A Quest for Provable Security against Side-Channel Attacks

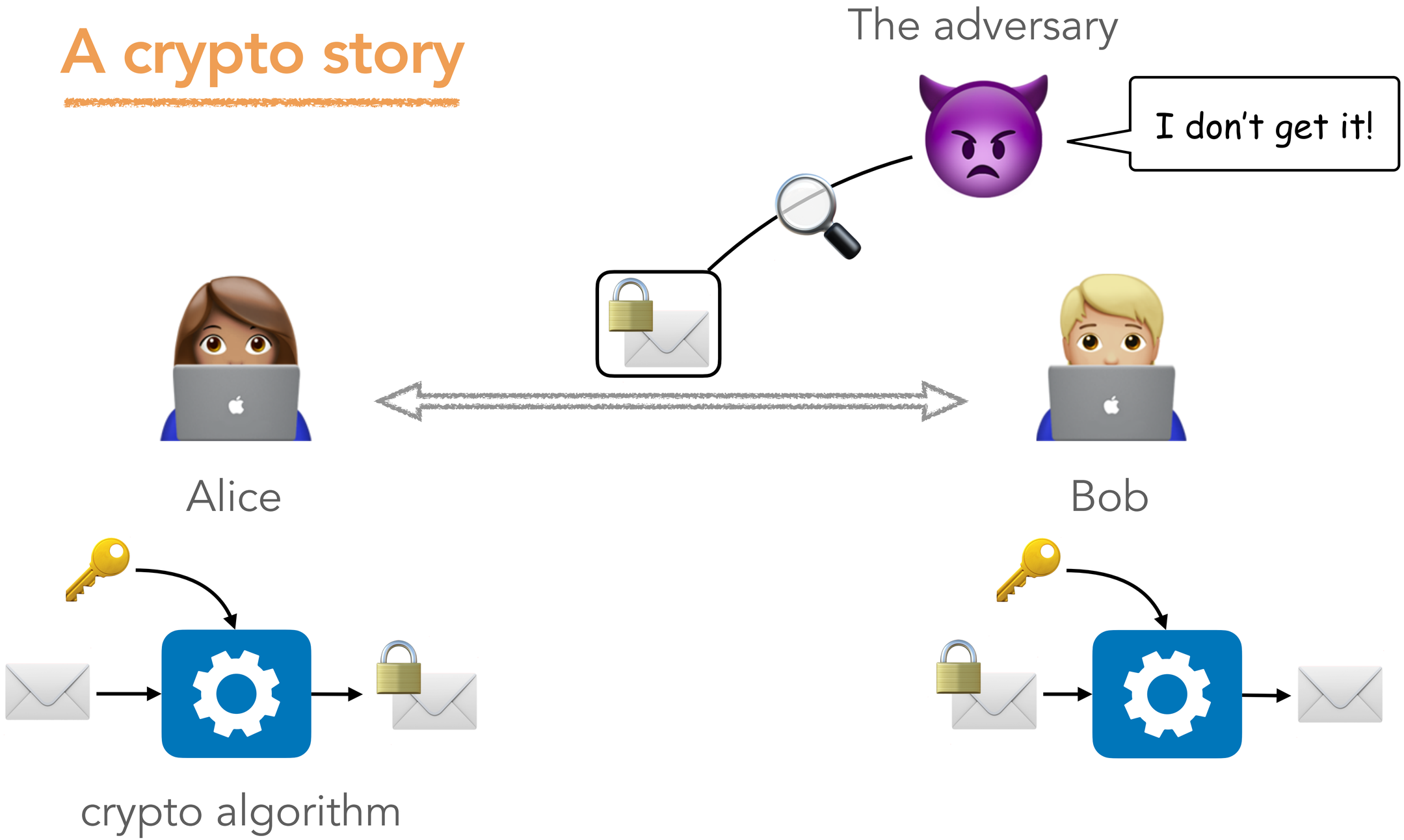
Matthieu Rivain

Cyber in Nancy

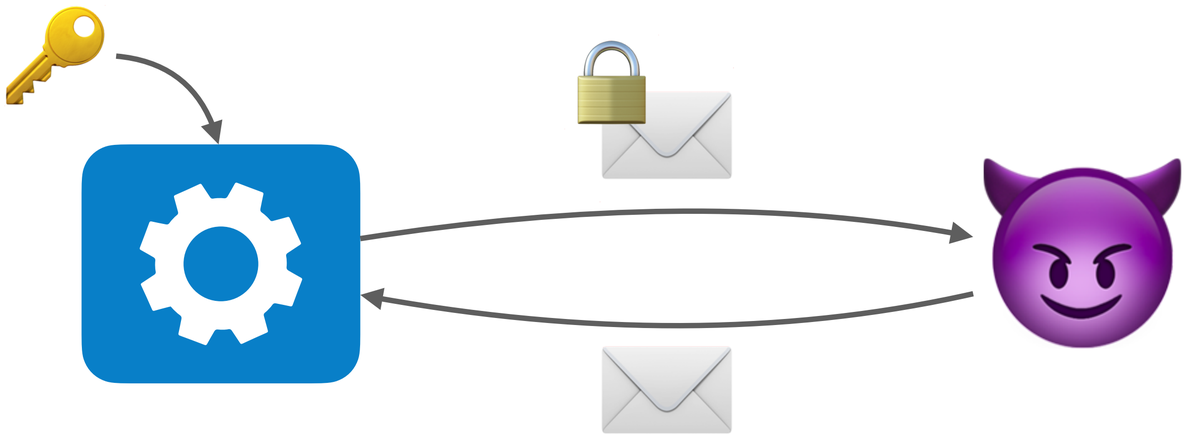
July 7, 2022, Loria, Nancy

Introduction

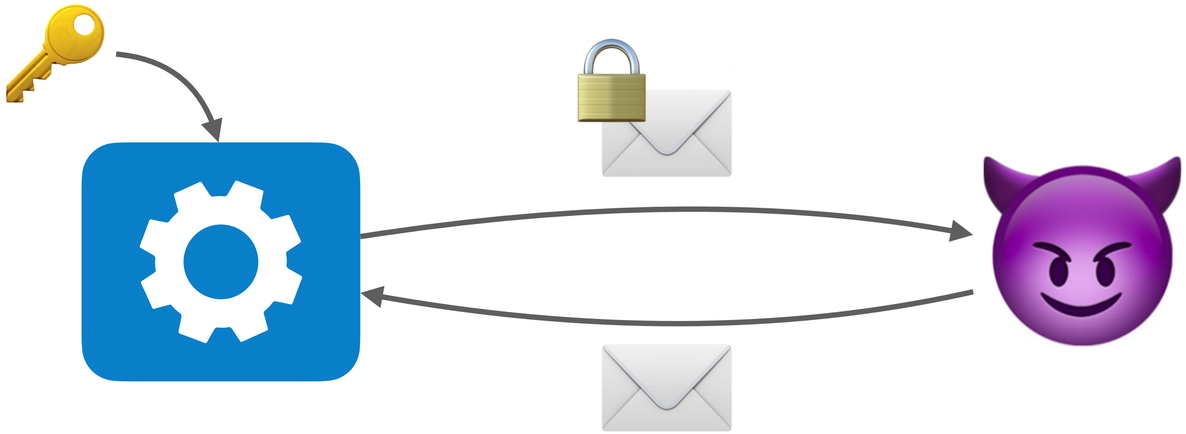
A crypto story



Provable security



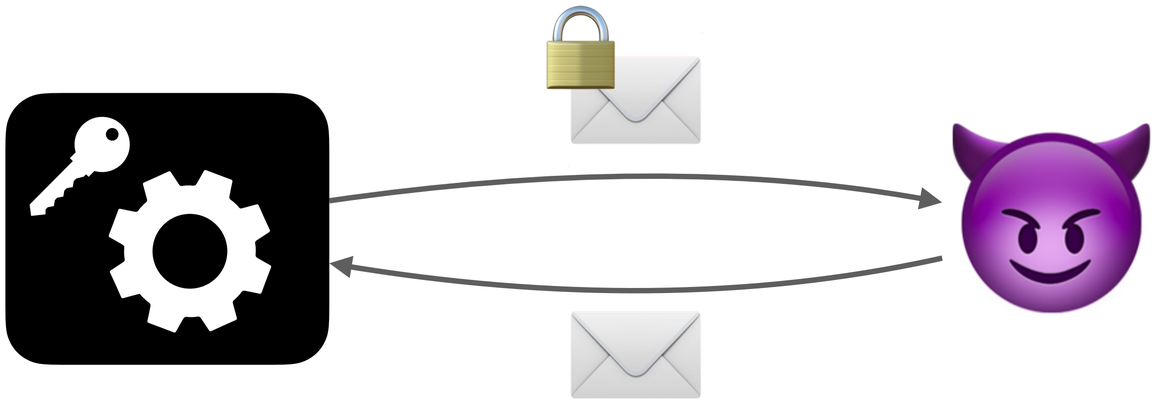
Provable security



😈 needs unaffordable computing power to recover 🔑

security proof

Provable security

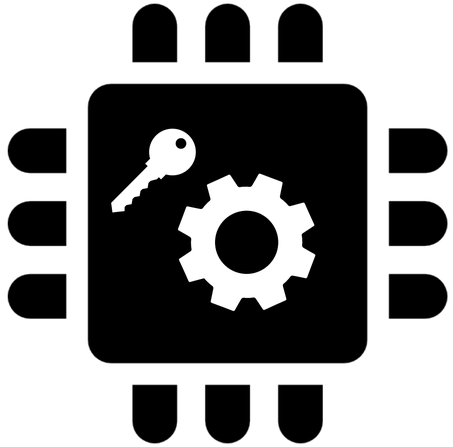


The “black-box model”

