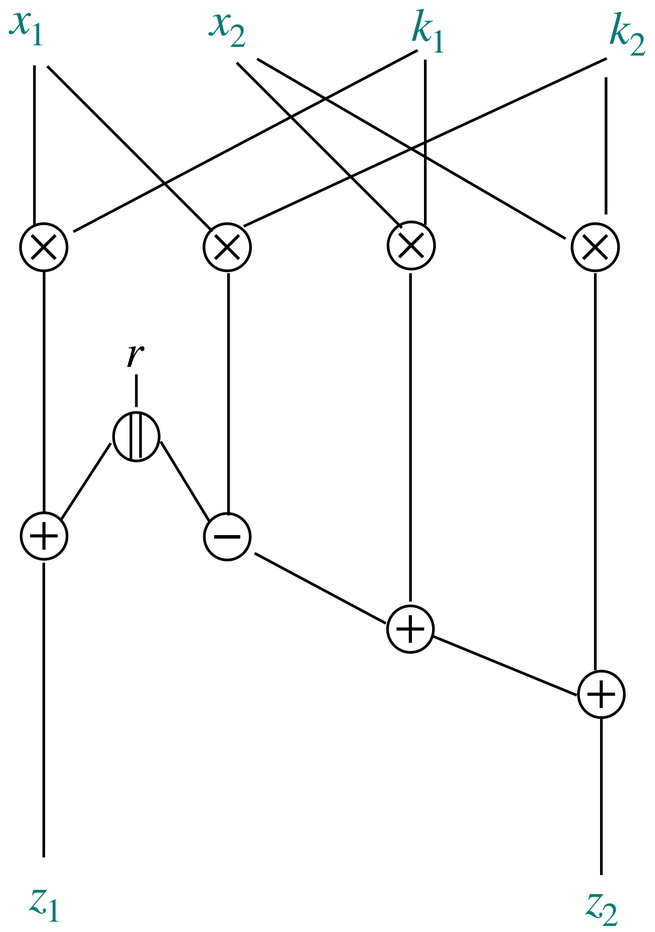
1) The relevance of the random probing model

2) Scaling up is a big challenge

3) Random-probing Raccoon

Random Probing Composability

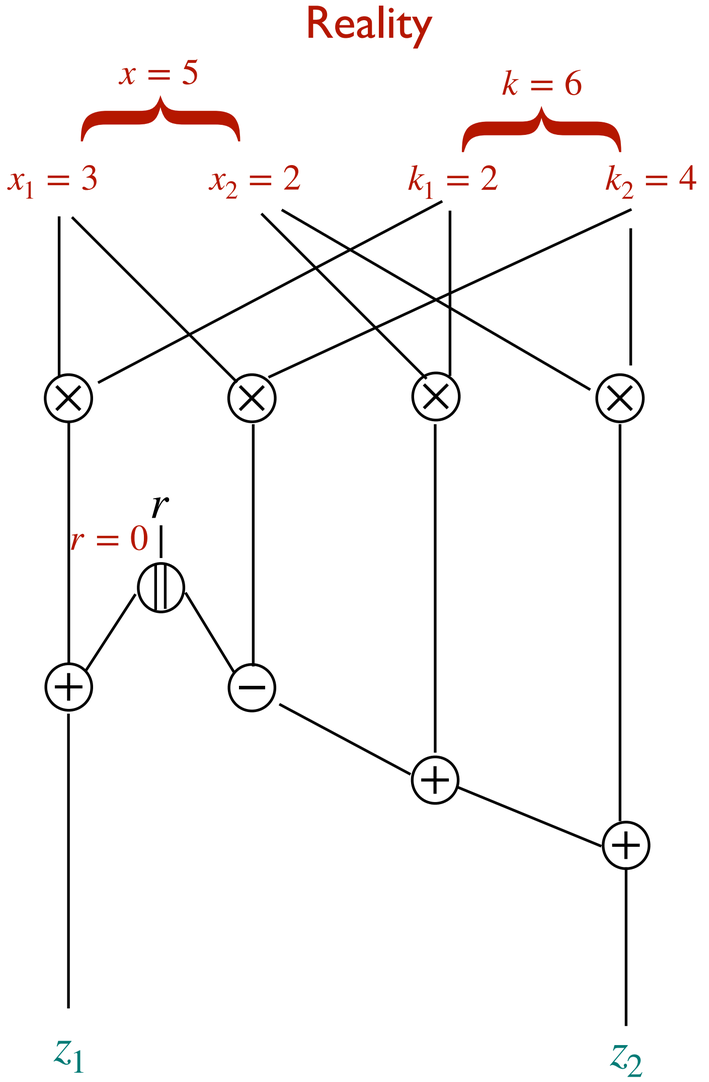
Attacker view



(p, ϵ, t)-threshold RPC

$\mathbb{P}$(« The simulator needs more than t shares of each $[\![x]\!]$ and $[\![k]\!]$ to simulate $\mathscr{L}$ + t output shares ») $\leq \epsilon$