😈 needs unaffordable computing power to recover 🔑

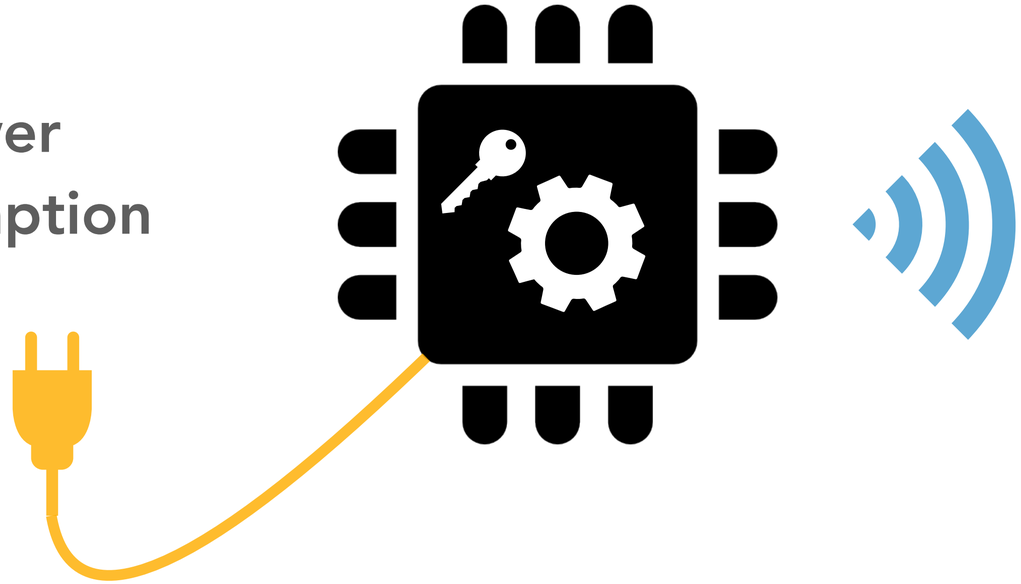
security proof

Side-channel attacks



Side-channel attacks

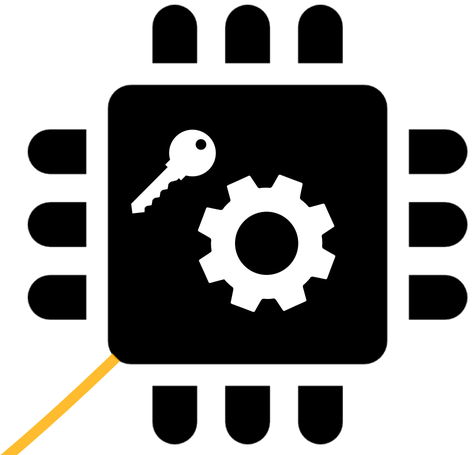
Power consumption

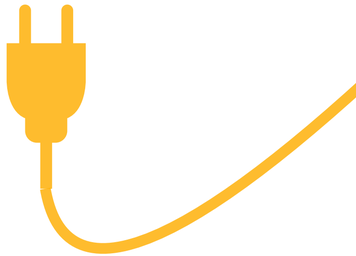


Electromagnetic emanations

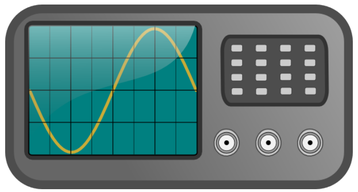
Side-channel attacks

Power consumption

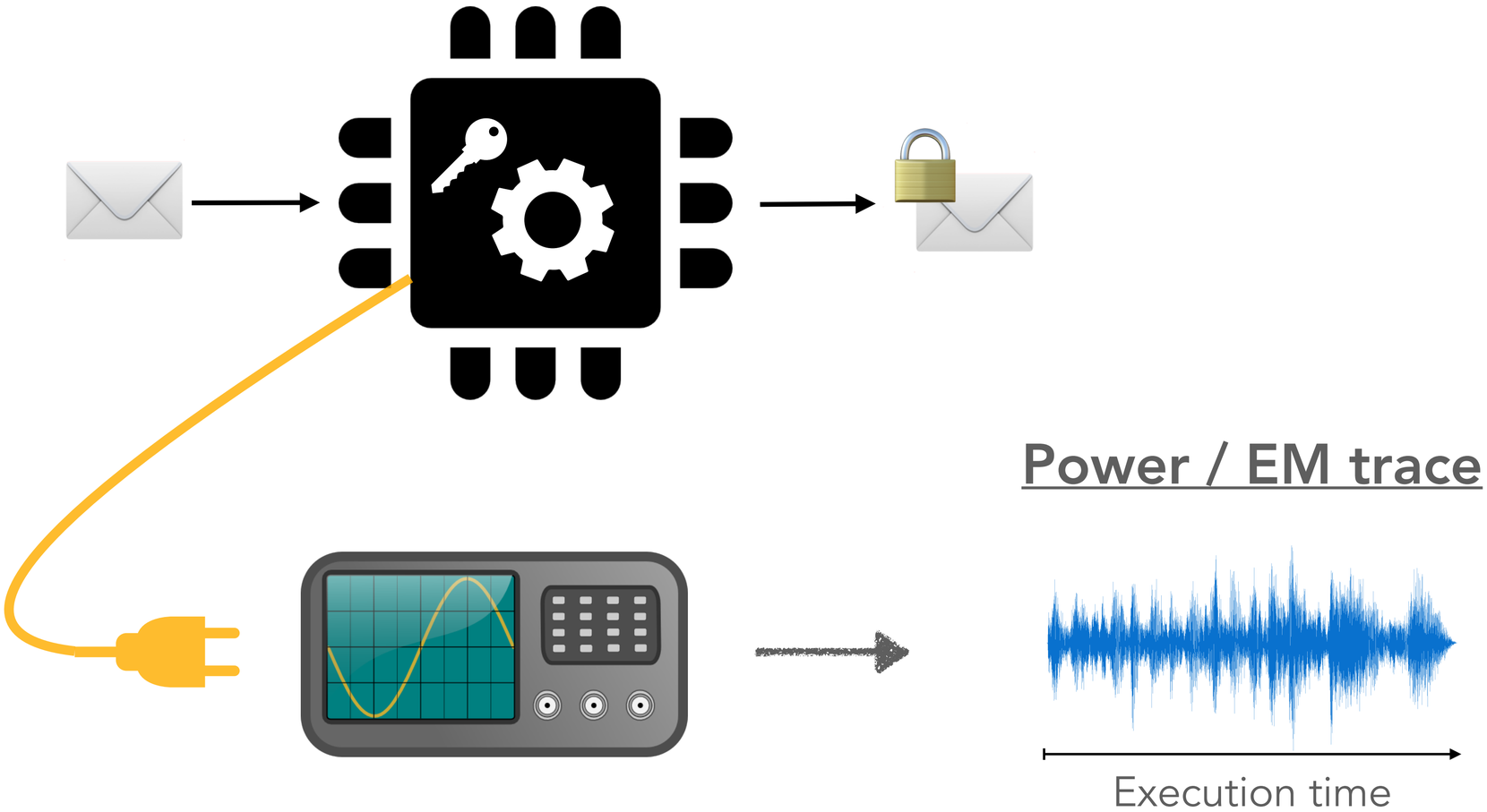




Electromagnetic emanations



Differential power analysis



Differential power analysis

Key guess

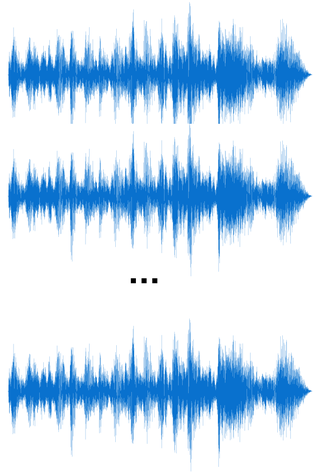
$f($ ✉️ , 🔑 $)$

$f($ ✉️ , 🔑 $)$

…

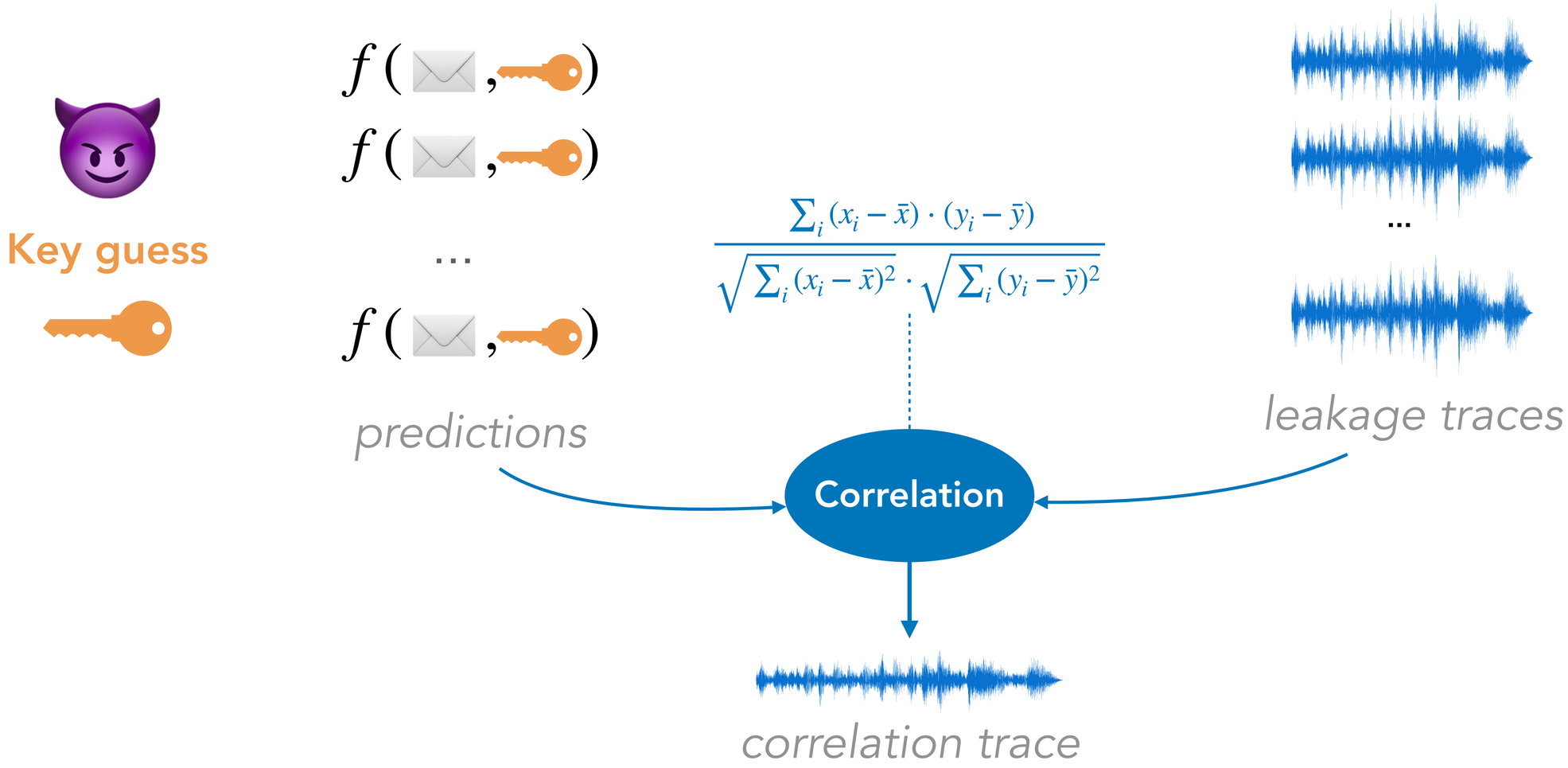
$f($ ✉️ , 🔑 $)$

predictions

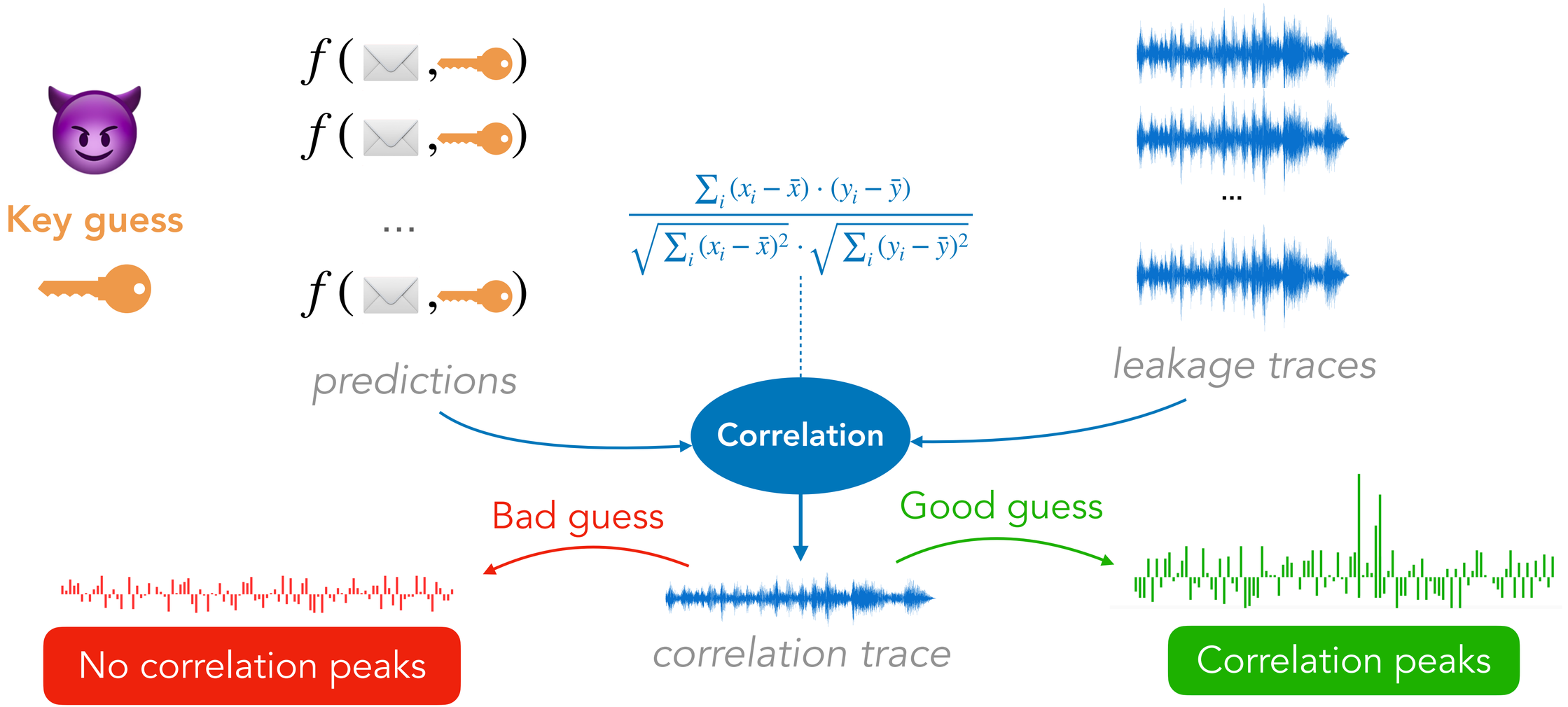


leakage traces

Differential power analysis

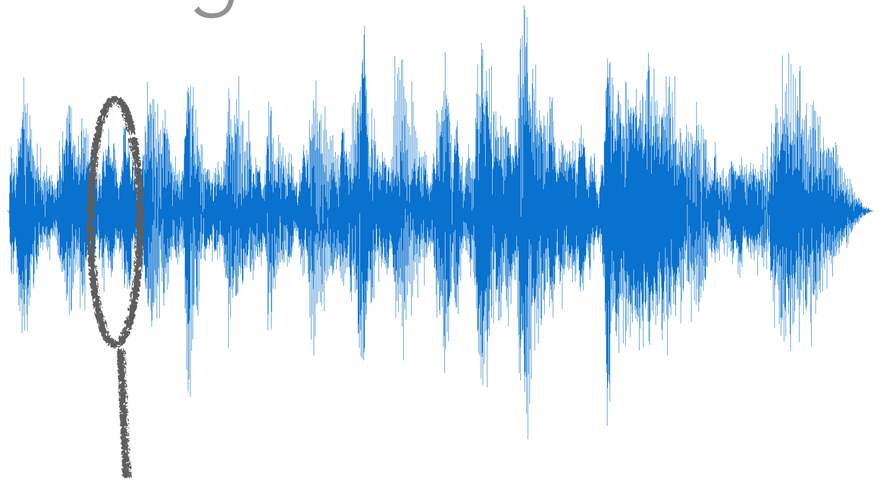


Differential power analysis



Example: DPA on AES

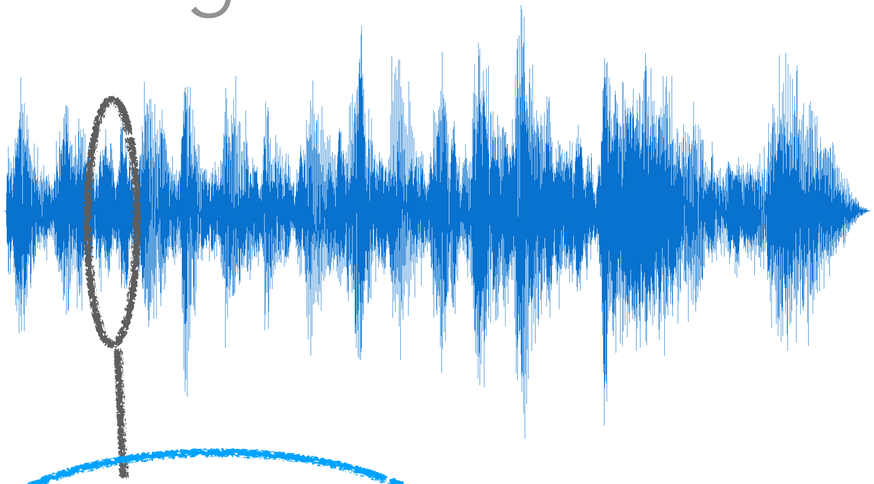
Leakage model



$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$

Example: DPA on AES

Leakage model

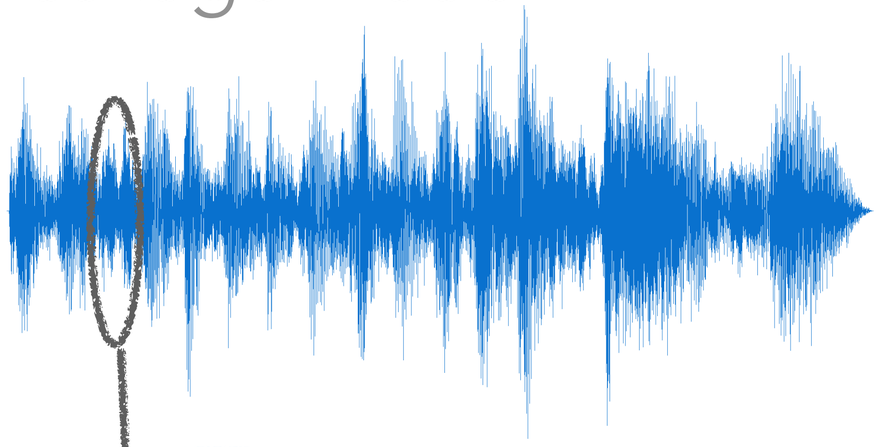


$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Example: DPA on AES

Leakage model



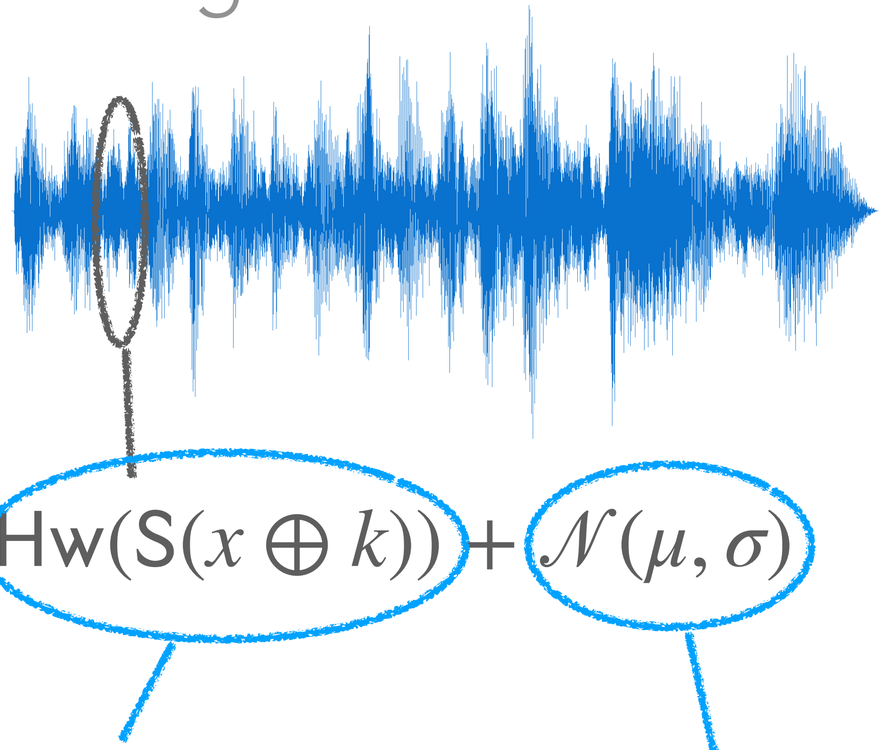
$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Gaussian noise

Example: DPA on AES

Leakage model



$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

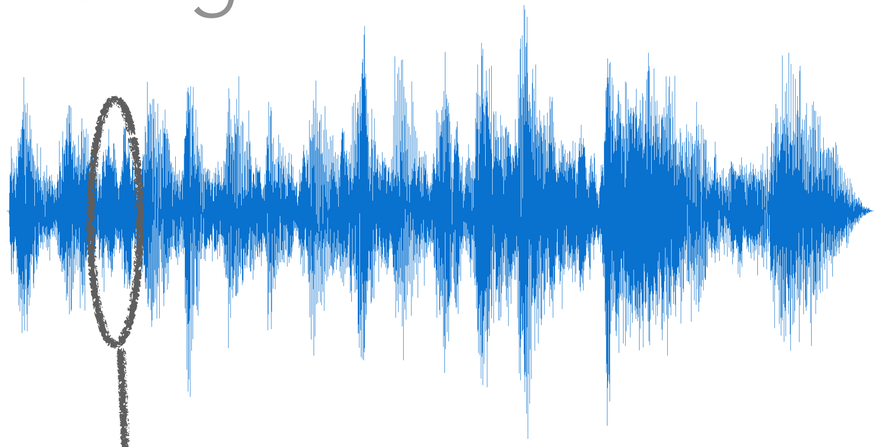
Gaussian noise

Prediction

$$f(\text{✉}, \text{🔑}) = \mathrm{Hw}(\mathrm{S}(x \oplus \hat{k}))$$

Example: DPA on AES

Leakage model



$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Gaussian noise

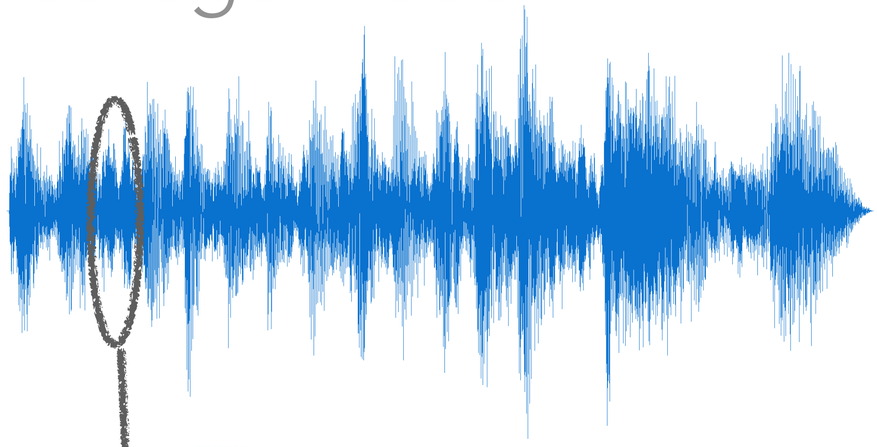
Prediction

(known) byte of the plaintext

$$f(\text{✉}, \text{🔑}) = \mathrm{Hw}(\mathrm{S}(x \oplus \hat{k}))$$

Example: DPA on AES

Leakage model



$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Gaussian noise

Prediction

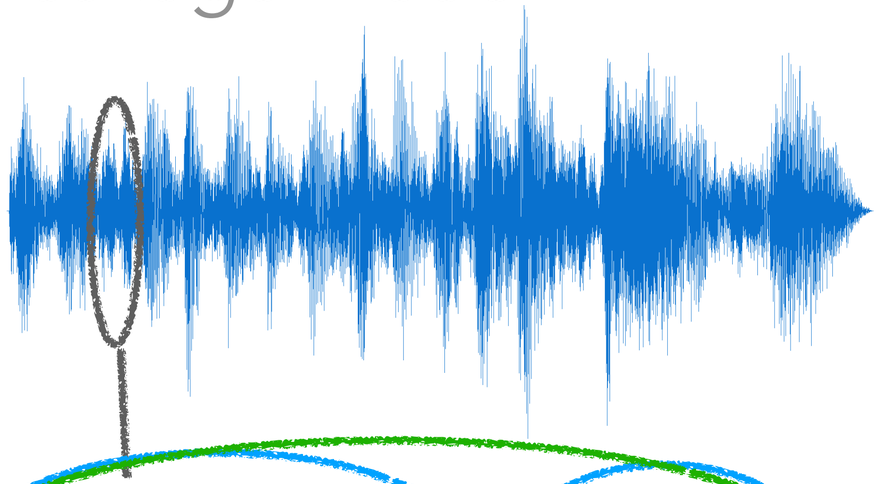
(known) byte of the plaintext

$$f(\text{✉}, \text{🔑}) = \mathrm{Hw}(\mathrm{S}(x \oplus \hat{k}))$$

(guess) byte of the key

Example: DPA on AES

Leakage model



$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Gaussian noise

Prediction

(known) byte of the plaintext

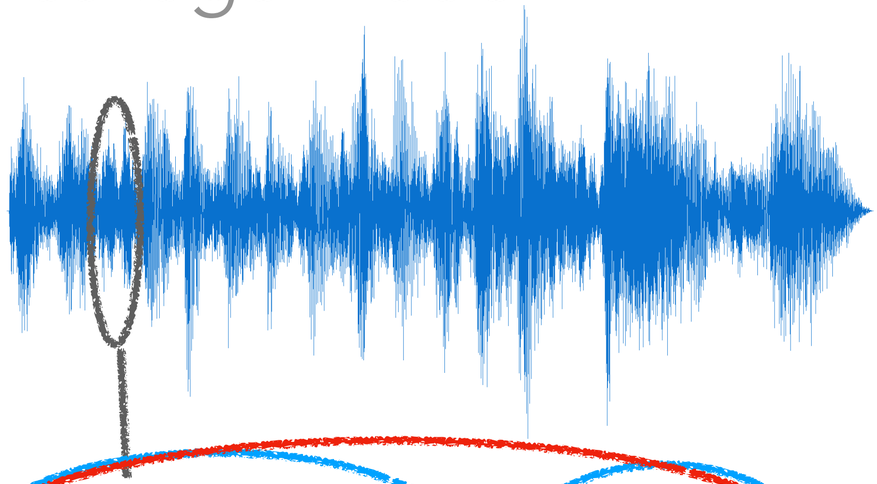
$$f(\text{✉}, \text{🔑}) = \mathrm{Hw}(\mathrm{S}(x \oplus \hat{k}))$$

(guess) byte of the key

high correlation for the good guess

Example: DPA on AES

Leakage model



$$\sim \mathrm{Hw}(\mathrm{S}(x \oplus k)) + \mathcal{N}(\mu, \sigma)$$

Hamming weight of s-box output

Gaussian noise

Prediction

(known) byte of the plaintext

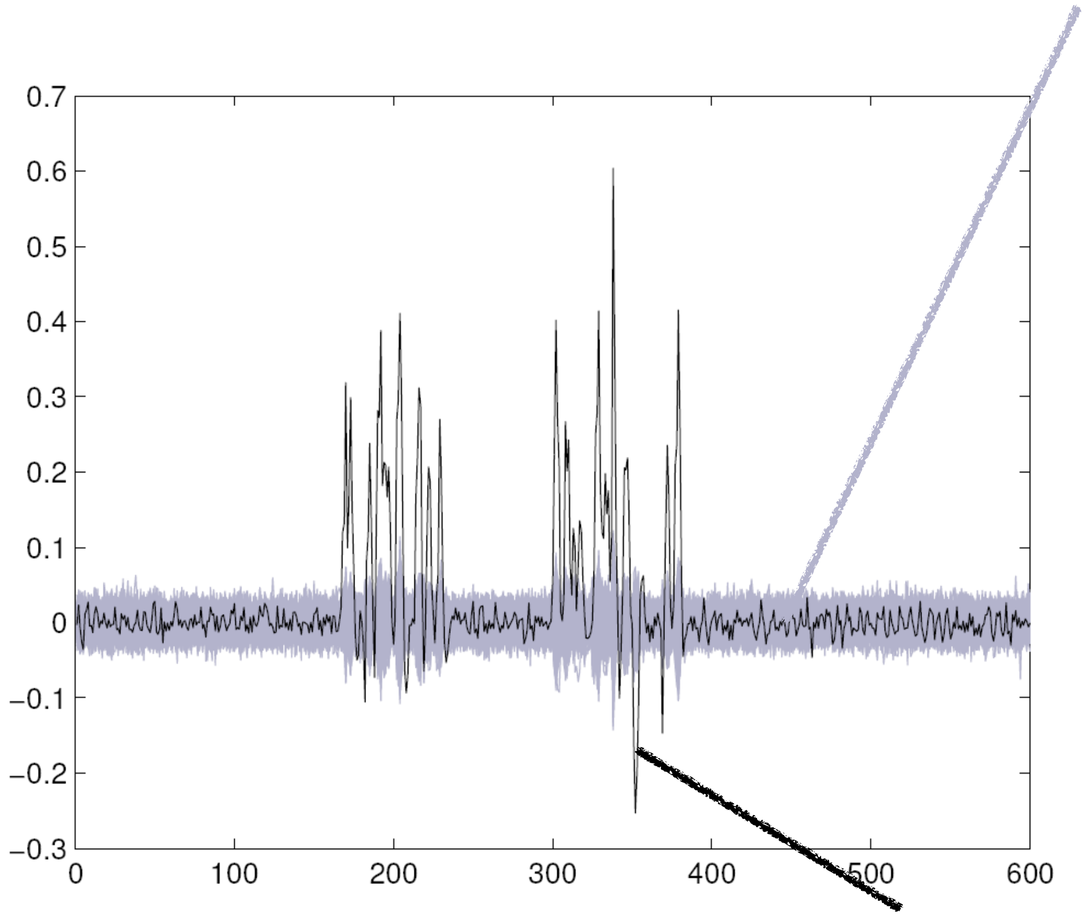
$$f(\text{✉}, \text{🔑}) = \mathrm{Hw}(\mathrm{S}(x \oplus \hat{k}))$$

(guess) byte of the key

low correlation for the wrong guess

Example: DPA on AES

correlation traces for wrong guesses



correlation trace for good guess

Masking

Apply secret sharing at the computation level

$$x = x_1 \oplus \cdots \oplus x_n$$

Masking

Apply secret sharing at the computation level

$$x = x_1 \oplus \cdots \oplus x_n$$

randomly generated

Masking

Apply secret sharing at
the computation level

$$x = x_1 \oplus \cdots \oplus x_n$$

randomly generated

constrained to keep the correctness

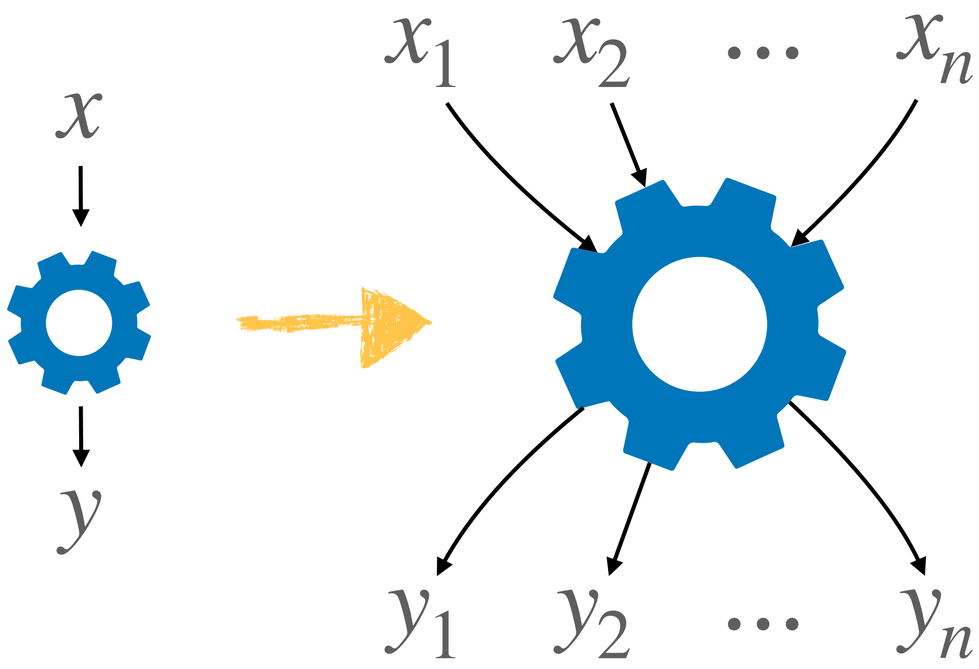
Masking

Apply secret sharing at the computation level

$$x = x_1 \oplus \cdots \oplus x_n$$

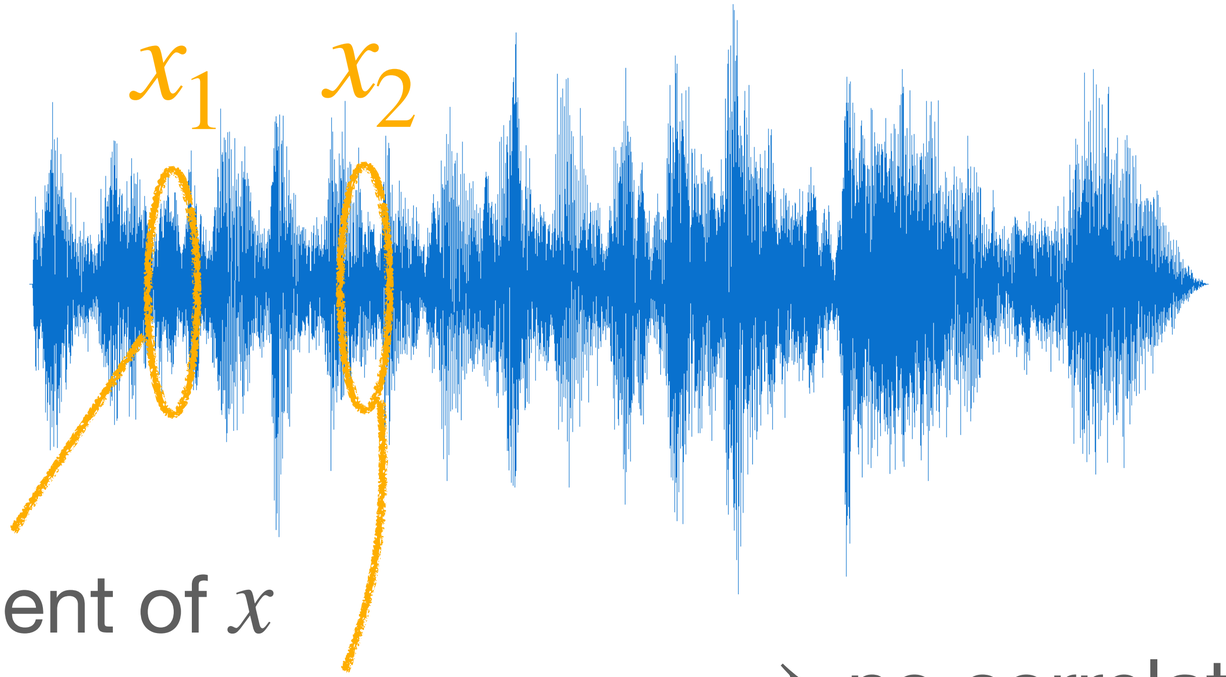
randomly generated

constrained to keep the correctness



Masking

First order masking: $x = x_1 \oplus x_2$



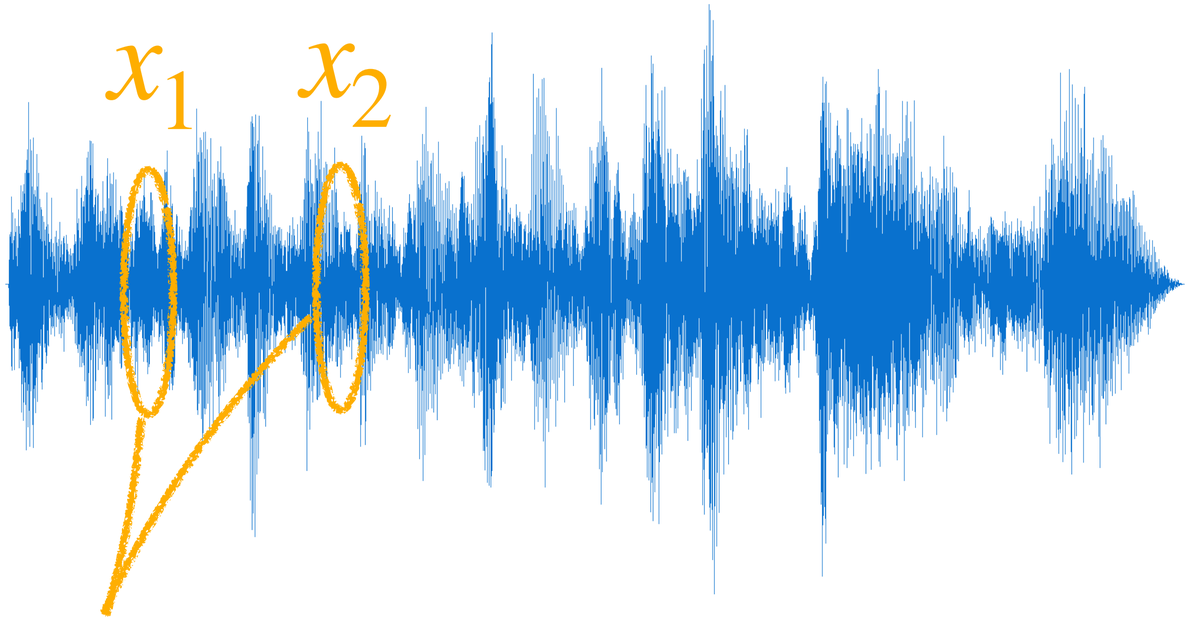
independent of x

independent of x

$\Rightarrow$ no correlation (DPA fails)

Masking

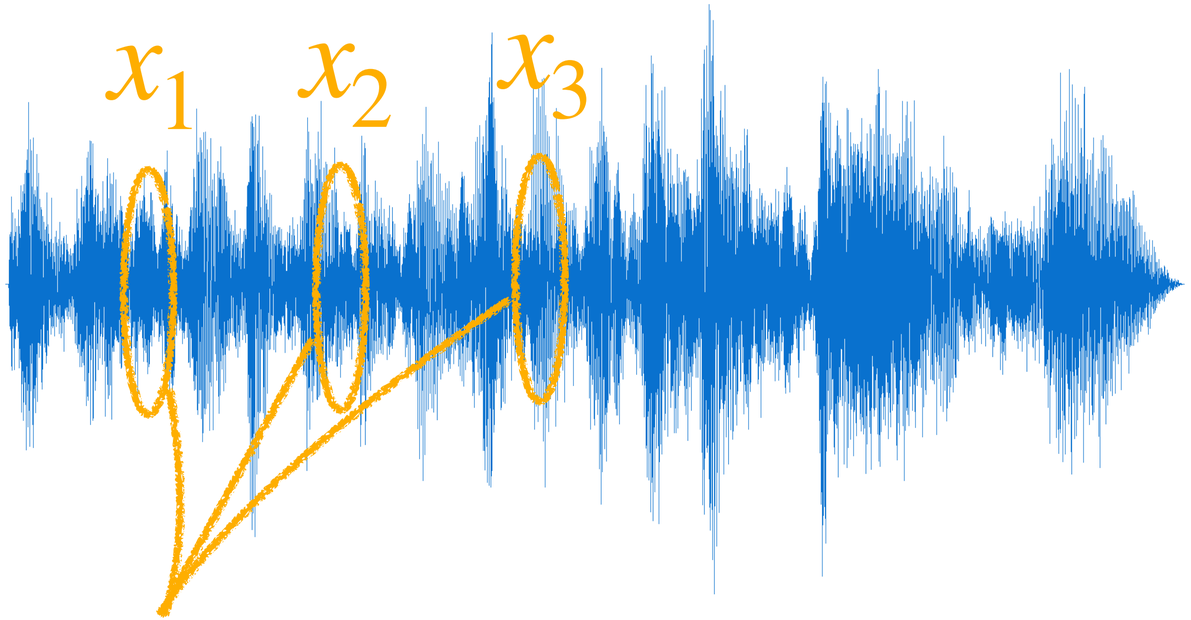
First order masking: $x = x_1 \oplus x_2$



⚠ jointly depend on x ⇒ **2nd order** DPA

Masking

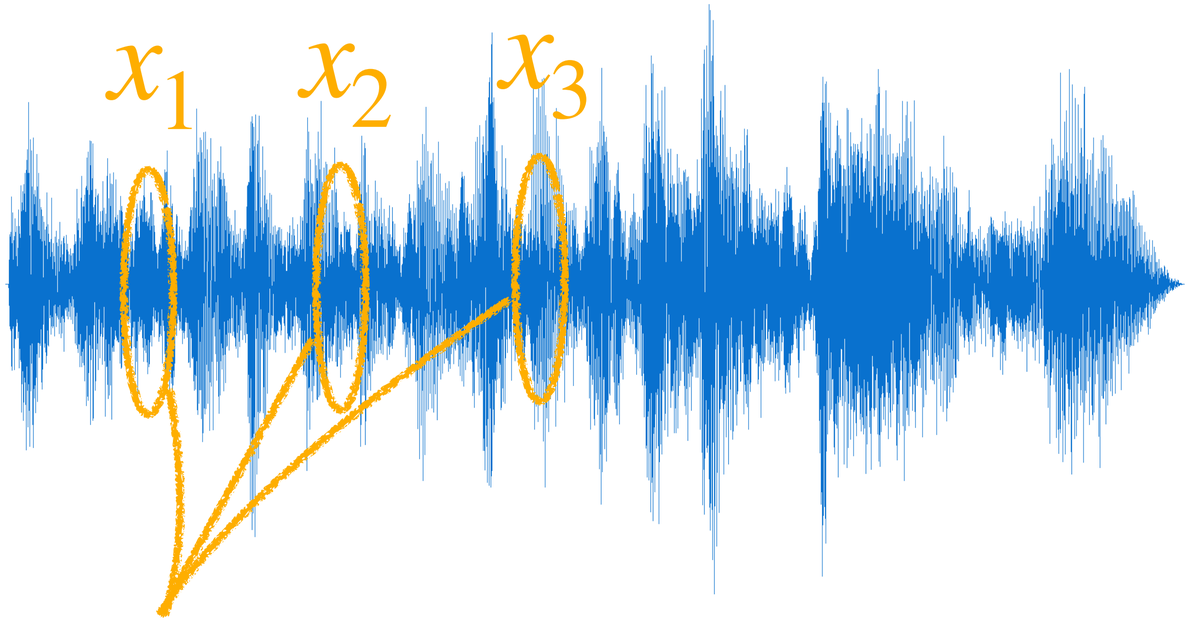
Second order masking: $x = x_1 \oplus x_2 \oplus x_3$



⚠ jointly depend on x $\Rightarrow$ **3rd order** DPA

Masking

Second order masking: $x = x_1 \oplus x_2 \oplus x_3$

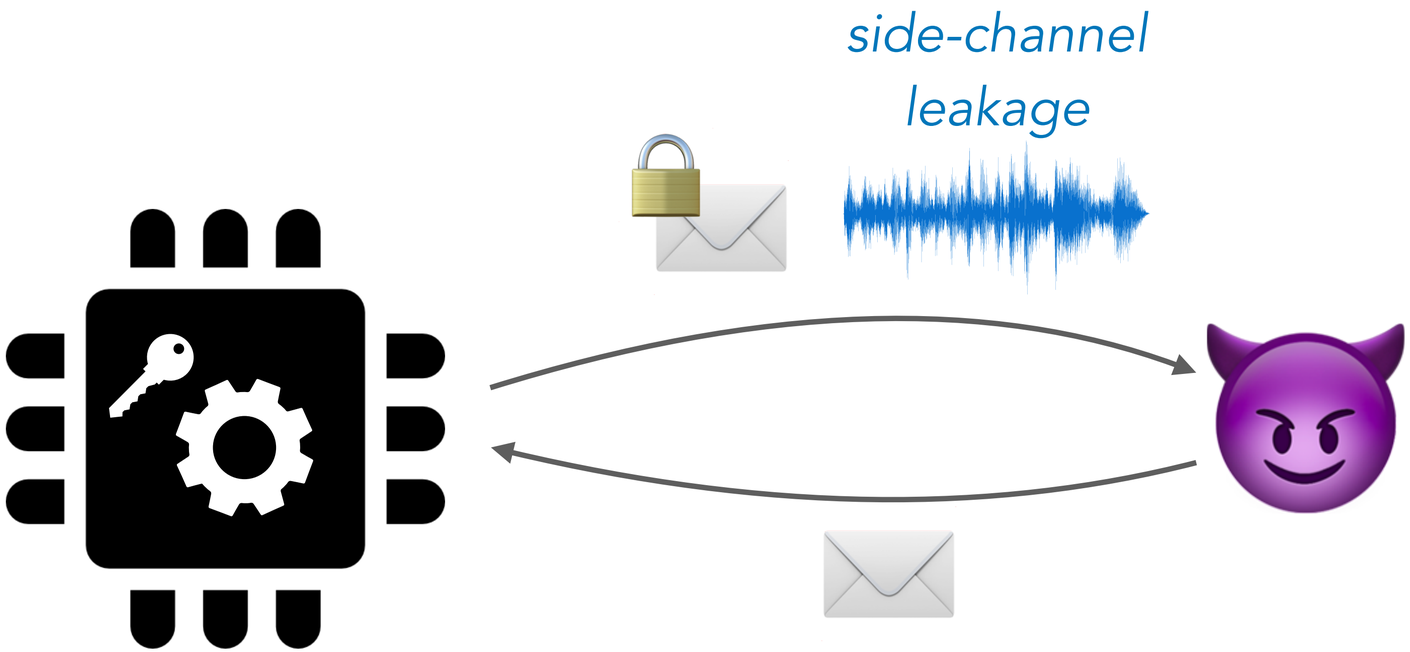



/


game!