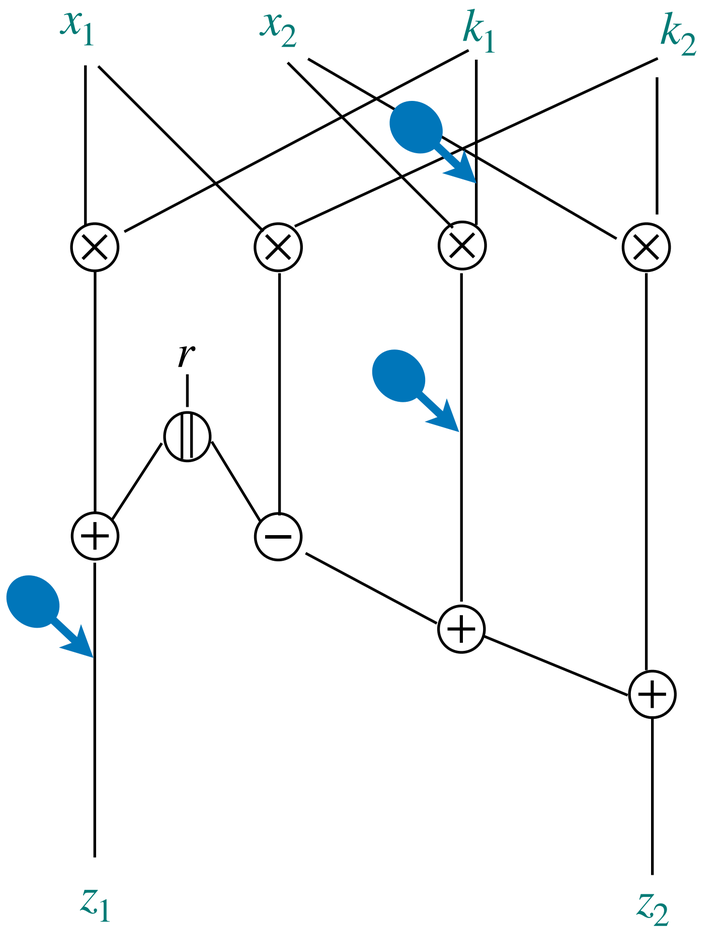
Reality



[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

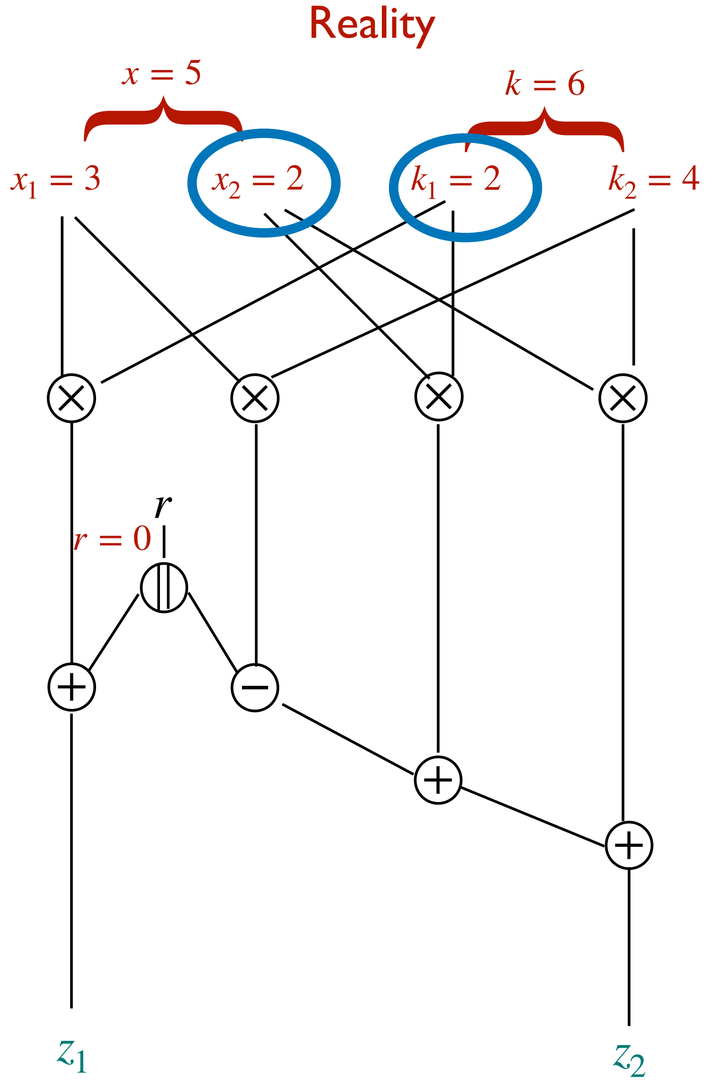
Attacker view



(p, ϵ, t)-threshold RPC

$\mathbb{P}$(« The simulator needs more than t shares of each $[\![x]\!]$ and $[\![k]\!]$ to simulate $\mathscr{L}$ + t output shares ») $\leq \varepsilon$

Reality



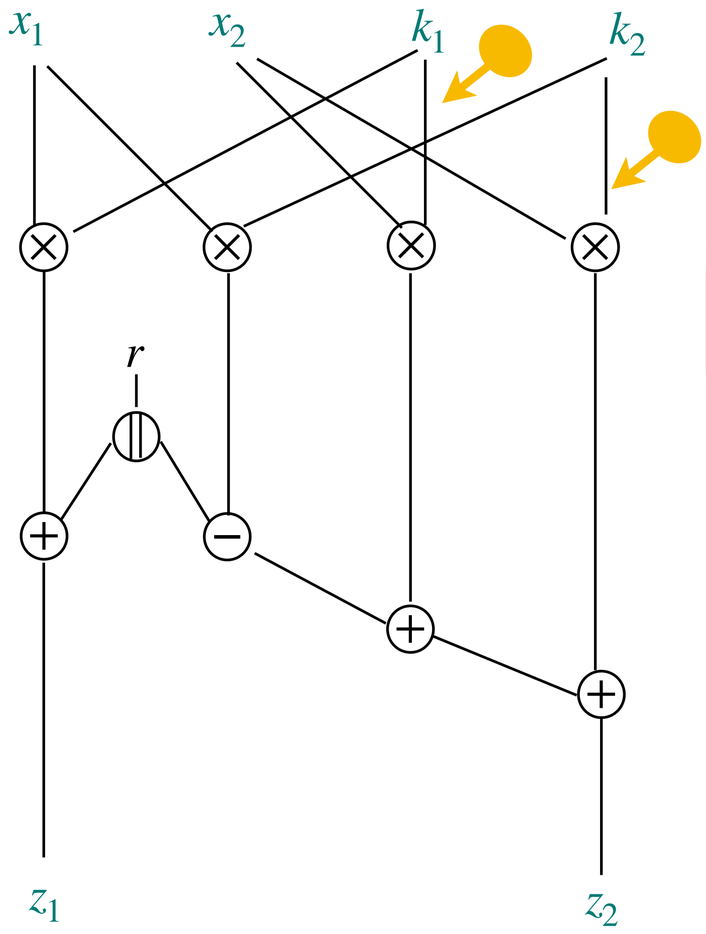
$\mathscr{W} = \{x_1k_1 + r, x_2k_1, k_1\}$ with proba $p^3(1-p)^{16}$

out $\leftarrow \{\$, x_2 \times k_1, k_1\}$

[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

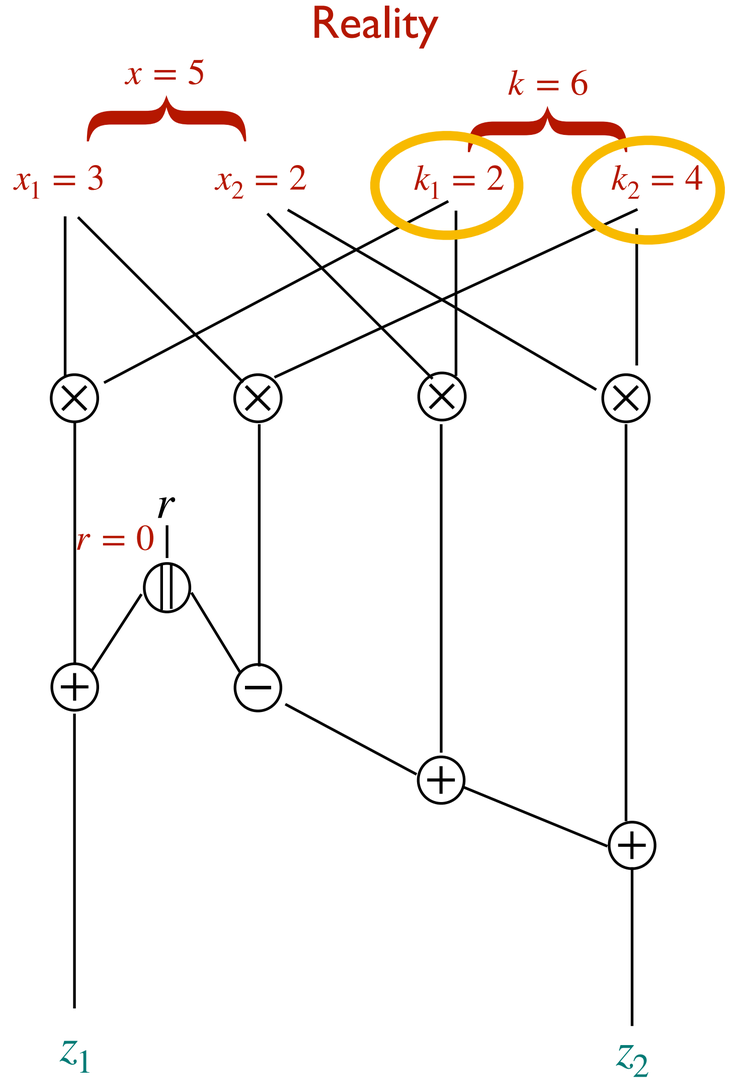
Attacker view



(p, ϵ, t)-threshold RPC

$\mathbb{P}$(« The simulator needs more than t shares of each $[\![x]\!]$ and $[\![k]\!]$ to simulate $\mathscr{L}$ + t output shares ») $\leq \varepsilon$