⚠ jointly depend on x ⇒ **3rd order** DPA

Provable security in the presence of leakage

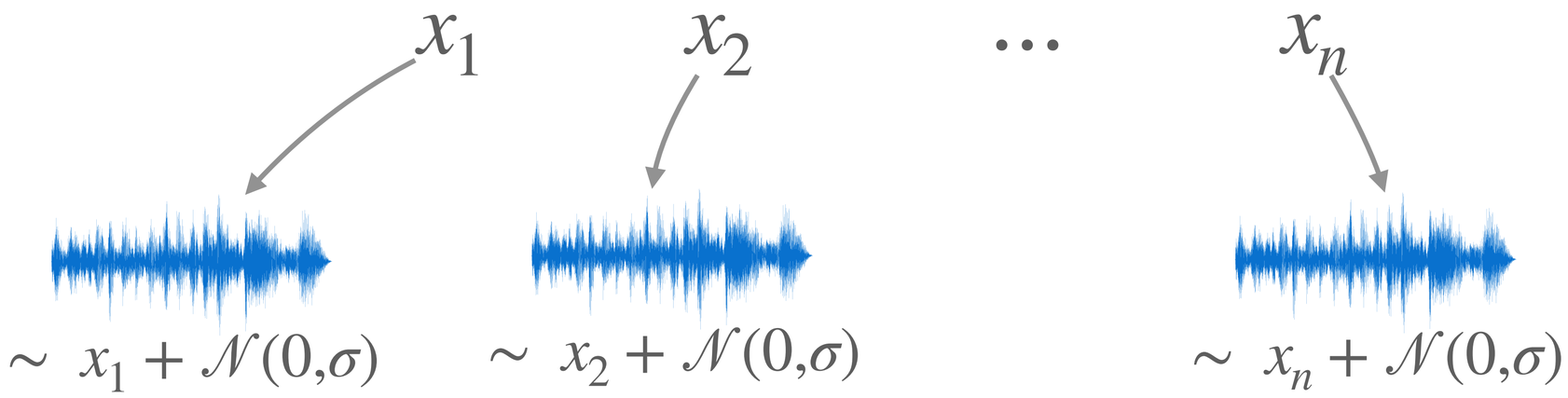


😈 needs unaffordable computing power to recover 🔑

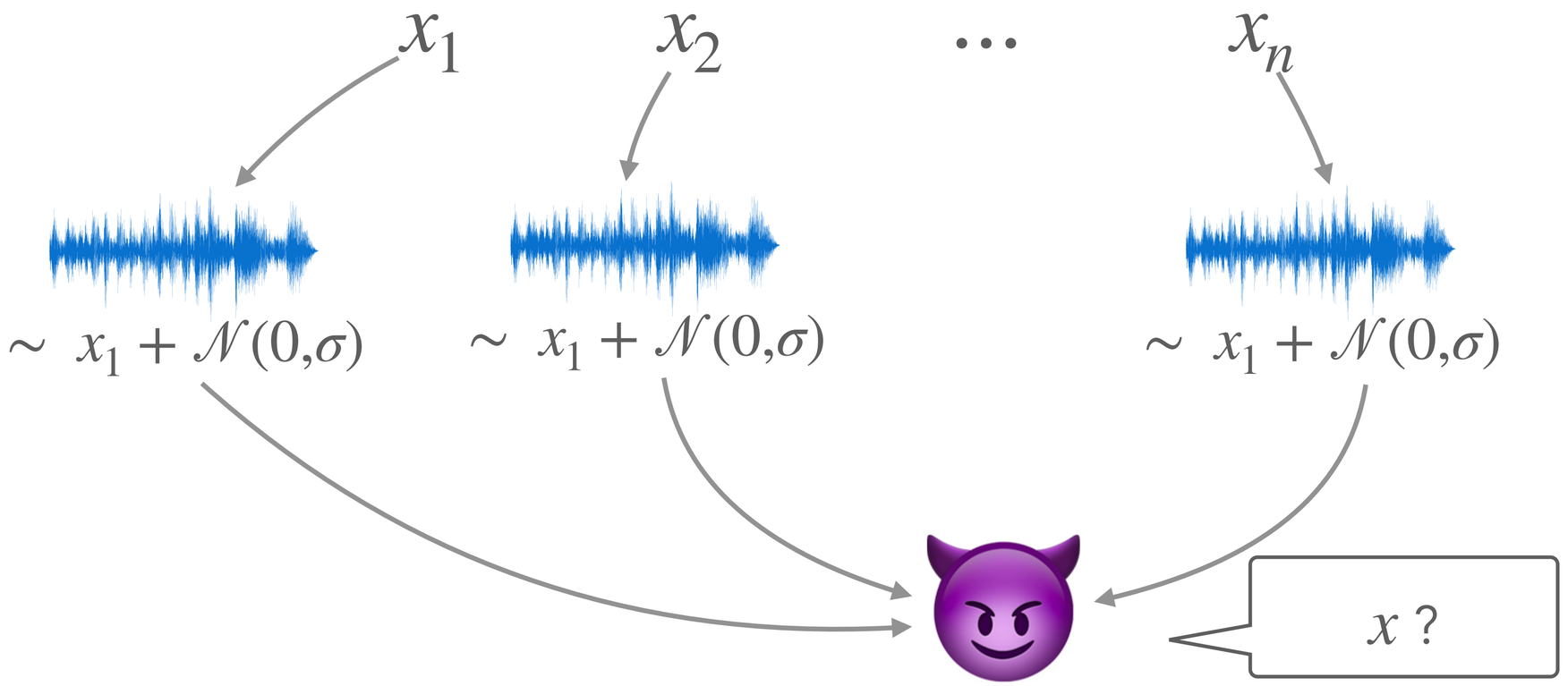
security proof

Modelling noisy leakage

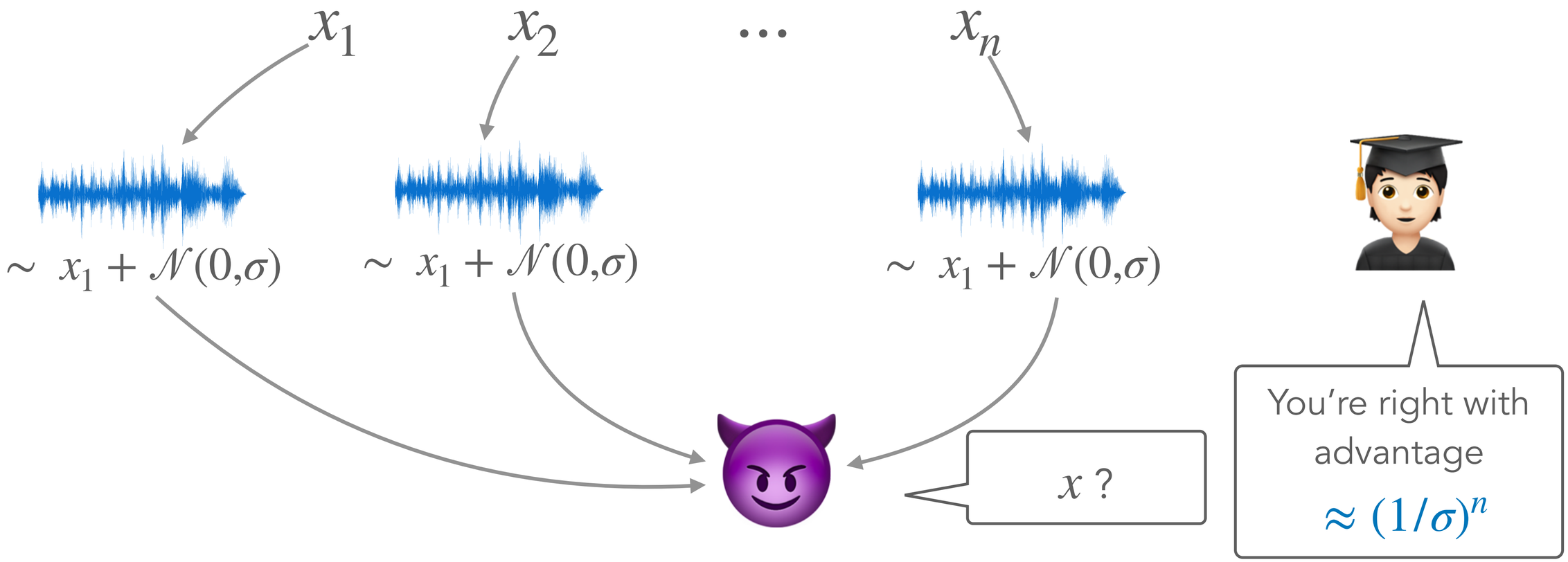
Motivation: soundness of masking with noise



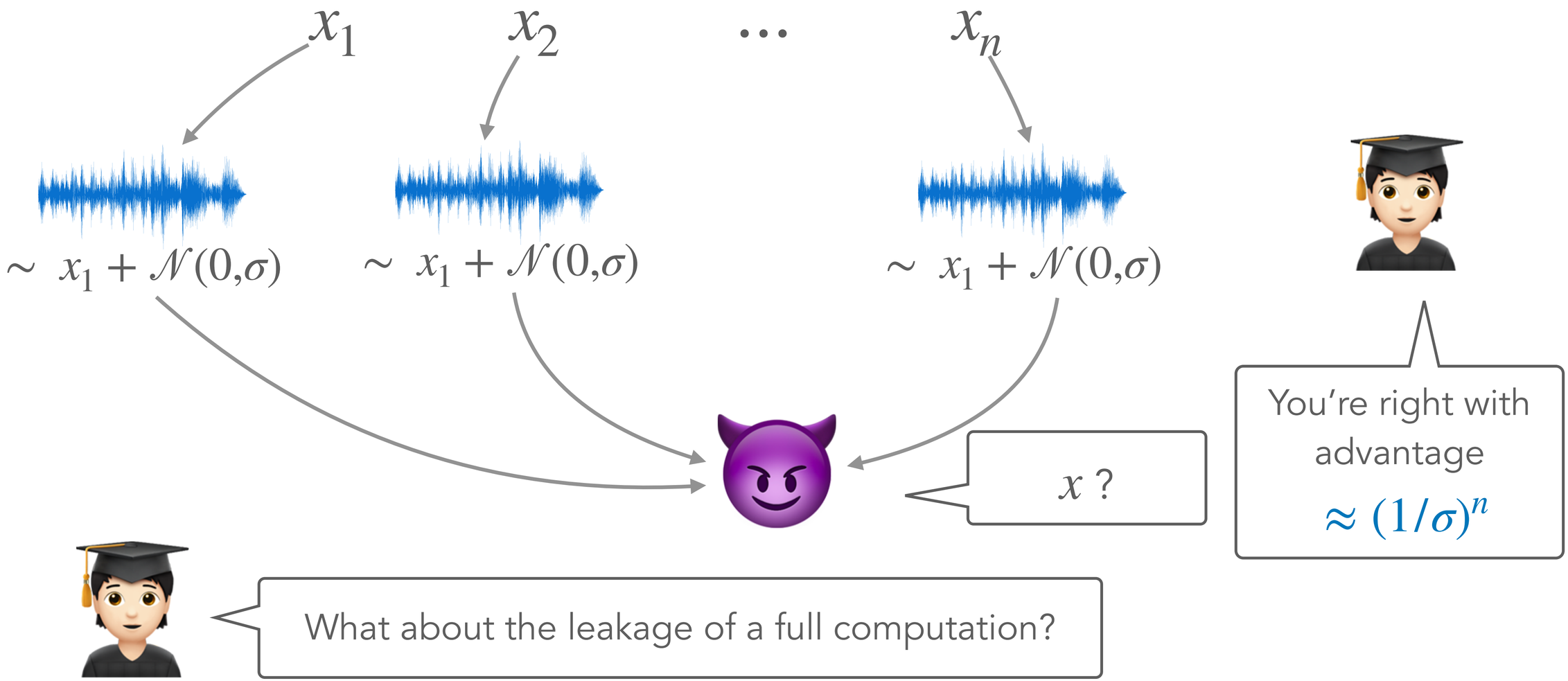
Motivation: soundness of masking with noise



Motivation: soundness of masking with noise



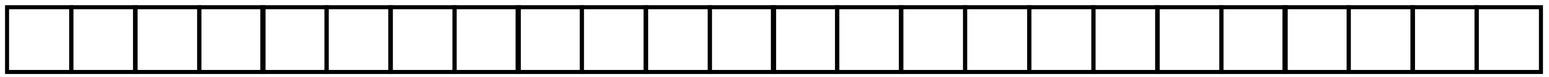
Motivation: soundness of masking with noise



The noisy leakage model

Assumption: “only computation leaks”

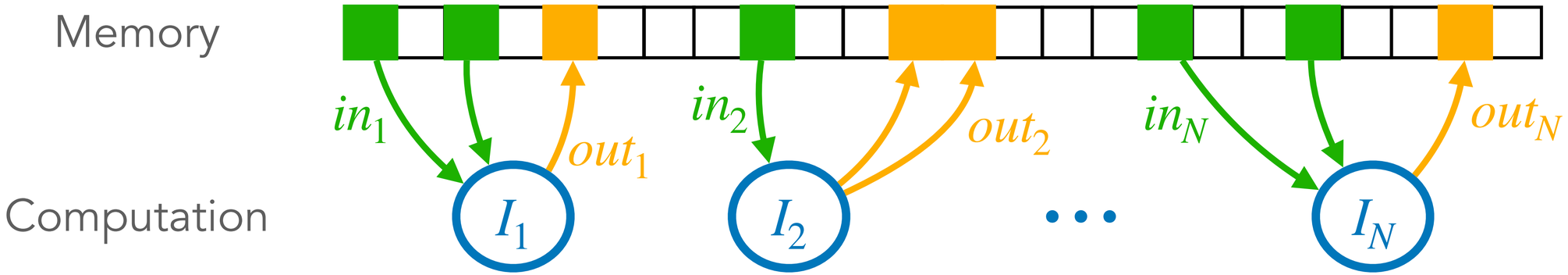
Memory



Computation

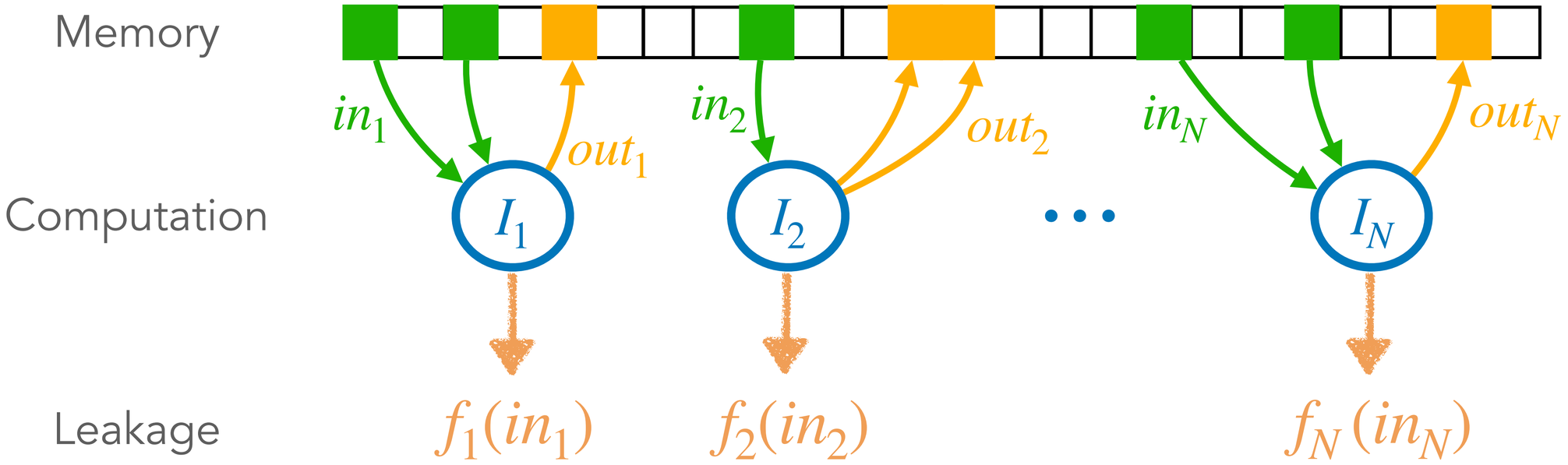
The noisy leakage model

Assumption: “only computation leaks”



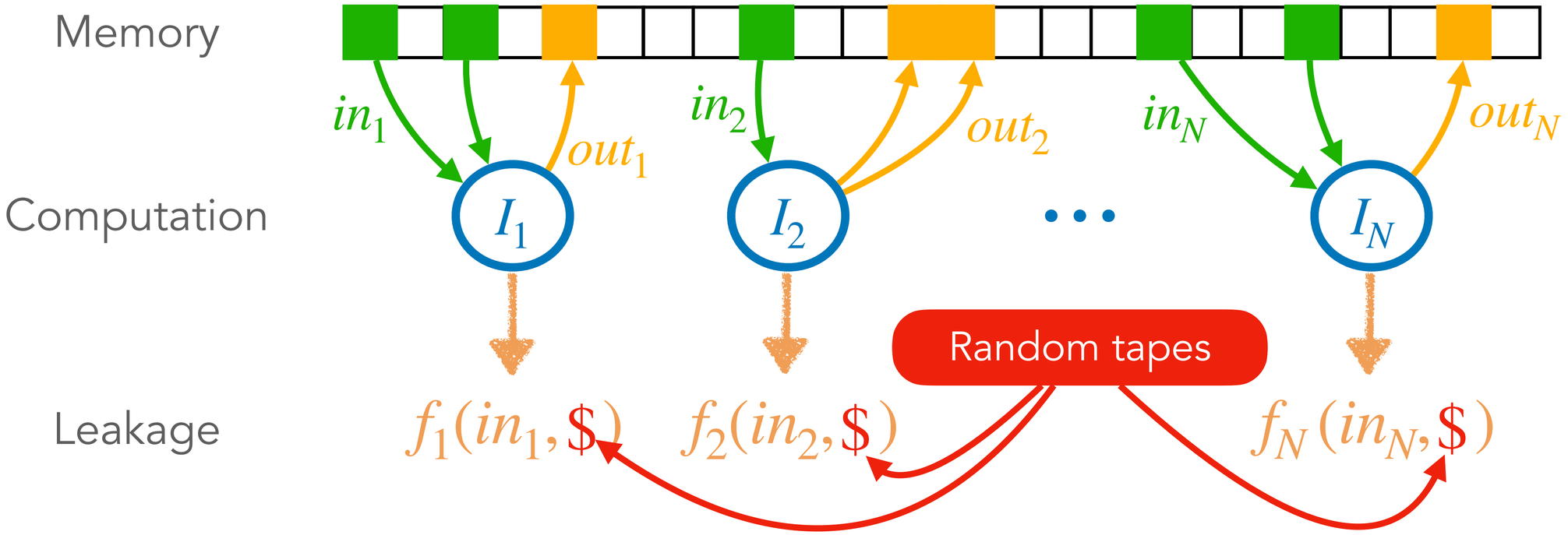
The noisy leakage model

Assumption: “only computation leaks”

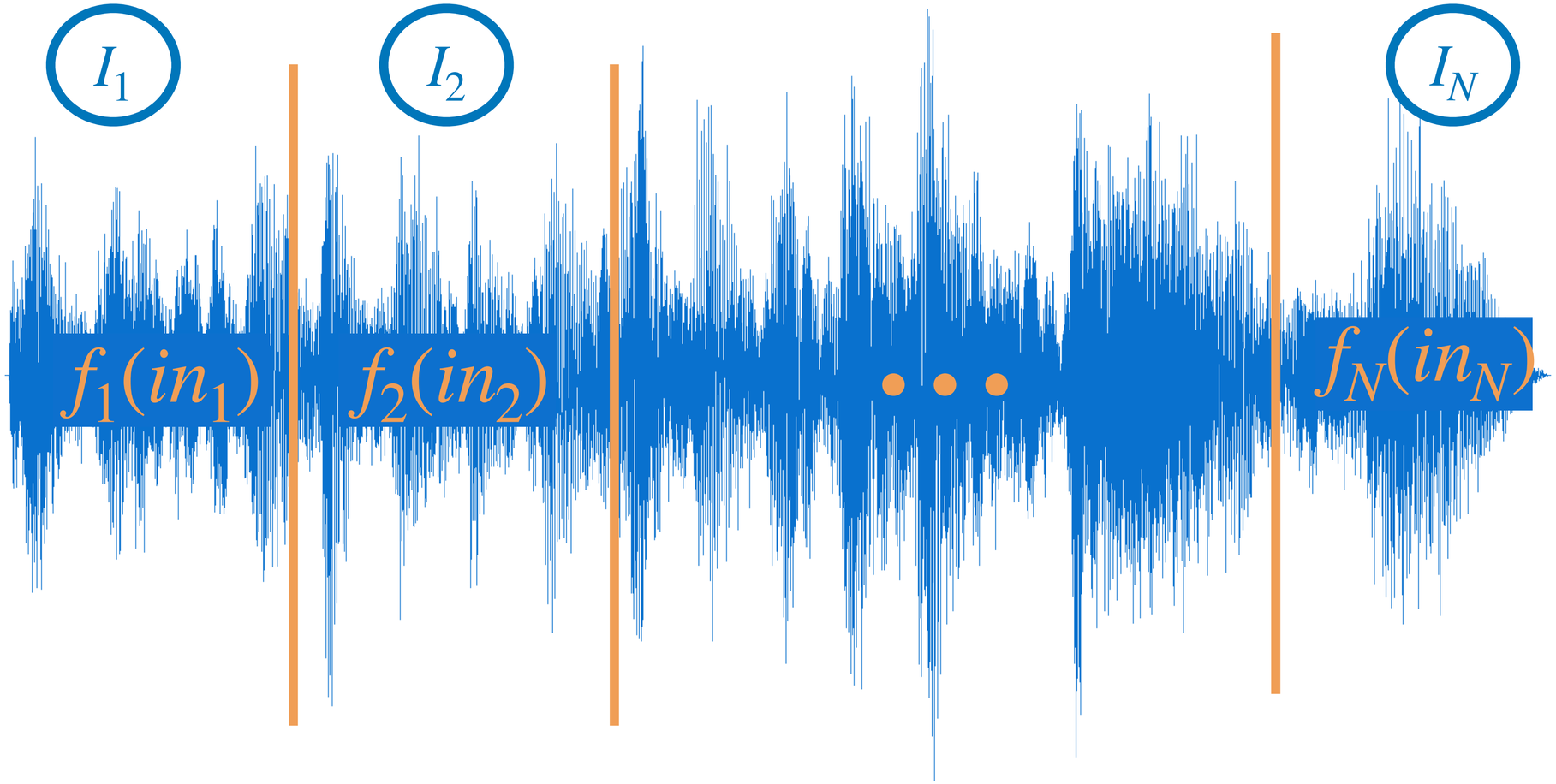


The noisy leakage model

Assumption: “only computation leaks”



The noisy leakage model



$f_i(in_i) \Rightarrow$ *multivariate noisy leakage*

Noisy leakage functions

A function is δ-**noisy** if (for $X \sim \mathcal{U}$):

$$\mathbb{E}_y\left[\Delta\left(X;\ \left(X \mid f(X) = y\right)\right)\right] \ \leq \ \delta$$

Noisy leakage functions

A function is δ-**noisy** if (for $X \sim \mathcal{U}$):

$$\mathbb{E}_y\left[\Delta\left(X;\ \left(X \mid f(X) = y\right)\right)\right] \ \leq \ \delta$$

Noisy leakage functions

A function is δ-**noisy** if (for $X \sim \mathcal{U}$):

$$\mathbb{E}_y\left[\Delta\left(X;\ \left(X \mid f(X) = y\right)\right)\right] \ \leq \ \delta$$

statistical distance between X and X and given $f(X) = y$

Noisy leakage functions

A function is δ-**noisy** if (for $X \sim \mathcal{U}$):

$$\mathbb{E}_y\left[\Delta\left(X;\ \left(X \mid f(X) = y\right)\right)\right] \leq \delta$$

expectation on the possible leakage values

statistical distance between X and X and given $f(X) = y$

Noisy leakage functions

more noise
⇒ smaller δ

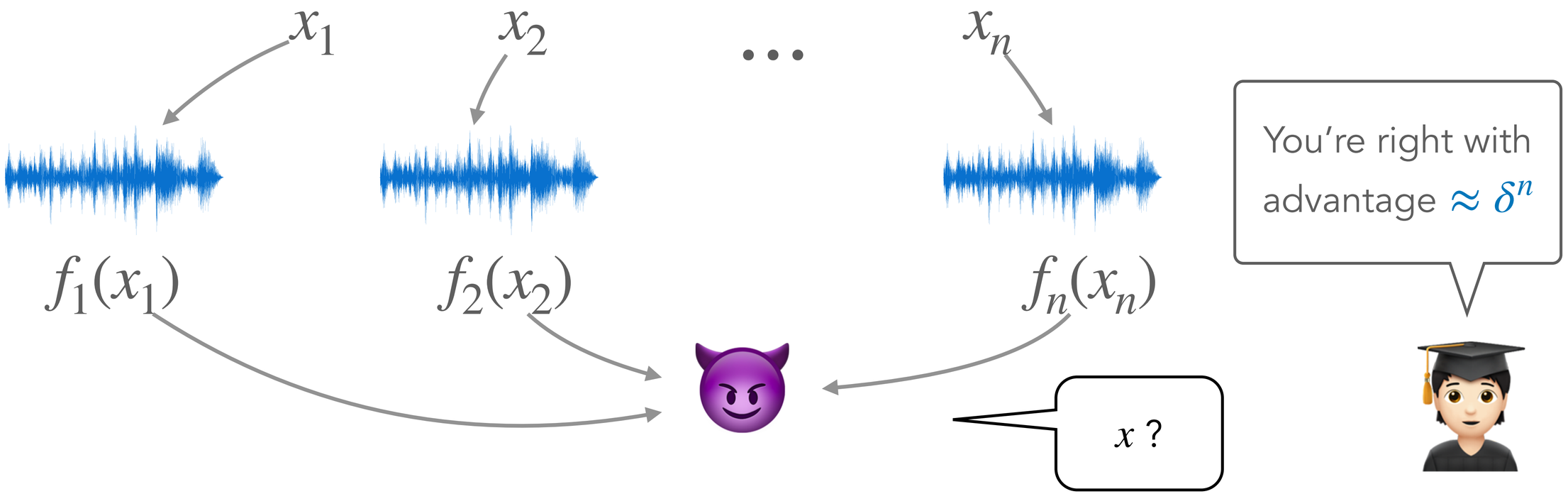
A function is **δ-noisy** if (for $X \sim \mathcal{U}$):

$$\mathbb{E}_y\left[\Delta\left(X;\ \left(X \mid f(X) = y\right)\right)\right] \leq \delta$$

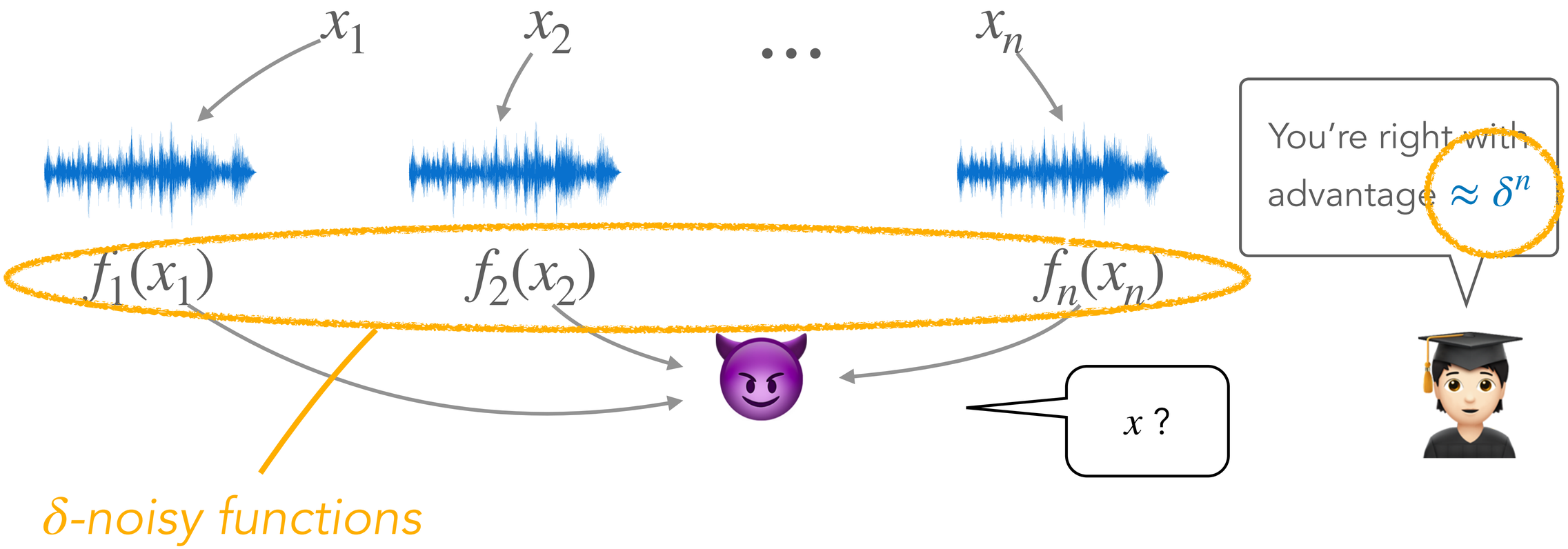
expectation on the possible leakage values

statistical distance between X and X and given $f(X) = y$

Soundness of masking with noise (generalised)

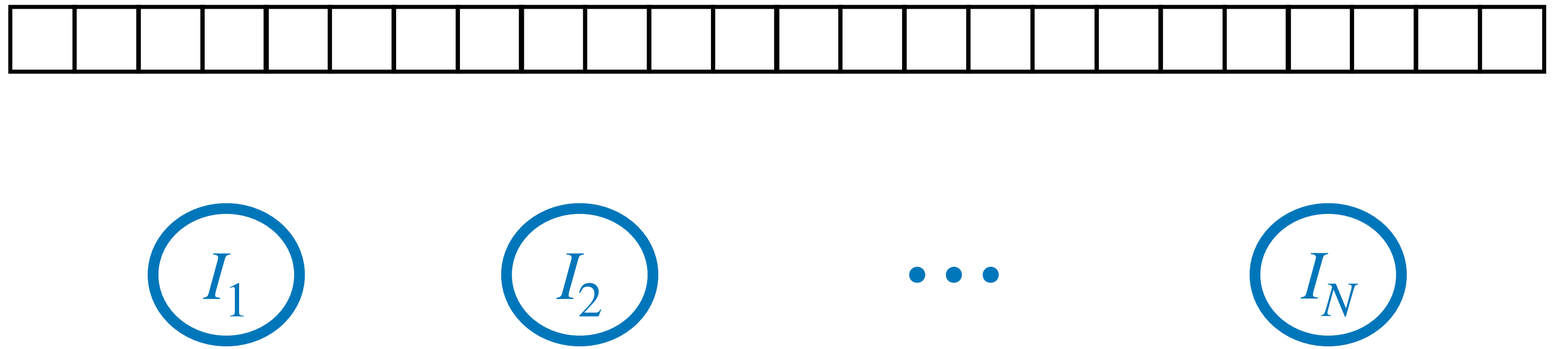


Soundness of masking with noise (generalised)