Reality

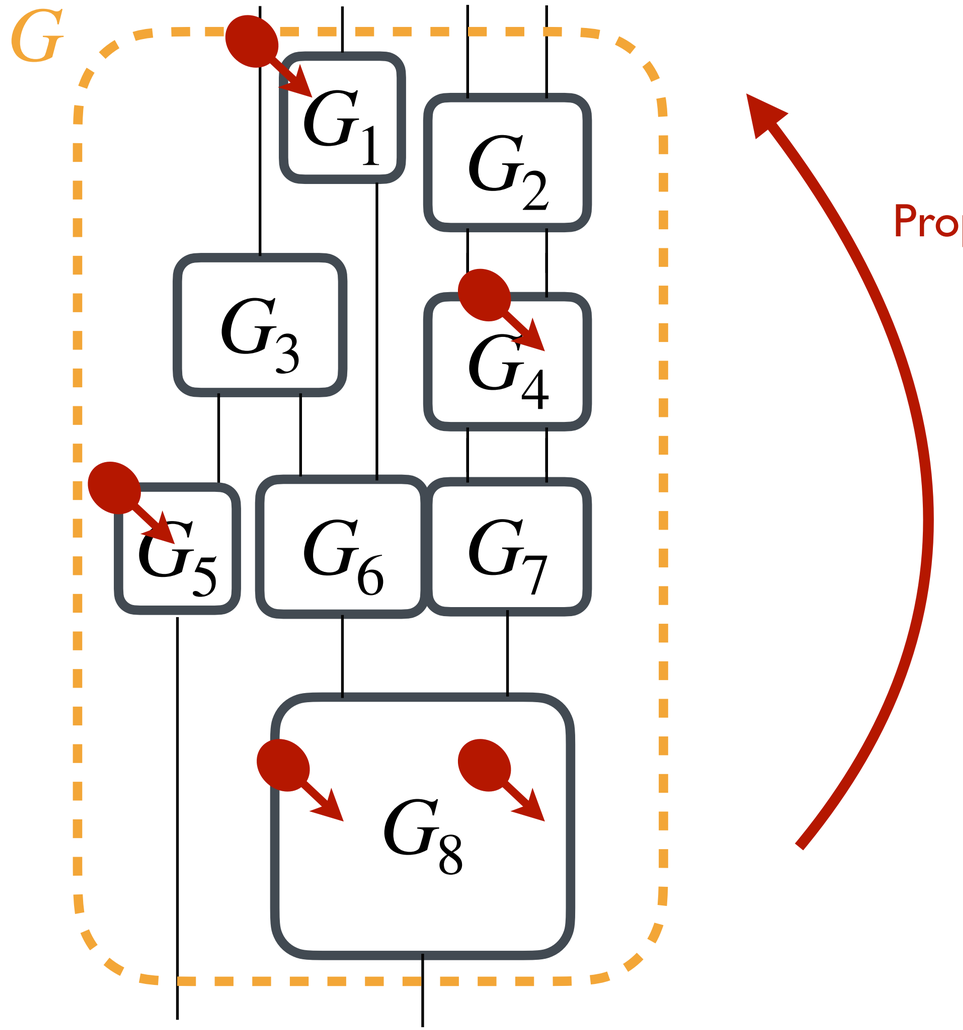


$\mathscr{W} = \{k_1, k_2\}$ with proba $p^2(1-p)^{17}$

out $\leftarrow \{k_1, k_2\}$

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

Composition with threshold RPC

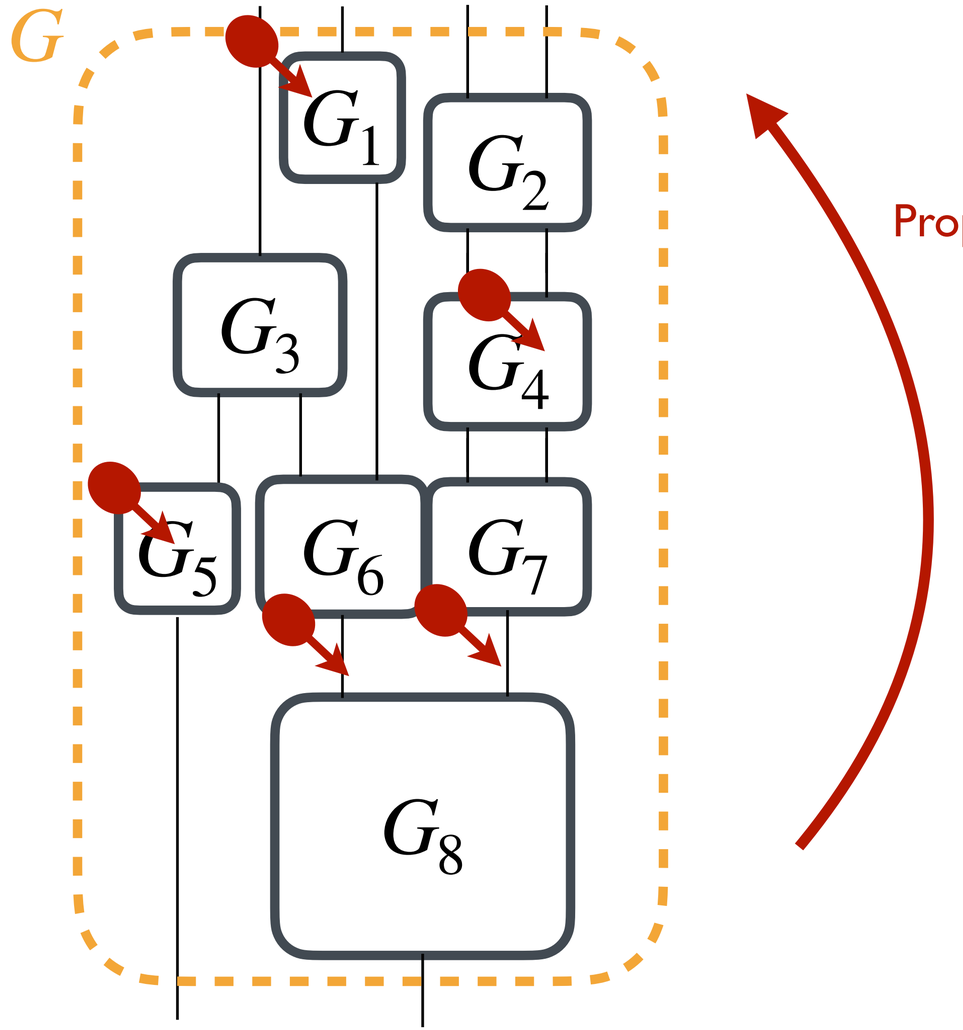


Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC

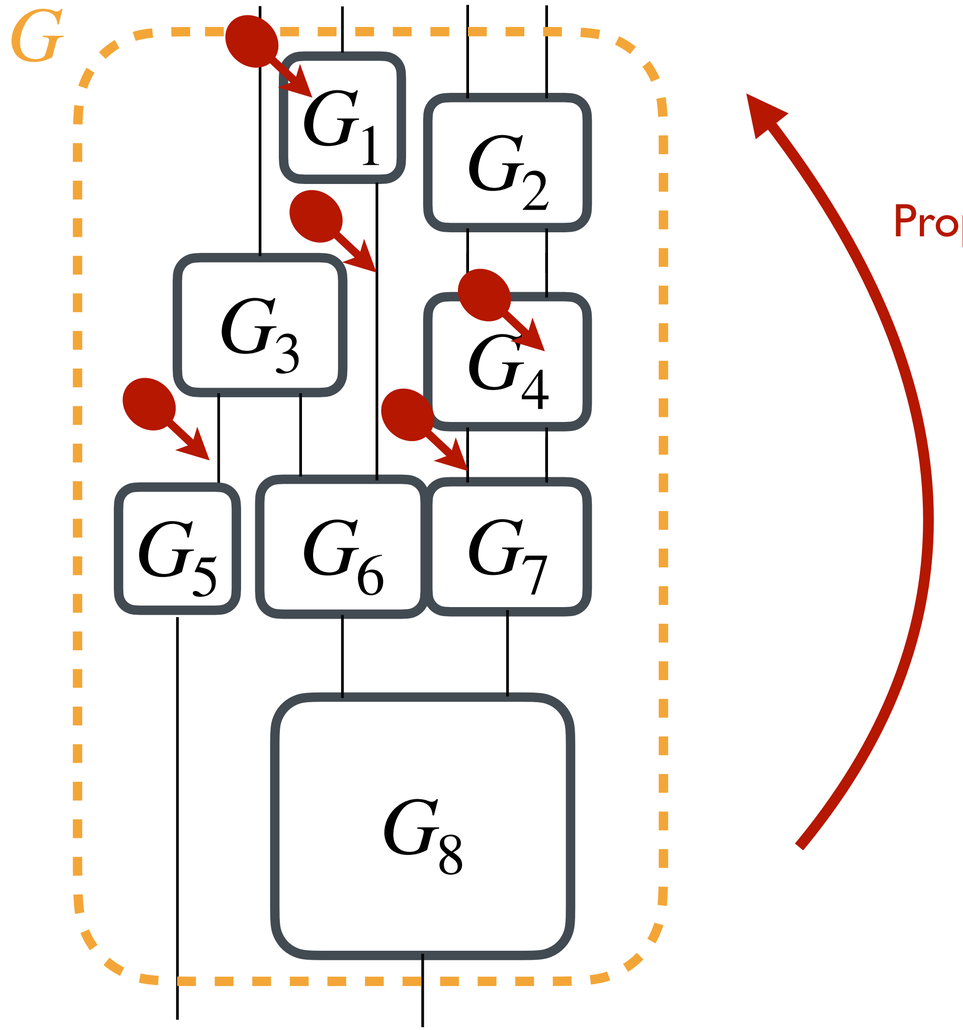


Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC

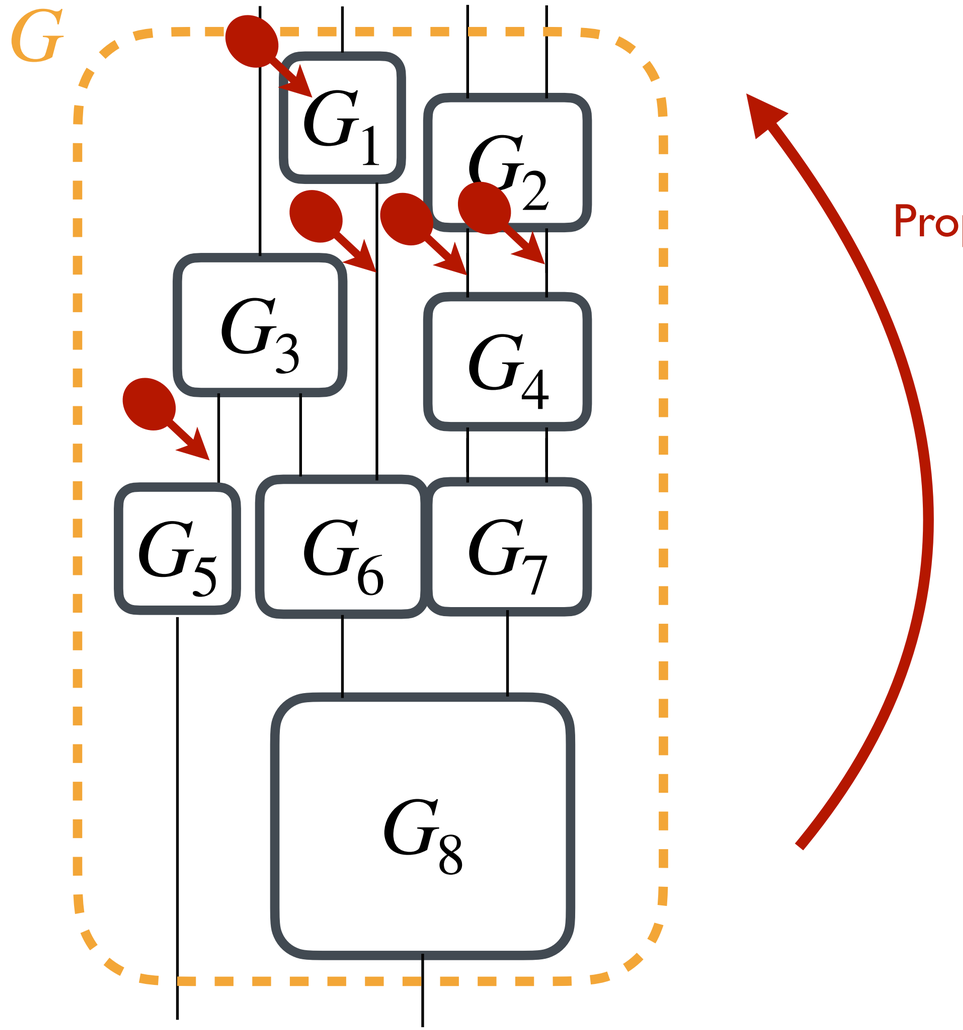


Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC

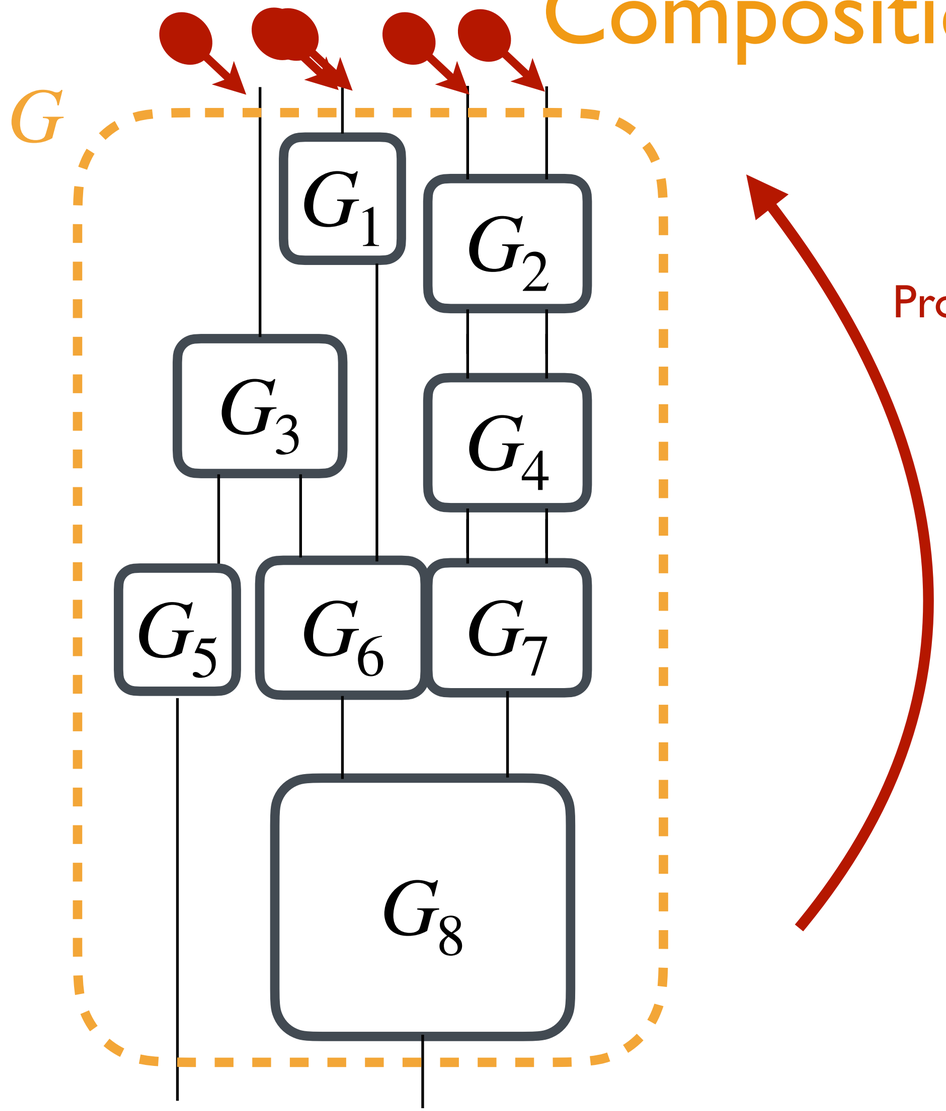


Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC

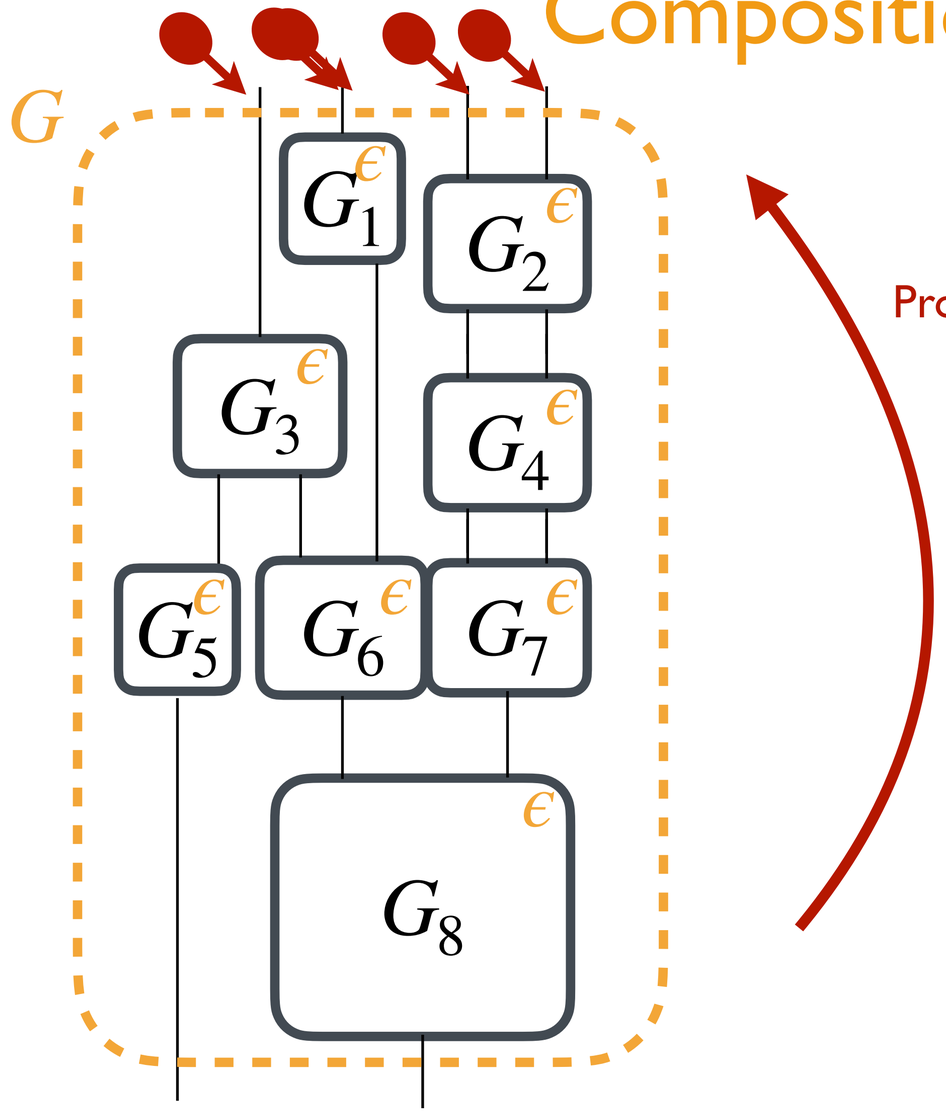


Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC



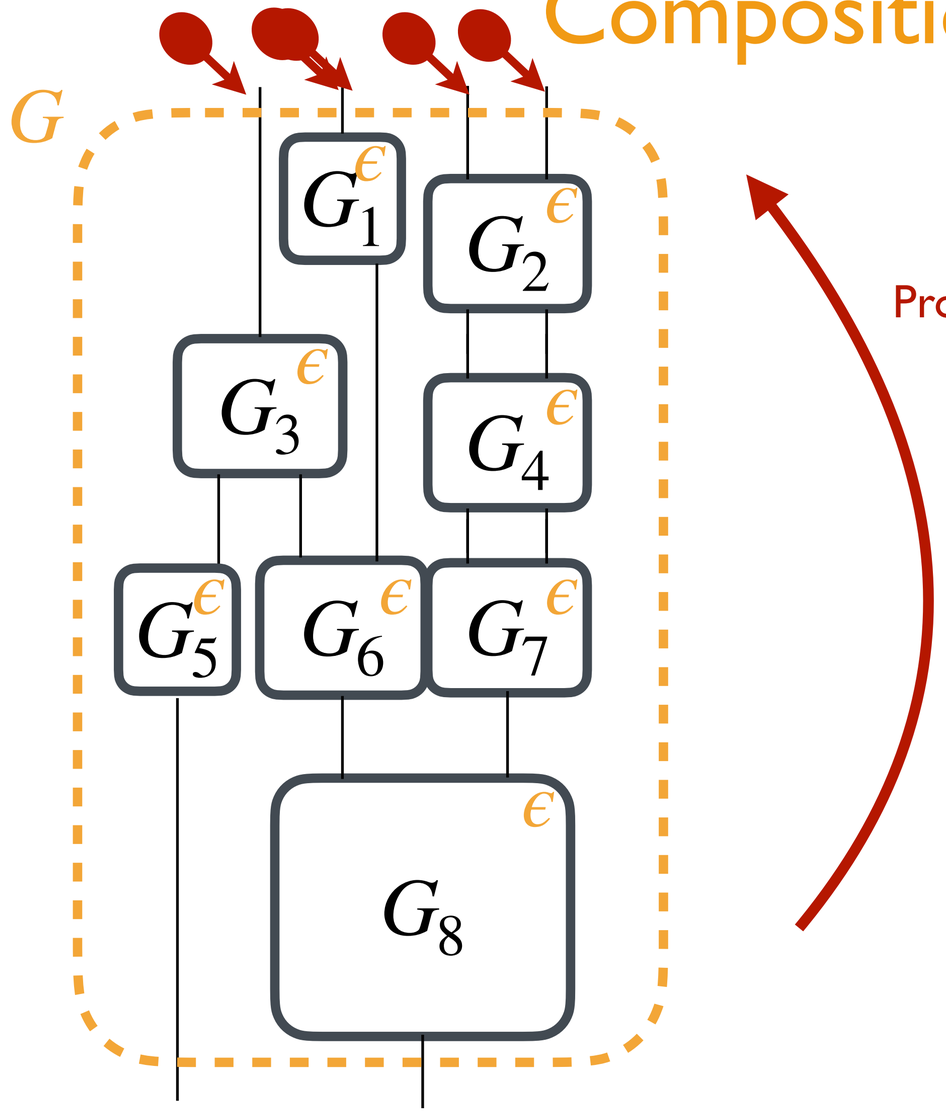
Threshold RPC:

Propagation of the leakage and the outputs to the 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

Composition with threshold RPC



Threshold RPC:

Propagation of the leakage and the outputs to the inputs