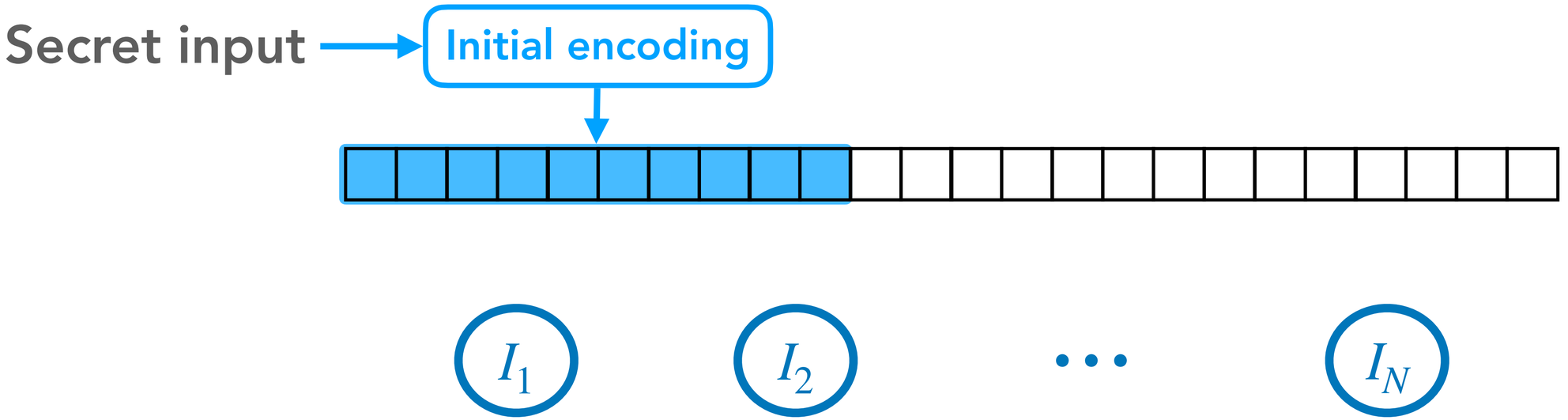


Simulation security

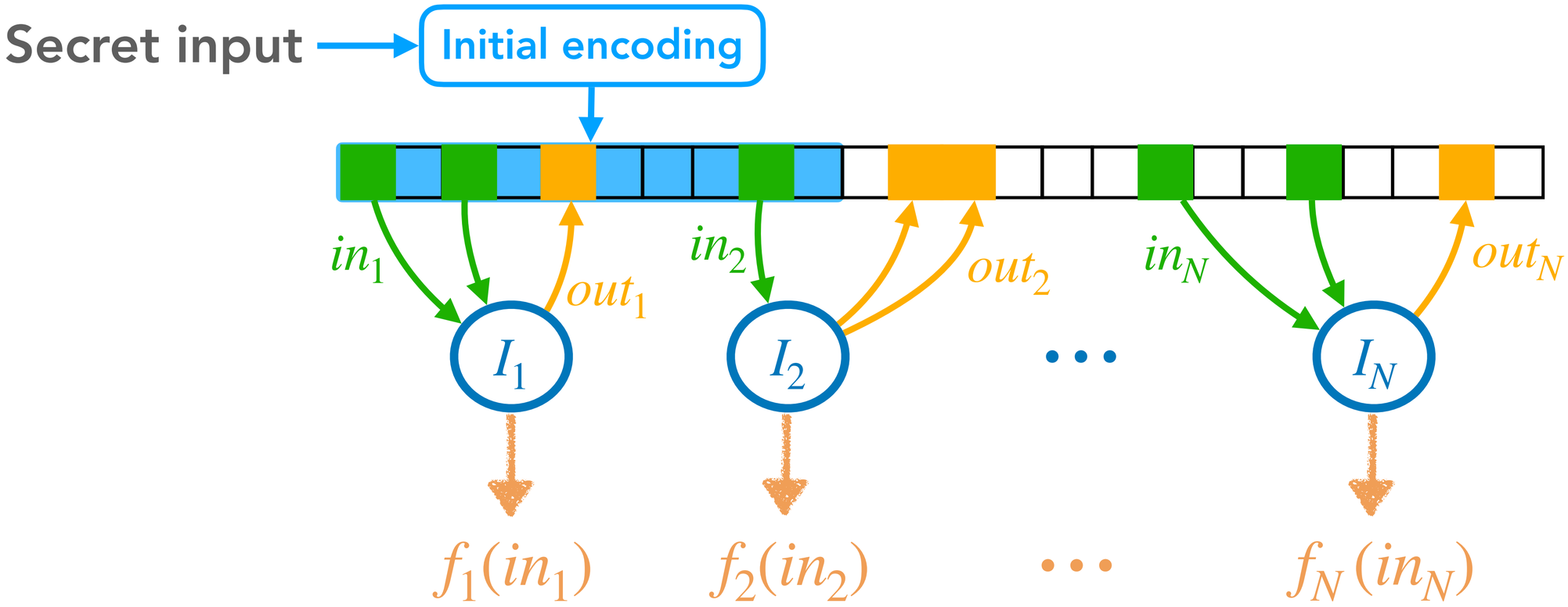
Secret input



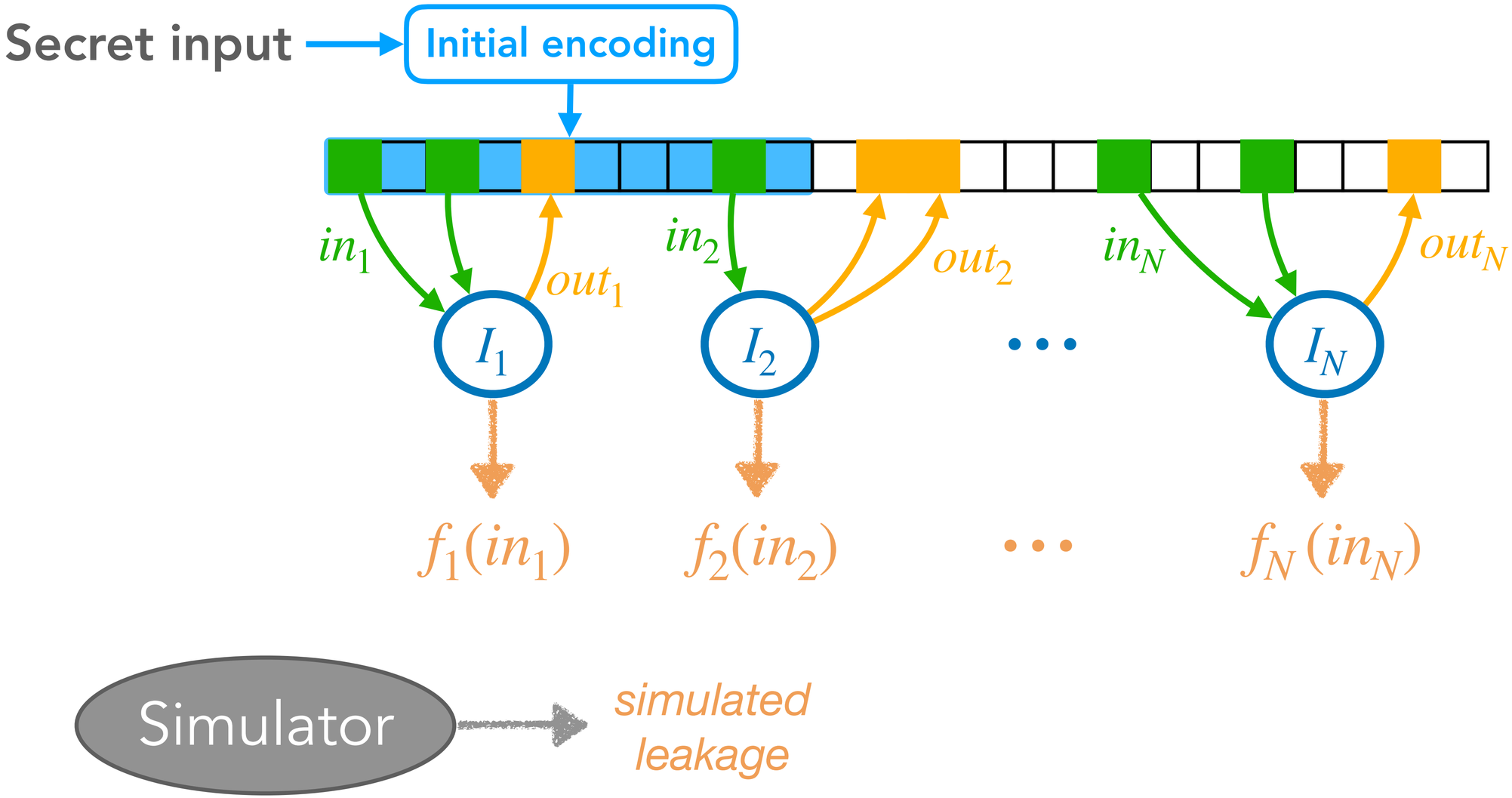
Simulation security



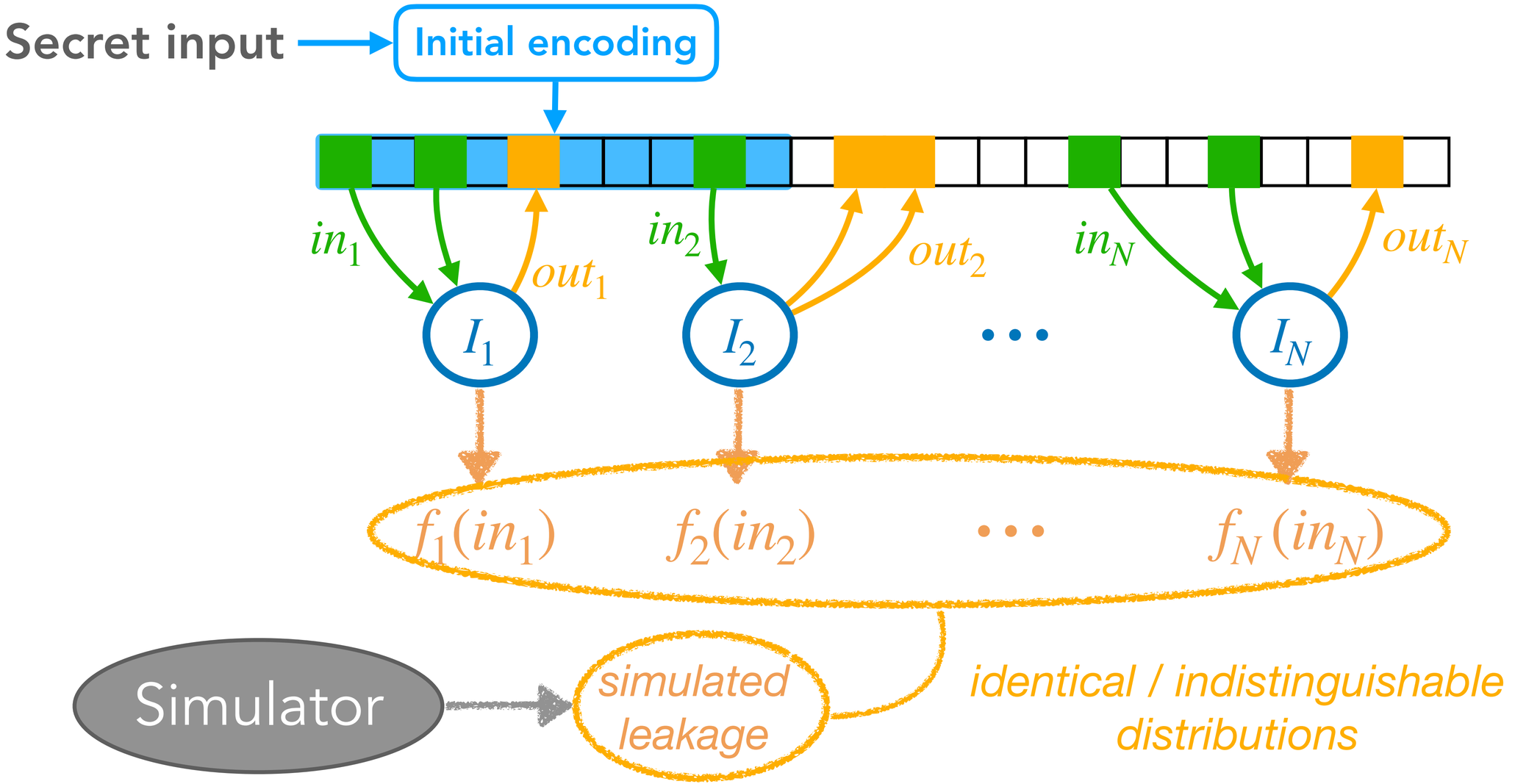
Simulation security



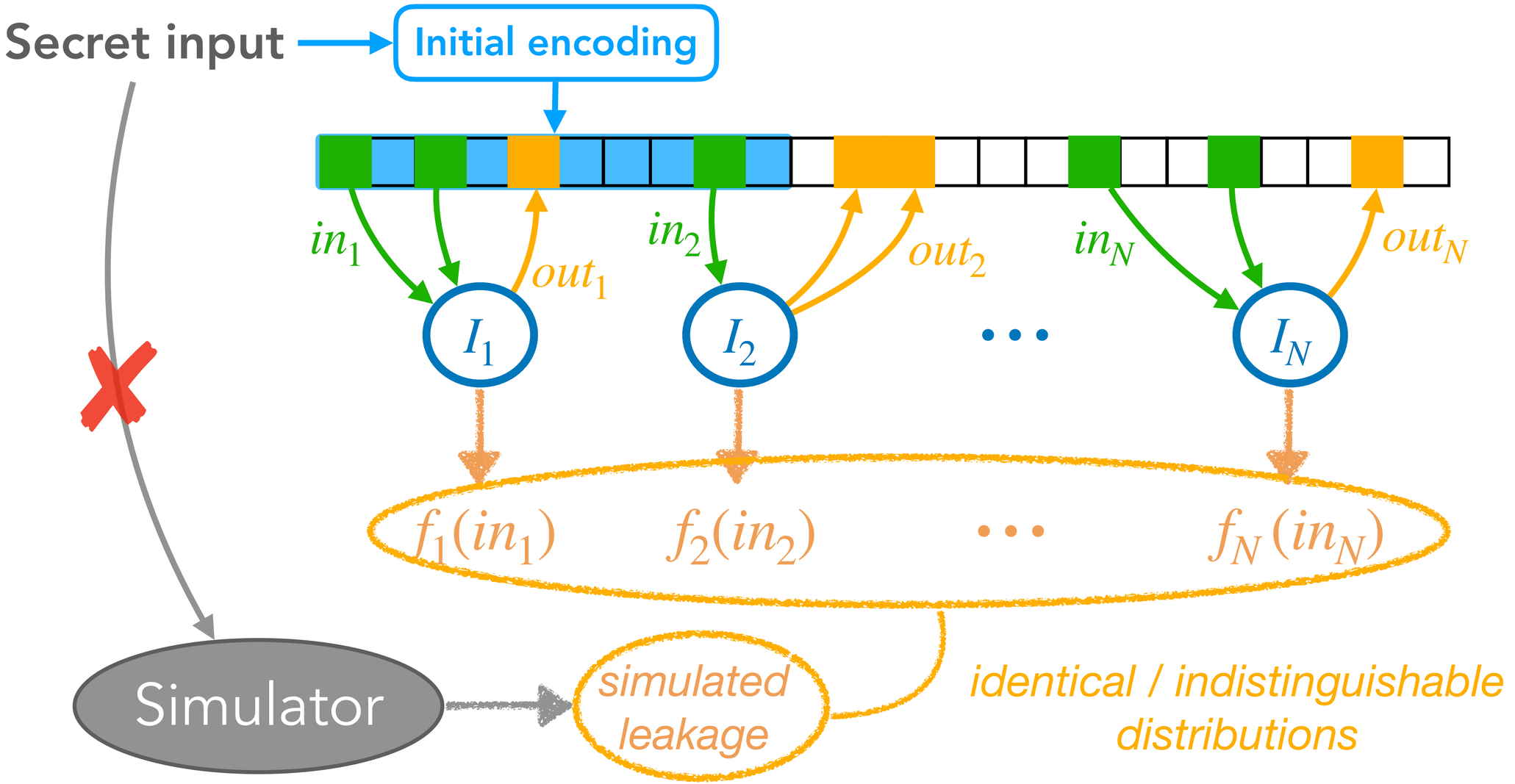
Simulation security



Simulation security

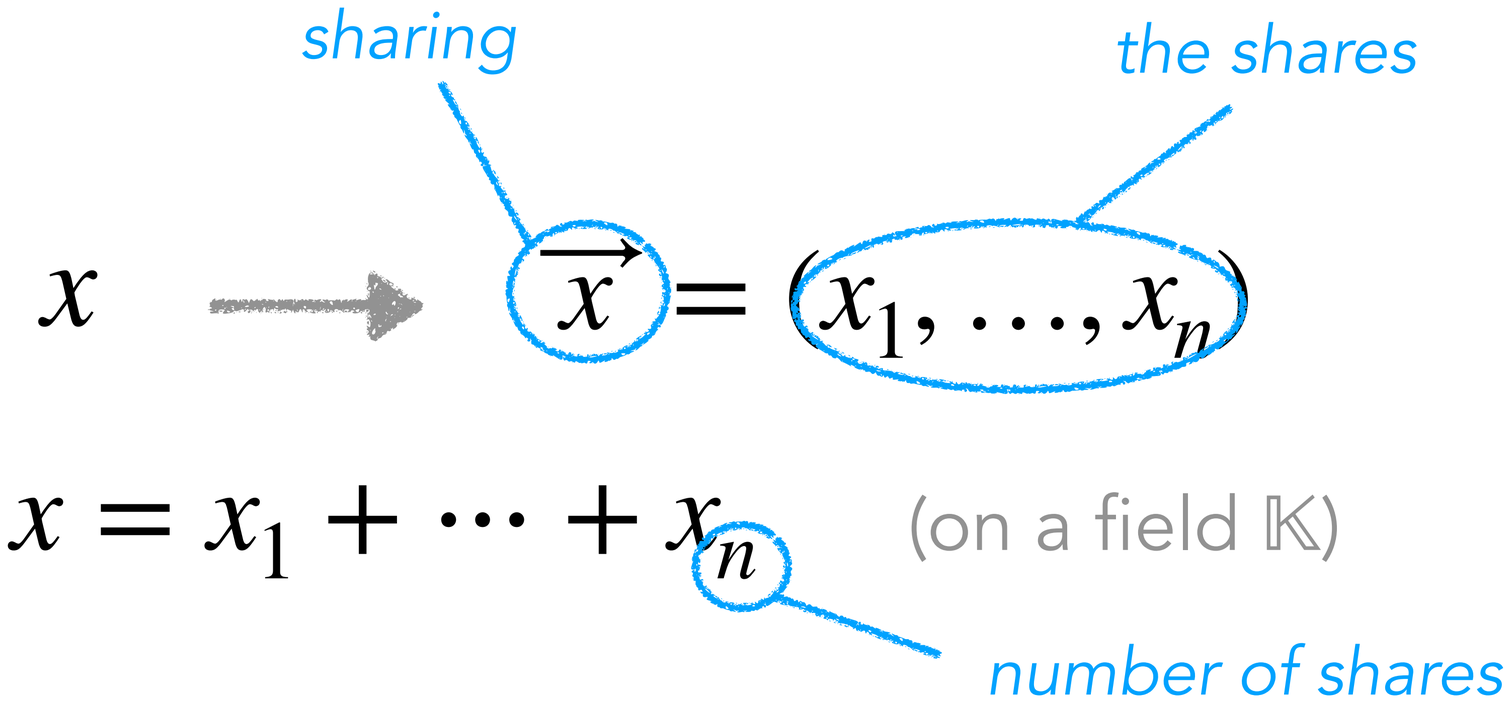


Simulation security

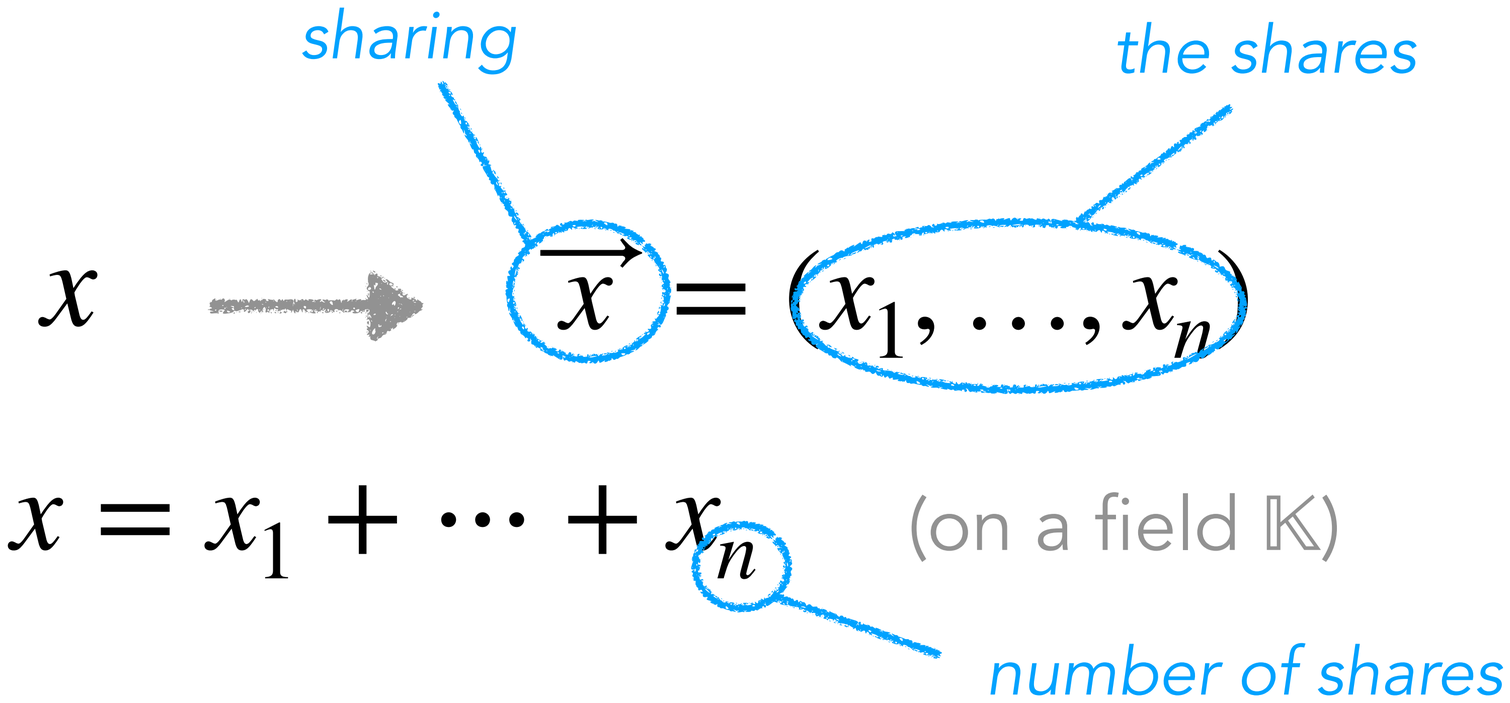


Masked computation

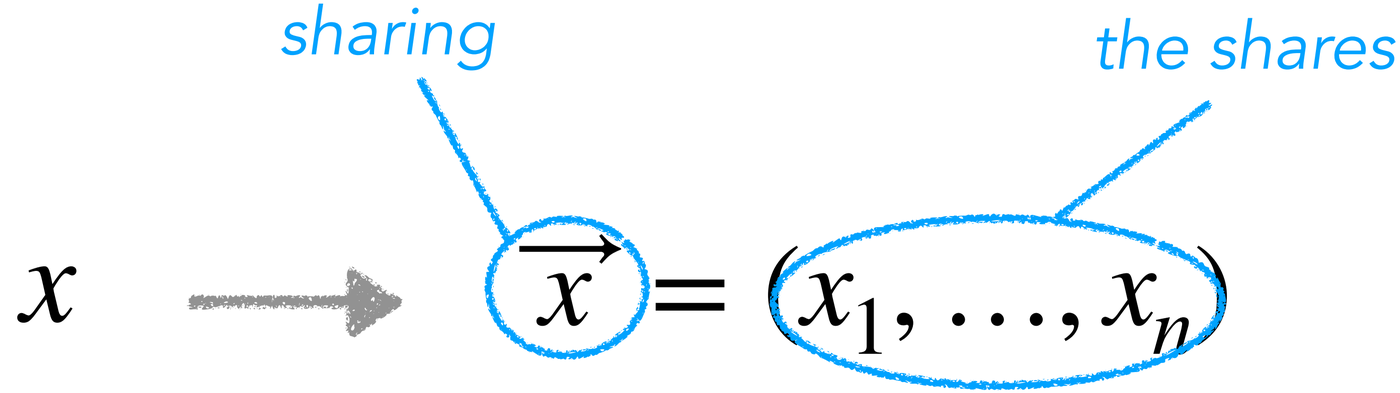
Masking


sharing
the shares
$x \longrightarrow \vec{x} = (x_1, \ldots, x_n)$
$x = x_1 + \cdots + x_n$
(on a field $\mathbb{K}$)
number of shares

Masking


sharing
the shares
$x \longrightarrow \vec{x} = (x_1, \ldots, x_n)$
$x = x_1 + \cdots + x_n$
(on a field $\mathbb{K}$)
number of shares

Masking



$$x = x_1 + \cdots + x_n$$ (on a field $\mathbb{K}$)

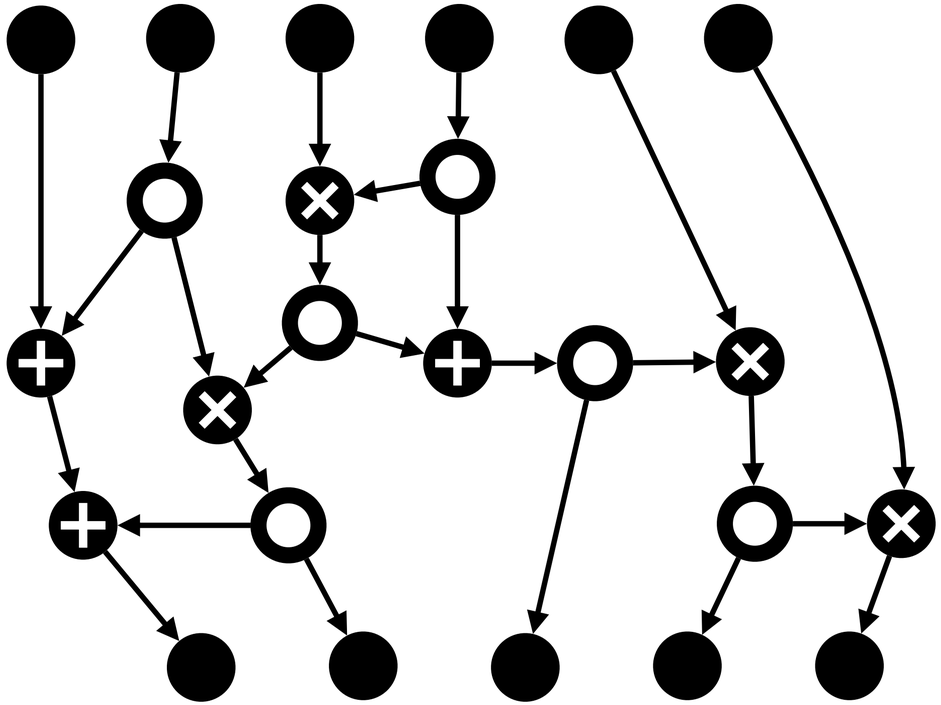
number of shares

⚠️ all the shares are necessary to recover x

🎲 any $n - 1$ shares are completely random

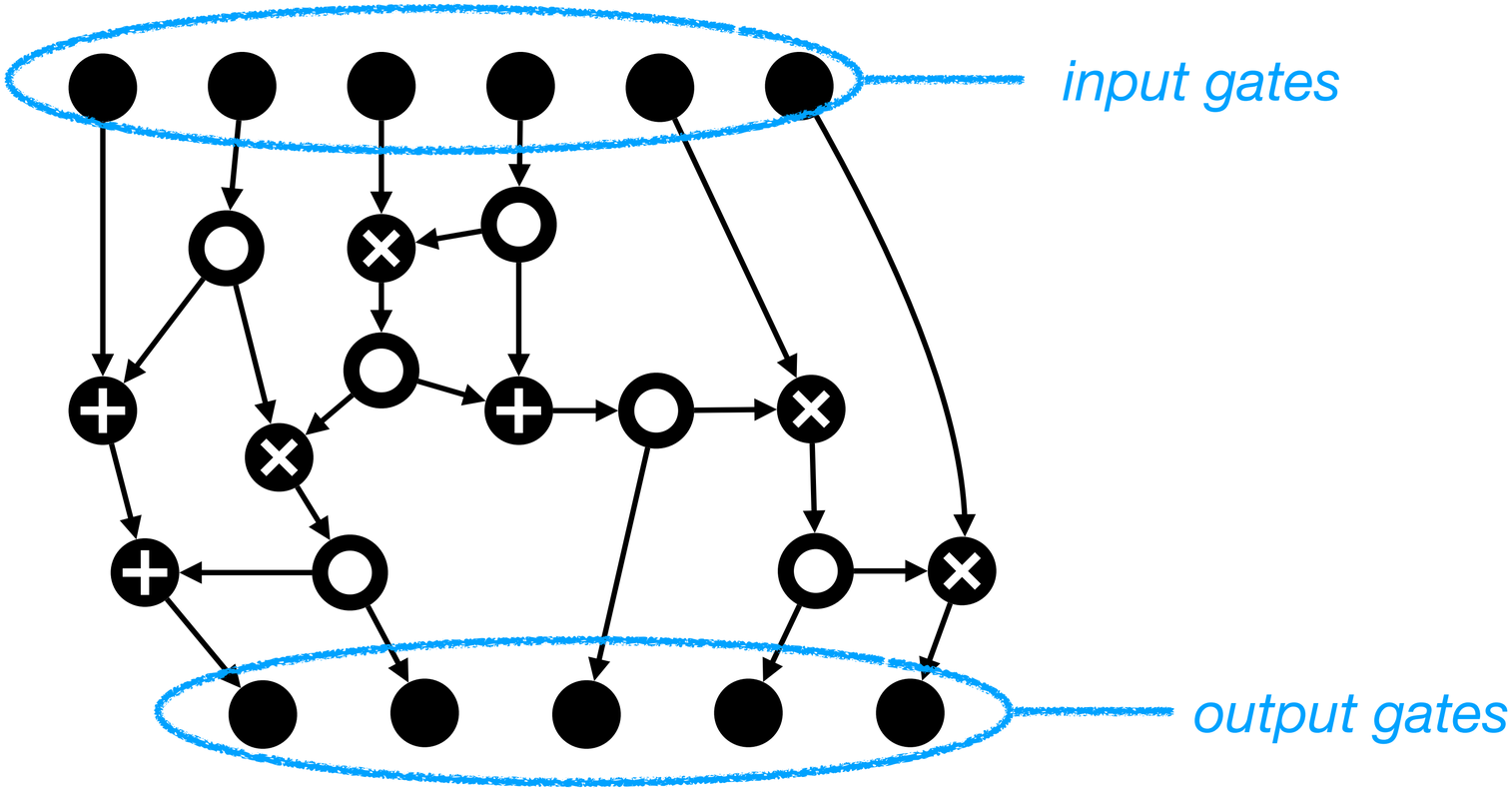
Circuit model

Crypto computation modelled as an arithmetic circuit on $\mathbb{K}$



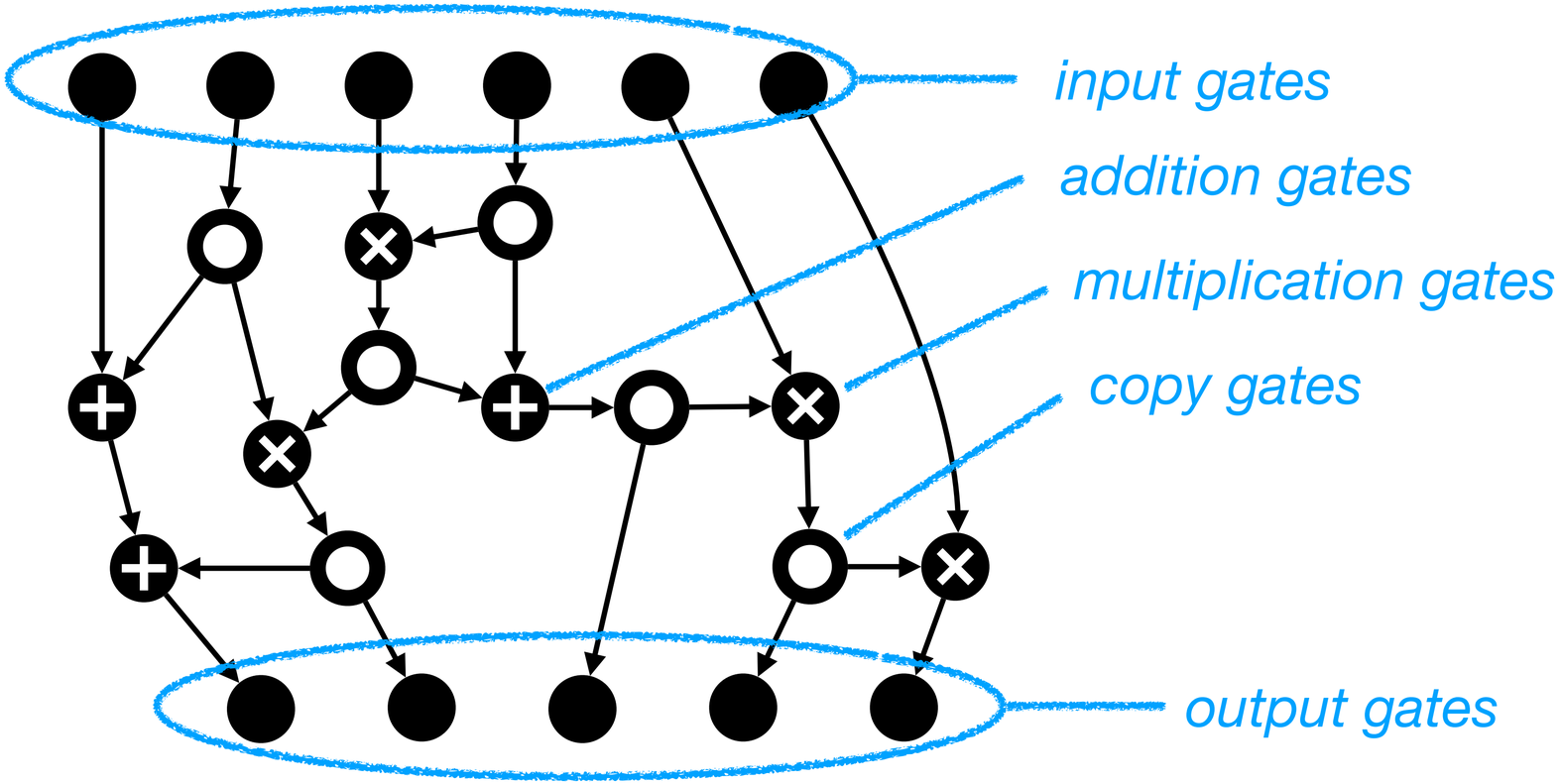
Circuit model

Crypto computation modelled as an arithmetic circuit on $\mathbb{K}$



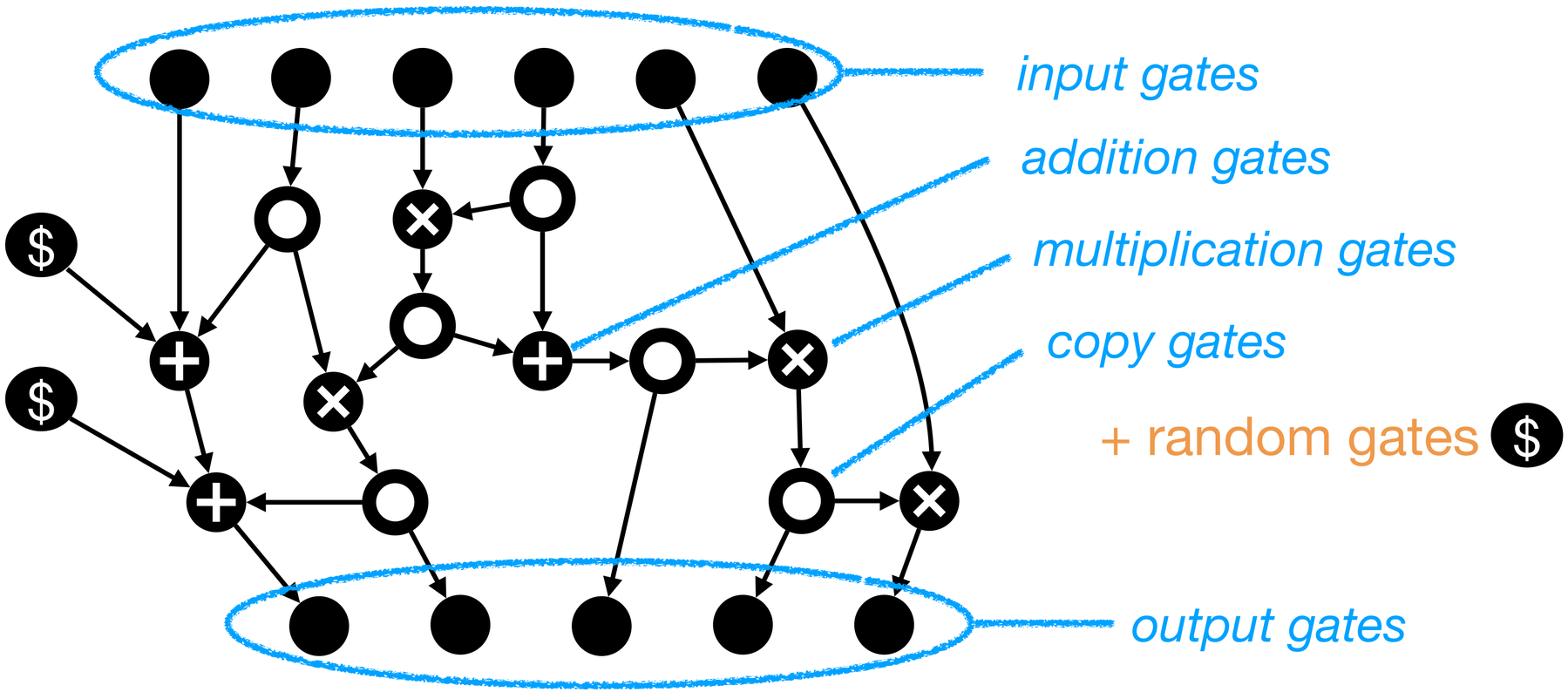
Circuit model

Crypto computation modelled as an arithmetic circuit on $\mathbb{K}$



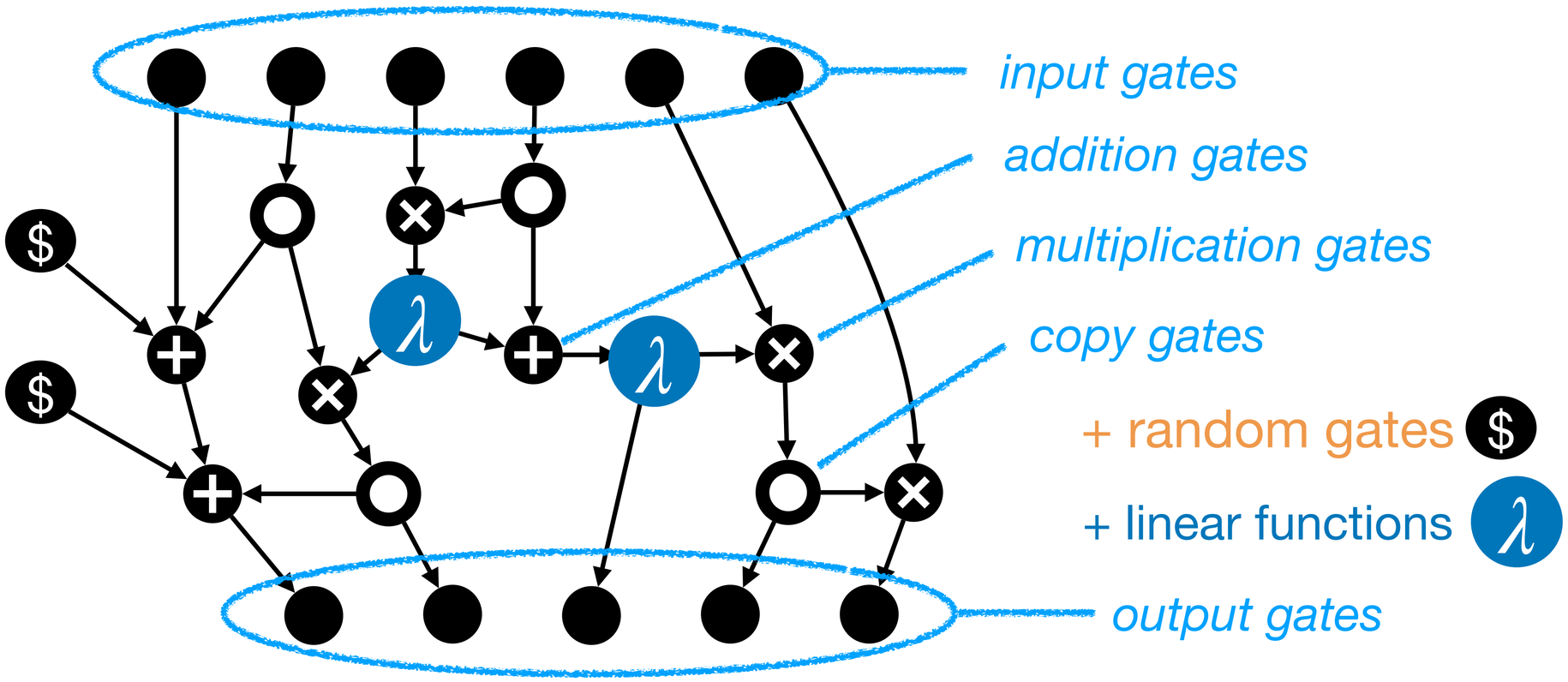
Circuit model

Crypto computation modelled as an arithmetic circuit on $\mathbb{K}$

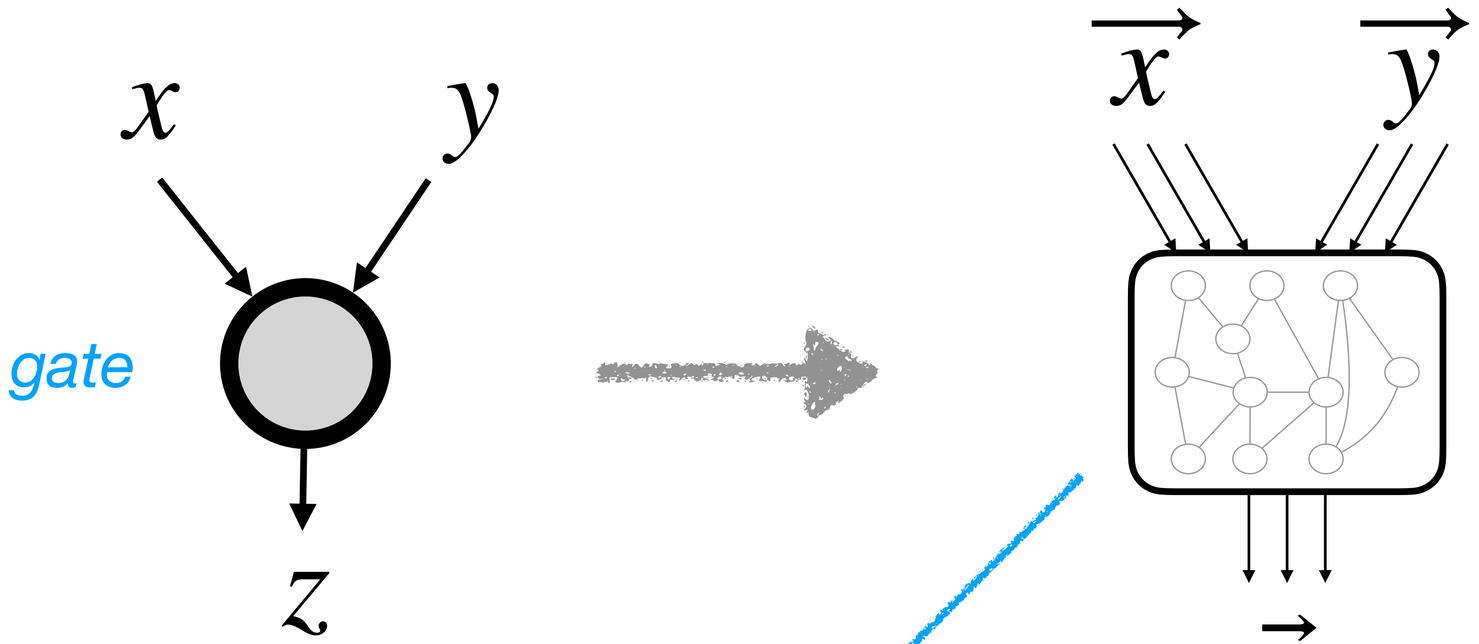


Circuit model

Crypto computation modelled as an arithmetic circuit on $\mathbb{K}$

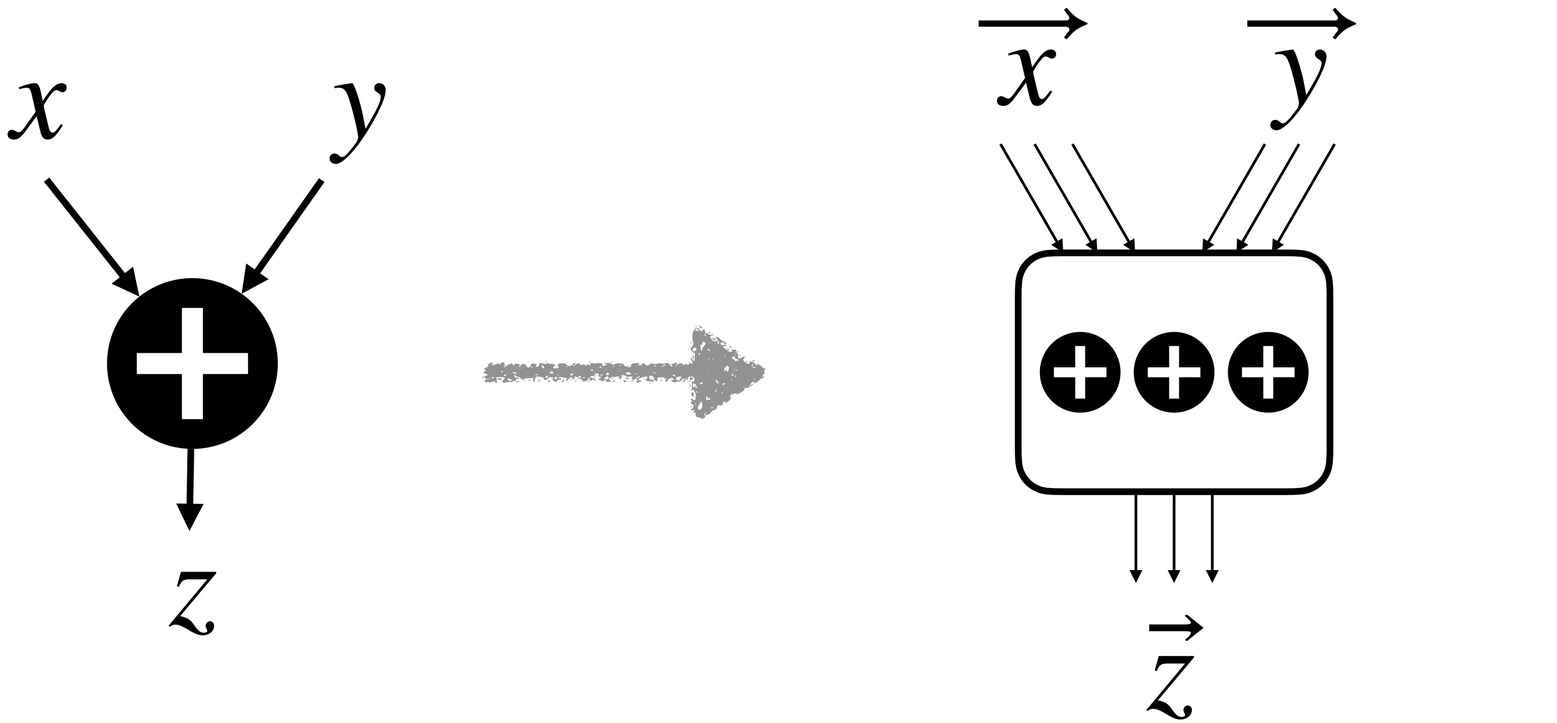


Gadgets



gadget : small circuit computing an operation on sharings

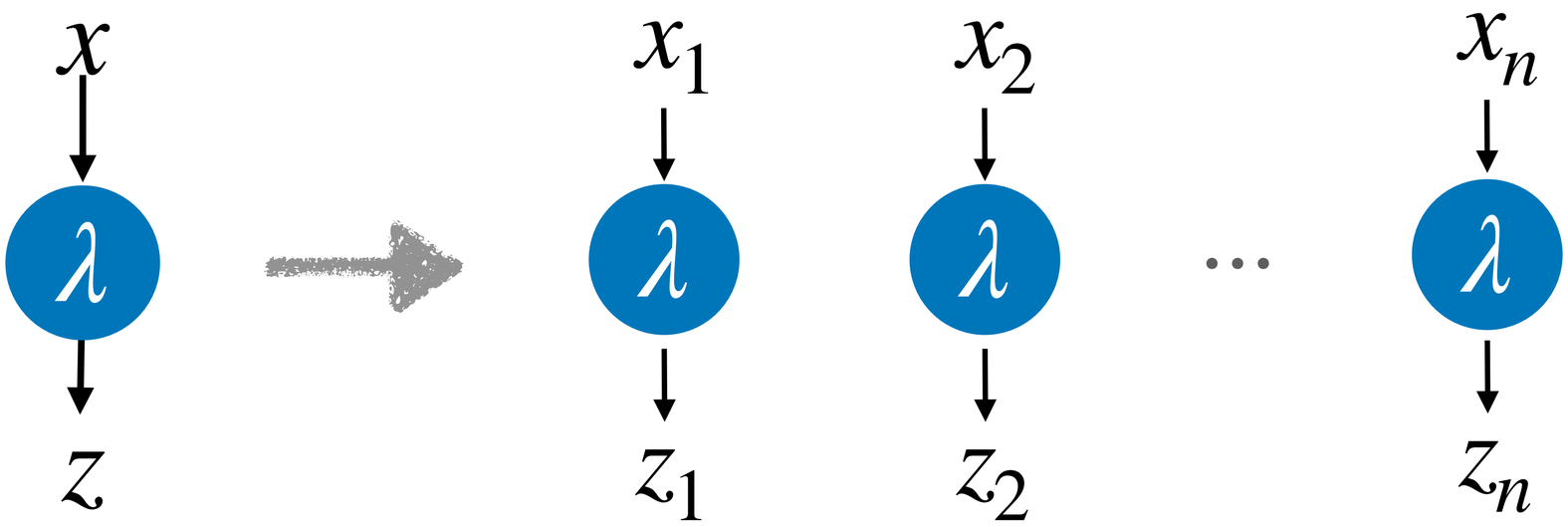
Adition gadget



sharewise computation

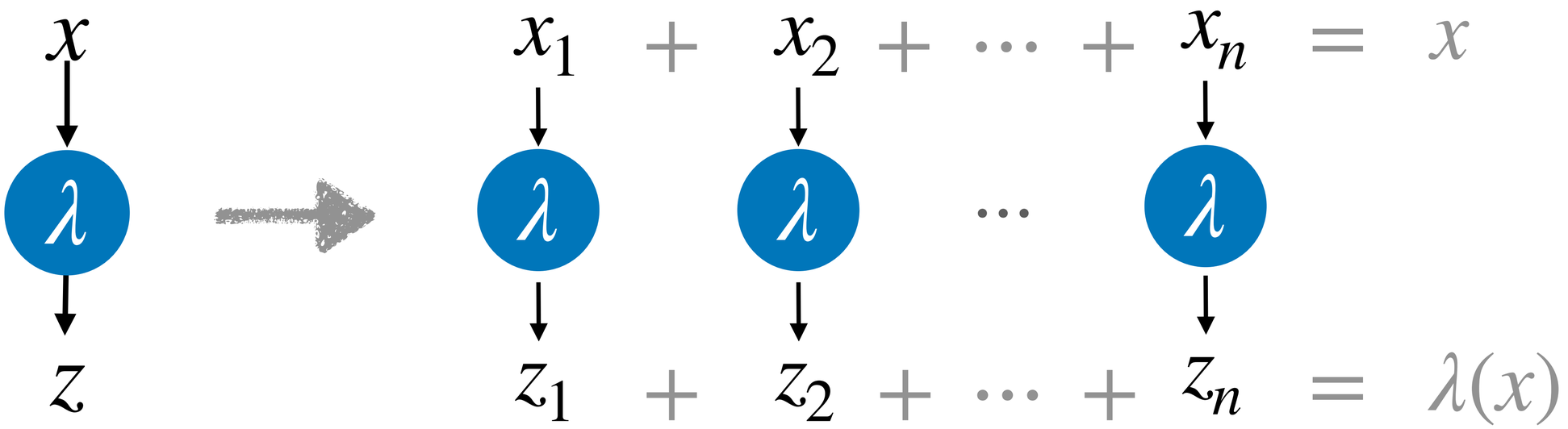
$\Rightarrow$ n addition gates

Linear gadget



sharewise computation ⇒ n evaluations of λ