Except with probability ϵ!

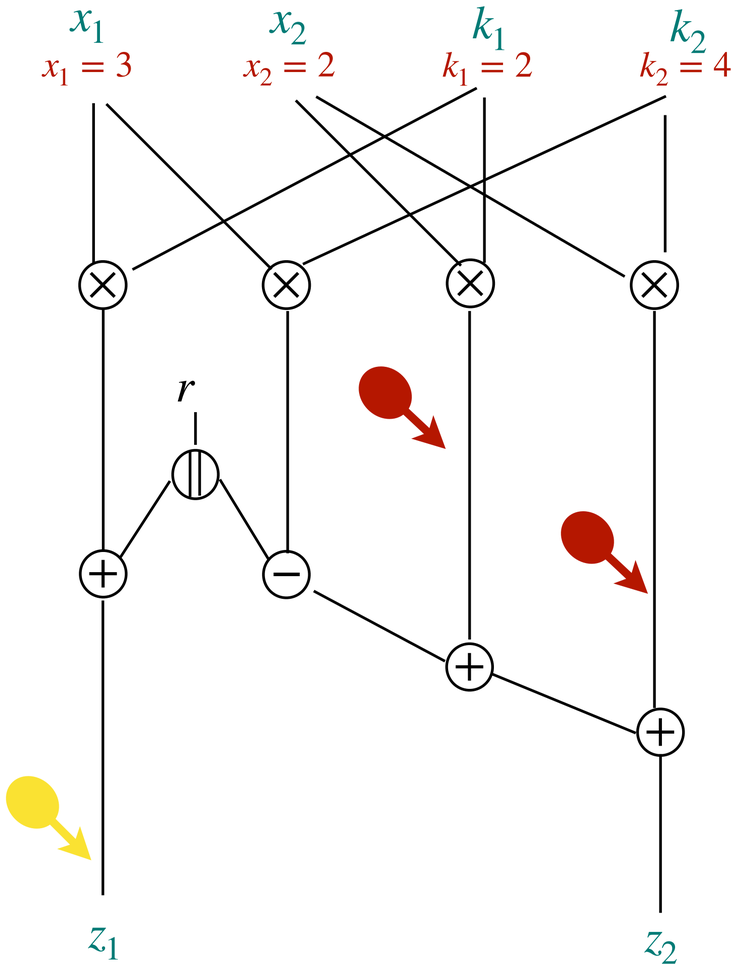
Composition

All G_i are (t, p, ϵ)-threshold RPC $\Longrightarrow$ G is (t, p, ϵ') -threshold RPC with

$$\epsilon' \leq 8\epsilon.$$

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

Tighter Compositions

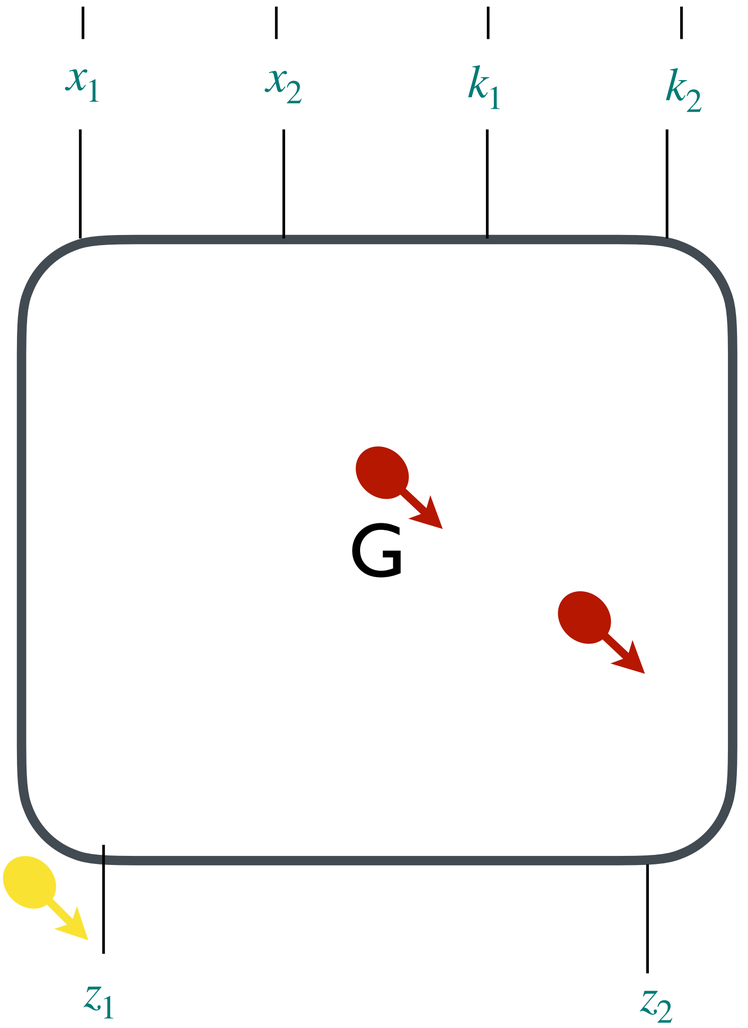


[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

[CFOS21] G. Cassiers, S. Faust, M. Orlt and F-X. Standaert. *Towards Tight Random Probing Security* published in Crypto 2021

Tighter Compositions

	Threshold RPC	General RPC	Cardinal RPC
	$\leq t$	All the sets	All the cardinals
	$> t$	All the sets	All the cardinals

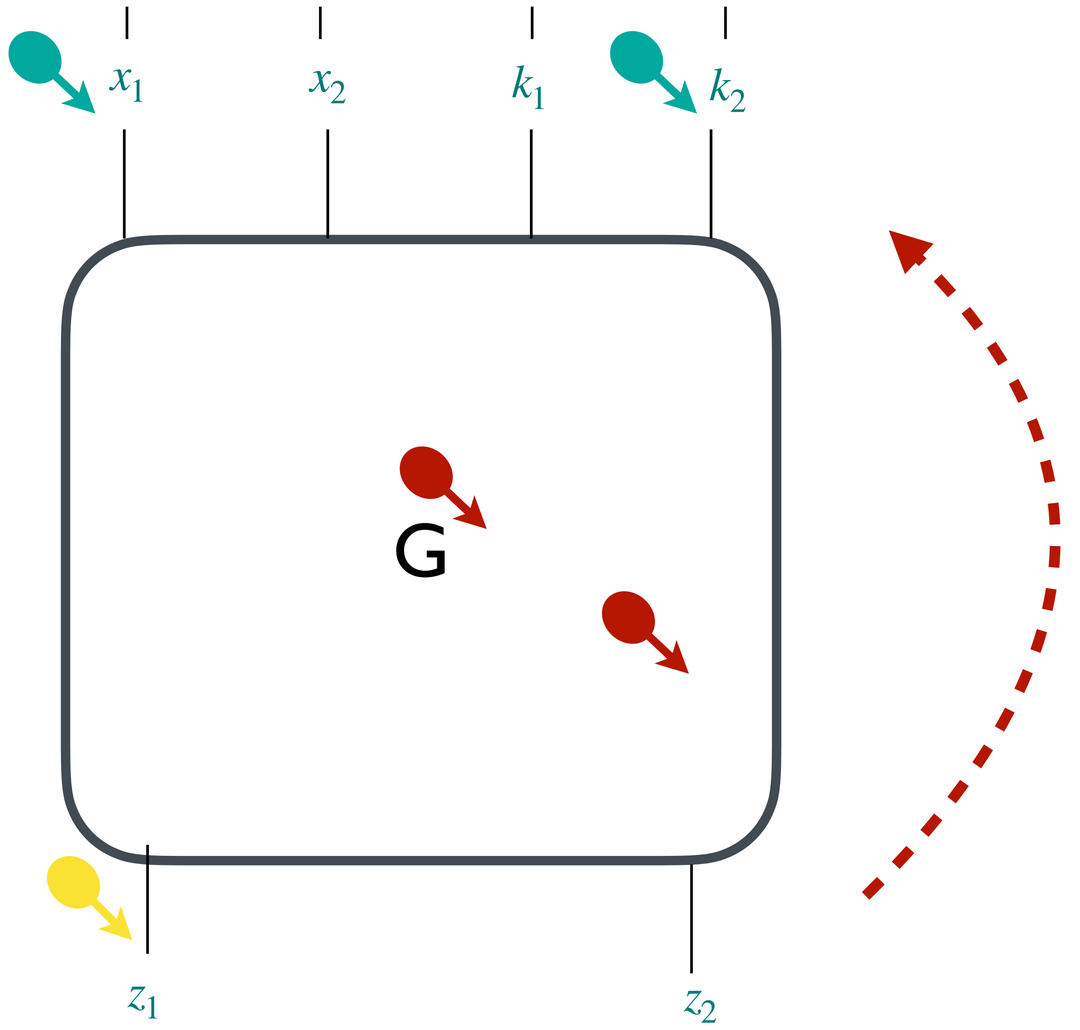


[BCPRT] *Random probing security: Verification, composition, expansion and new constructions.* Belaïd, S., Coron, J.S., Prouff, E., Rivain, M., Taleb, A.R., CRYPTO 2020

[CFOS21] G. Cassiers, S. Faust, M. Orlt and F-X. Standaert. *Towards Tight Random Probing Security* published in Crypto 2021

Tighter Compositions

	Threshold RPC	General RPC	Cardinal RPC
	$\leq t$	All the sets	All the cardinals
	$> t$	All the sets	All the cardinals



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

[CFOS21] G. Cassiers, S. Faust, M. Orlt and F-X. Standaert. *Towards Tight Random Probing Security* published in Crypto 2021

1) The relevance of the random probing model

2) Scaling up is a big challenge

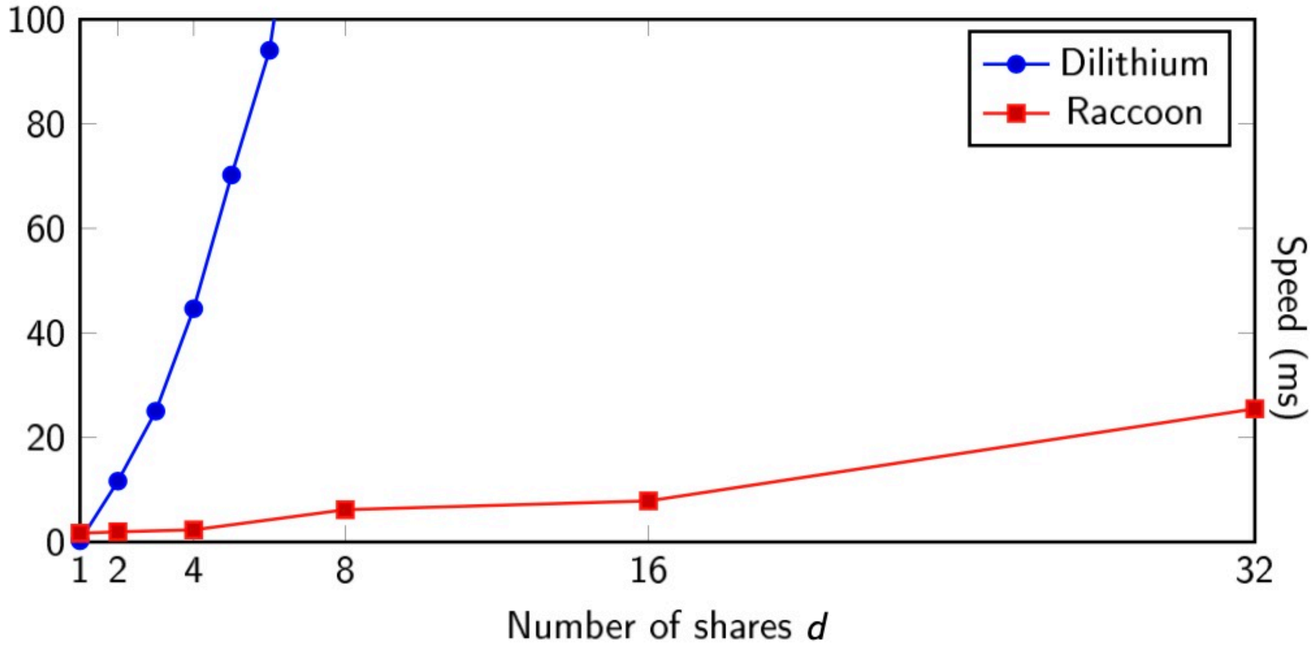
3) Random-probing Raccoon

1) The relevance of the random probing model

2) Scaling up is a big challenge

3) Random-probing Raccoon

Raccoon Signature Scheme



Raccoon 128-16

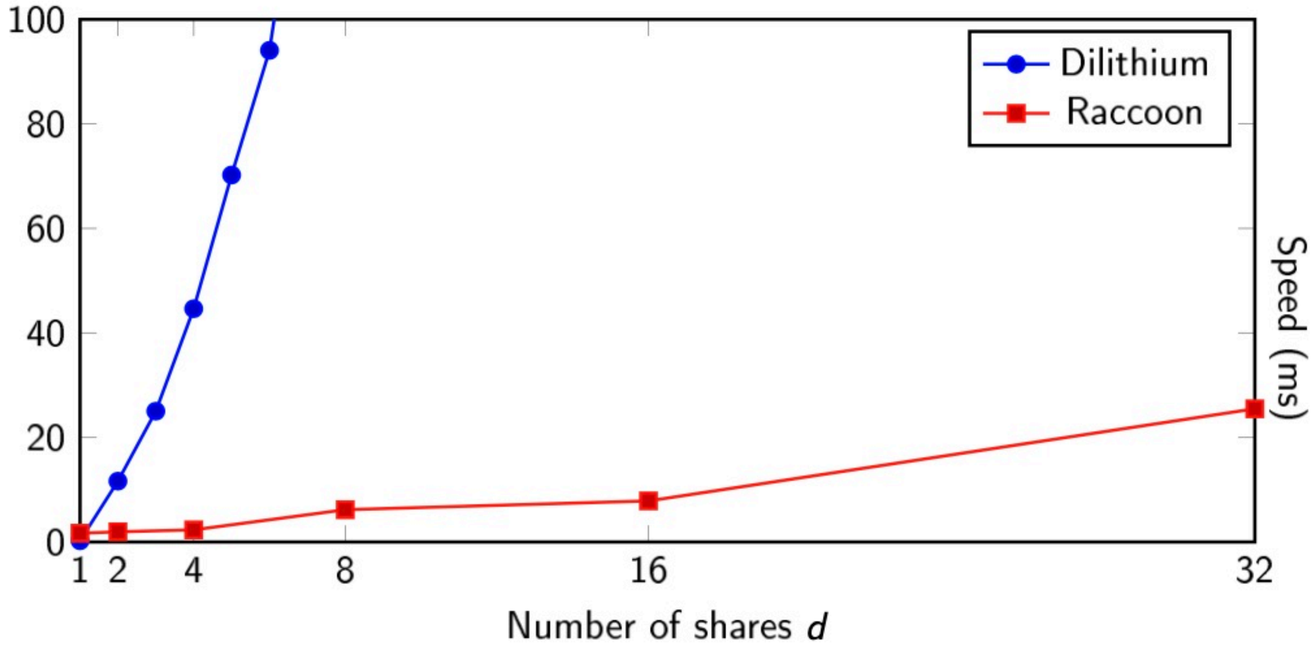
q	5498245831720 97
n	512
k	5
l	4
d	16
T	2

➡ Quasi-linear in the masking order
➡ Proof in the $(d-1)$-probing model
➡ Same assumptions as Dilithium/ML-DSA

Signatures $4\times$ larger

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

Raccoon Signature Scheme



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

Signatures $4\times$ larger

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

Not selected for NIST additional post-quantum signatures (RIP)

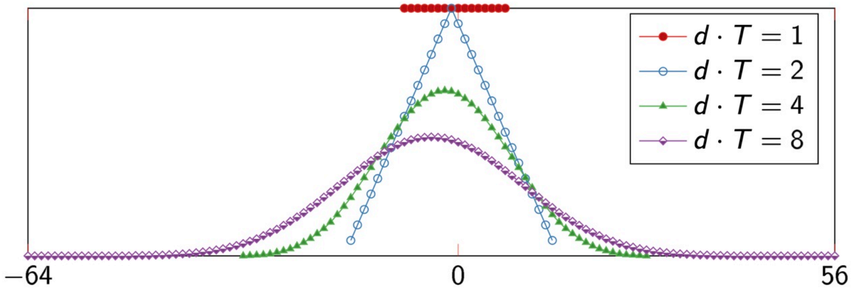
Random Probing Raccoon

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



Distribution of the random that is added

Signature

1. $[\![r]\!] = \mathrm{Refresh}(0,\ldots,0)$
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 $c = \mathrm{H}(w, \mathrm{msg}, \mathrm{vk})$
7. Compute the response $[\![z]\!] = [\![s]\!] \cdot c + [\![r]\!]$
8. Decode $[\![z]\!]$ to z
   No Rejection Sampling
9. The signature is sig = (c, z)

Random Probing Raccoon

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 $[\![s]\!]$

Signature

1. $[\![r]\!] = \mathsf{Refresh}(0,\ldots,0)$
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 $\mathsf{sig} = (\mathsf{c}, \mathsf{z})$

« Add noise to »

Add $d \cdot T$ small uniform randoms

Random Probing Raccoon

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 $[\![s]\!]$

Signature

1. $[\![r]\!] = (0,\ldots,0)$
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 sig = (c, z)

« Add noise to »

Add $d \cdot T$ small uniform randoms

Random Probing Raccoon

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 $[\![s]\!]$

Signature

1. $[\![r]\!] = (0,\ldots,0)$
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 $\mathsf{sig} = (\mathsf{c}, \mathsf{z})$

« Add noise to »

Add $d \cdot T$ small uniform randoms

Random Probing Raccoon

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 $[\![s]\!]$

Signature

1. $[\![r]\!] = (0,\ldots,0)$
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 $\mathsf{sig} = (\mathsf{c}, \mathsf{z})$

A New Notion

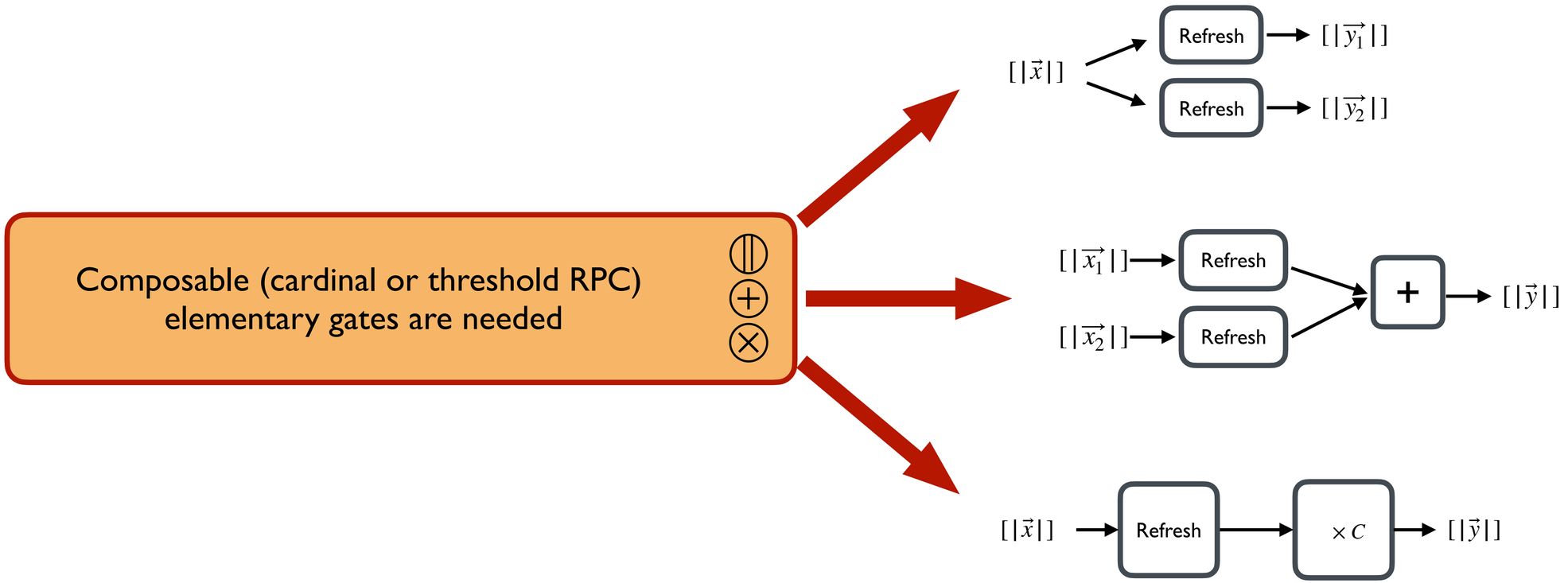
Random Probing Security with Auxiliary Inputs and public Outputs (RPS-AI-O)

« Add noise to »

Add $d \cdot T$ small uniform randoms

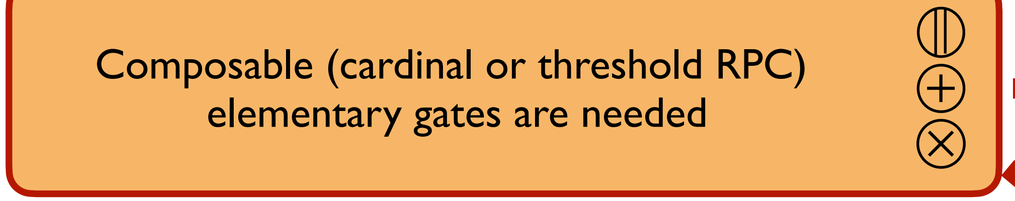
New gadgets

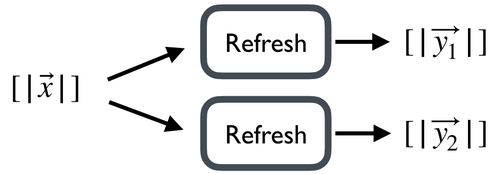
Composable (cardinal or threshold RPC) elementary gates are needed

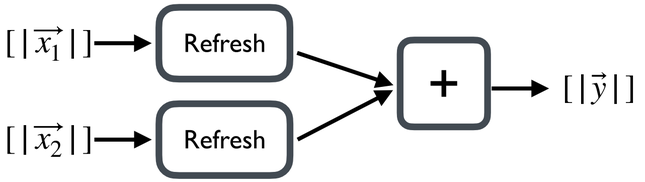


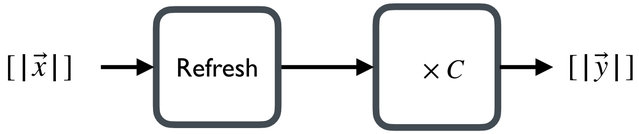
New gadgets

Composable (cardinal or threshold RPC) elementary gates are needed







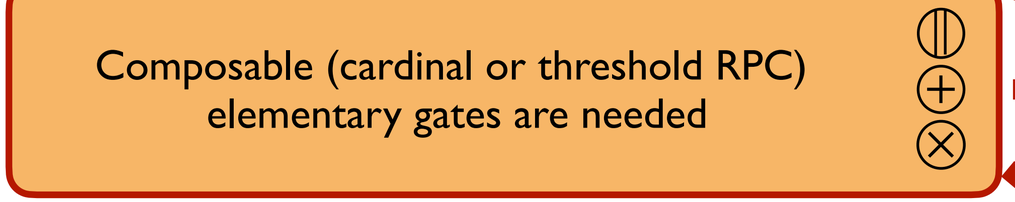


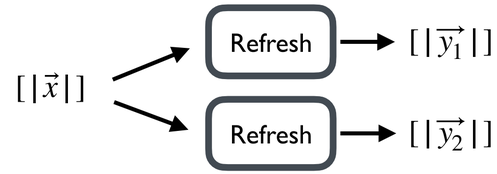
To be composable, they need to include some refreshes

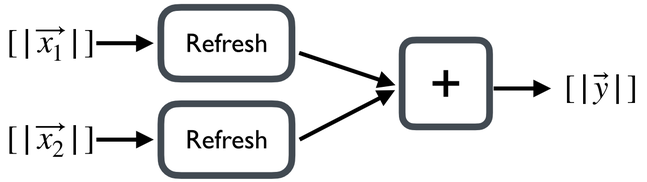
Refresh ?

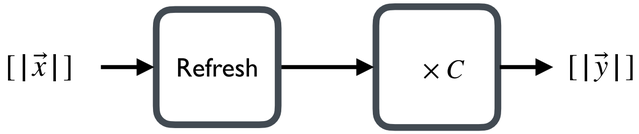
New gadgets

Composable (cardinal or threshold RPC) elementary gates are needed







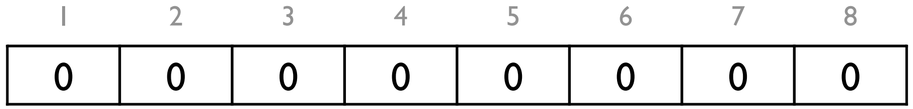


To be composable, they need to include some refreshes

Refresh ?

New Random Probing Composable Refresh

$[|z|] =$



New Random Probing Composable Refresh

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

1st iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	r_1	0	0	0	$-r_1$	0

New Random Probing Composable Refresh

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

1st iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	r_1	0	0	0	$-r_1$	0

2nd iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
r_2	0	r_1	0	0	0	$-r_1$	$-r_2$

New Random Probing Composable Refresh

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

1st iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	r_1	0	0	0	$-r_1$	0

2nd iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
r_2	0	r_1	0	0	0	$-r_1$	$-r_2$

3rd iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
r_2	r_3	$r_1 - r_3$	0	0	0	$-r_1$	$-r_2$

New Random Probing Composable Refresh

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

1st iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
0	0	r_1	0	0	0	$-r_1$	0

2nd iteration

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

$[|z|] =$

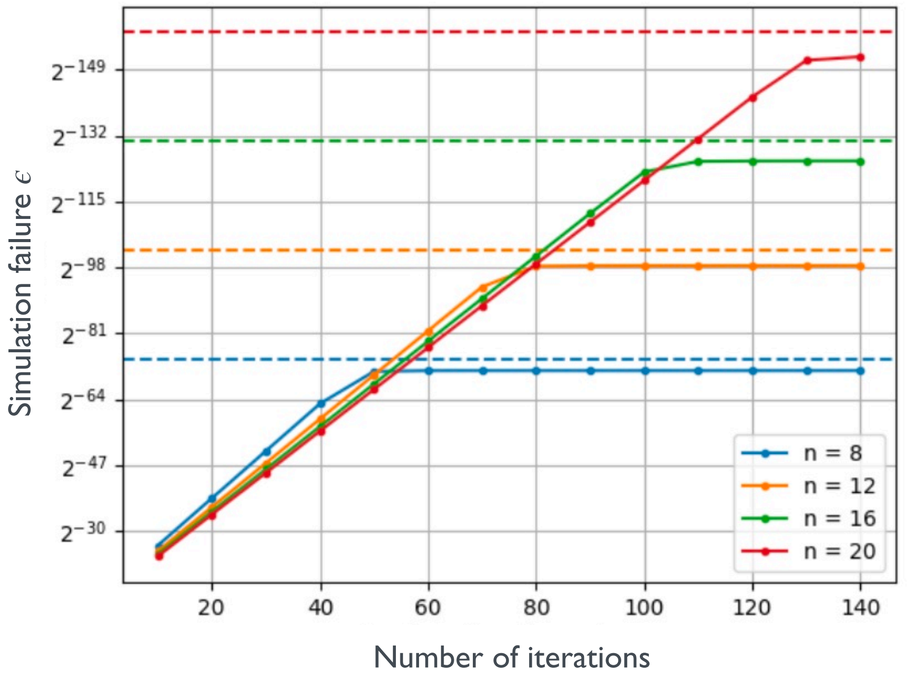
1	2	3	4	5	6	7	8
r_2	0	r_1	0	0	0	$-r_1$	$-r_2$

3rd iteration

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

$[|z|] =$

1	2	3	4	5	6	7	8
r_2	r_3	$r_1 - r_3$	0	0	0	$-r_1$	$-r_2$



RPC-AI advantage of RPRefresh from cardinal-RPC

$p = 2^{-16}$

$t = n/2$

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)

- EUF-CMA secure even if 15 values of each auxiliary inputs 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	$\times 20$	1.82e9	1.02e8	$\times 30$	3.44e9
# linear mult.	8.39e7	$\times 1$	8.39e7	1.01e8	$\times 1$	1.01e8
# randoms	3.60e5	$\times 2000$	6.57e8	5.57e5	$\times 2500$	1.42e9
Security RPS/C	1		2^{-132}	1		2^{-130}

Raccoon 128-16 ($n = 16$ shares)

- EUF-CMA secure even if 15 values of each auxiliary inputs leak
- $p = 2^{-24}$

Current state of the art

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

☑ Composition achievable by combining the enveloppes.

☑ Complexity and penalty factor estimation for Raccoon.

Current state of the art

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

☑ Composition achievable by combining the enveloppes.

☑ Complexity and penalty factor estimation for Raccoon.

Exciting work still lies ahead !

☐ More advanced gadgets
- ➡ Mask conversions, comparisons (secadd)
- ➡ Sampling with specific distributions
- ➡ Quasilinear refresh

☐ Optimized composition for tighter bounds
- ➡ Comparing existing composition techniques

☐ Formal verification

☐ Efficient implementations

[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