Linear gadget



sharewise computation ⇒ n evaluations of λ

Multiplication gadget

$$z = x \cdot y = \left(\sum\nolimits_i x_i\right)\left(\sum\nolimits_i y_i\right) = \sum\nolimits_{i,j} x_i y_j$$

Multiplication gadget

$$z = x \cdot y = \left(\sum_i x_i\right)\left(\sum_i y_i\right) = \sum_{i,j} x_i y_j$$

split into n shares

$$z_1$$
$$z_2$$
$$\vdots$$
$$z_n$$

Multiplication gadget

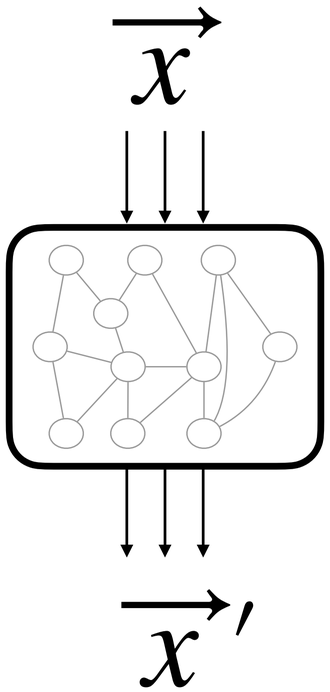
$$z = x \cdot y = \left(\sum_i x_i\right)\left(\sum_i y_i\right) = \sum_{i,j} x_i y_j$$

split into n shares

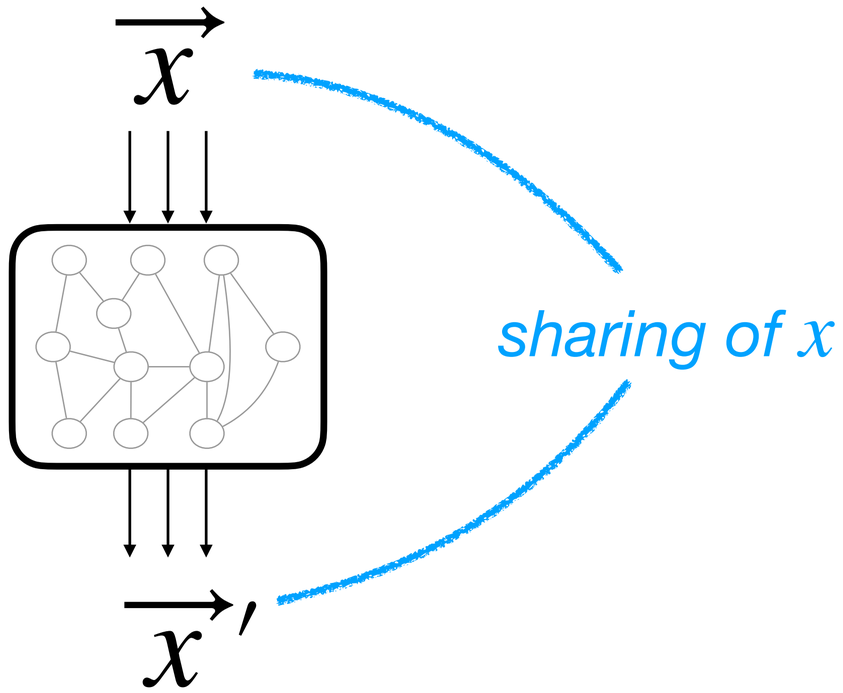
$$z_1$$
$$z_2$$
$$\vdots$$
$$z_n$$

+ fresh randomness

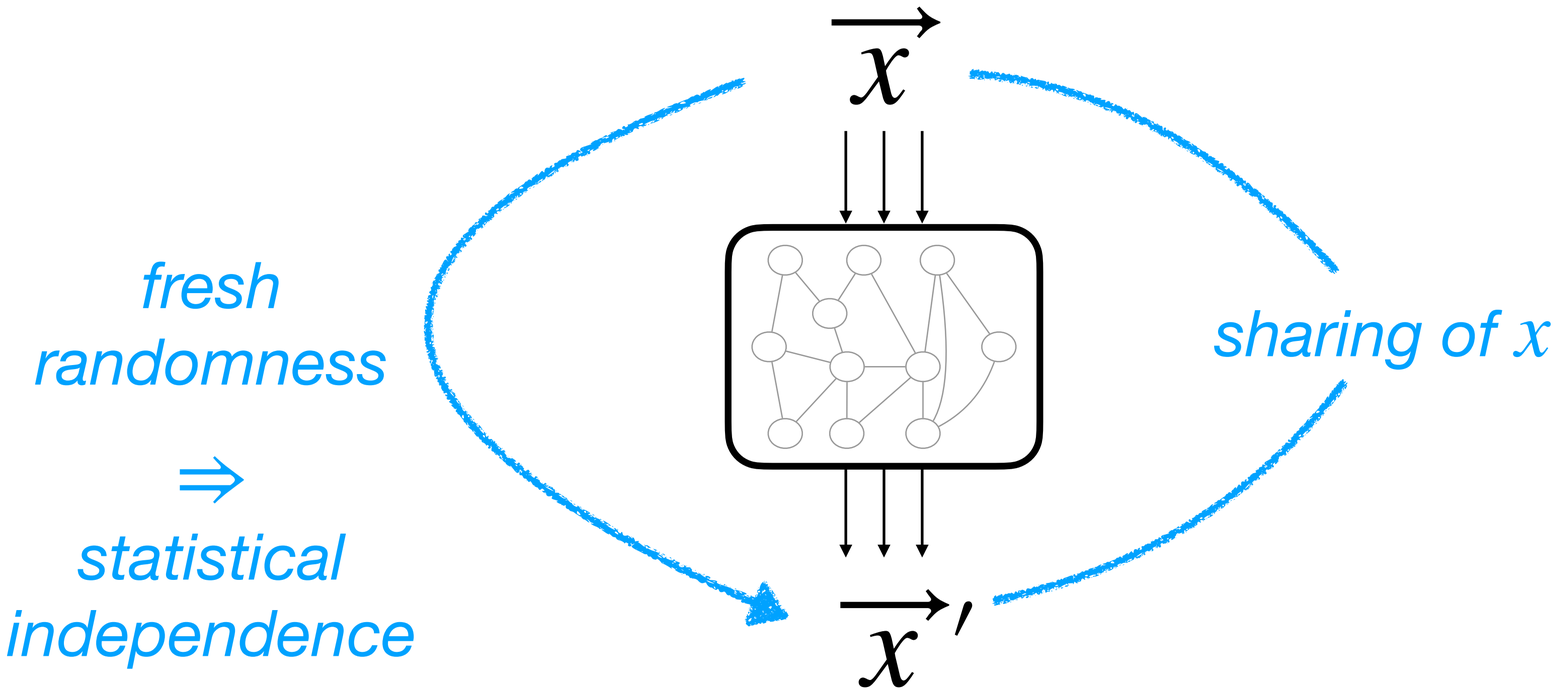
Refresh gadget



Refresh gadget



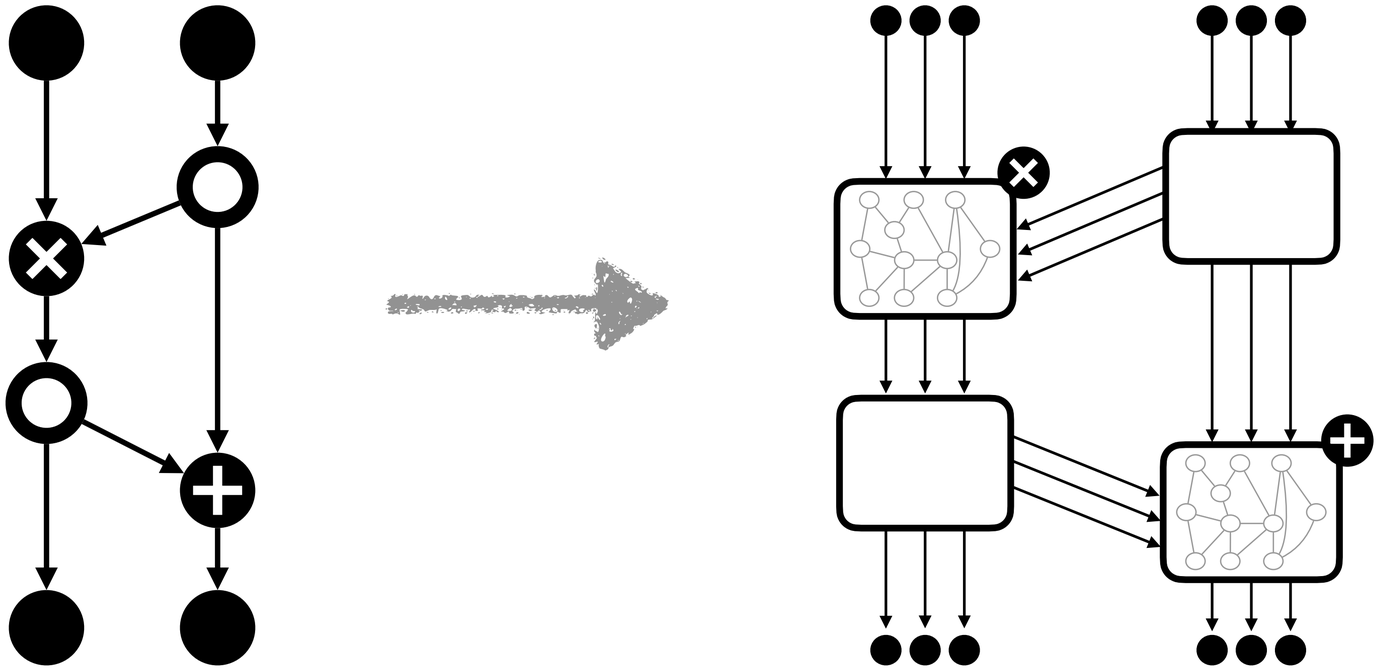
Refresh gadget



Standard circuit compiler

wire $\rightarrow$ n wires (sharing)

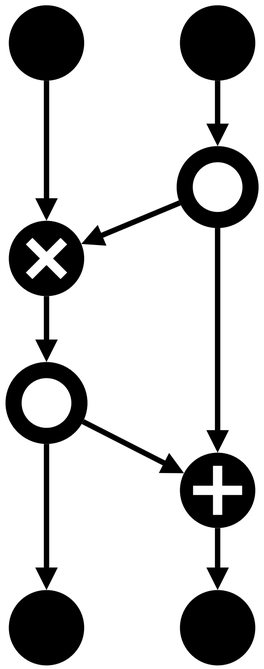
gate $\rightarrow$ gadget



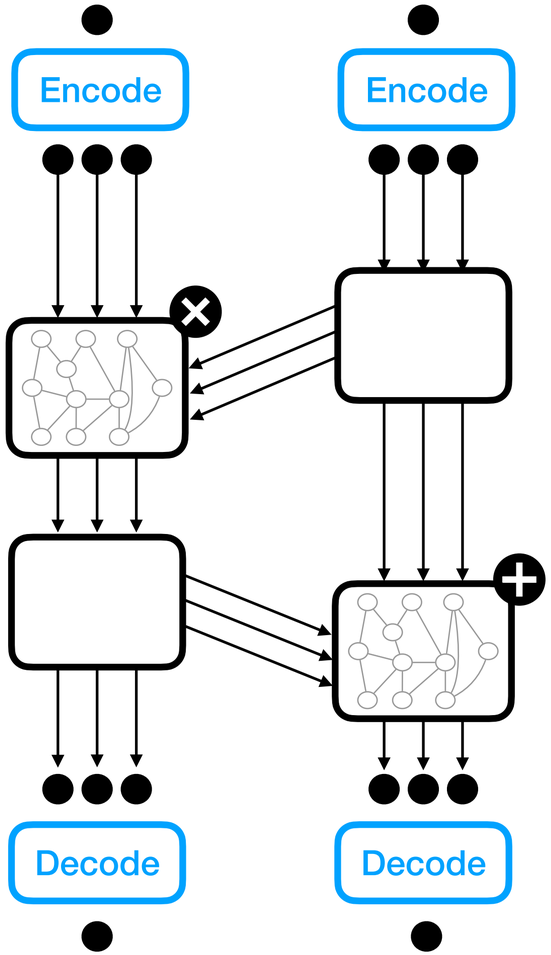
Standard circuit compiler

wire $\rightarrow$ n wires (sharing)

gate $\rightarrow$ gadget



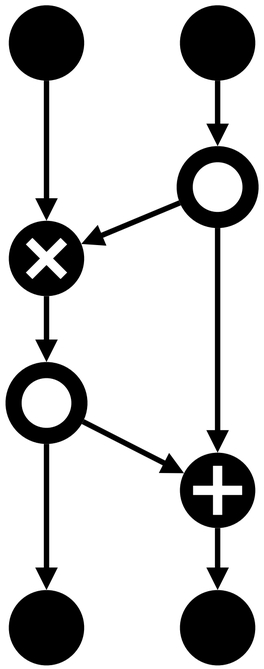
functional equivalence



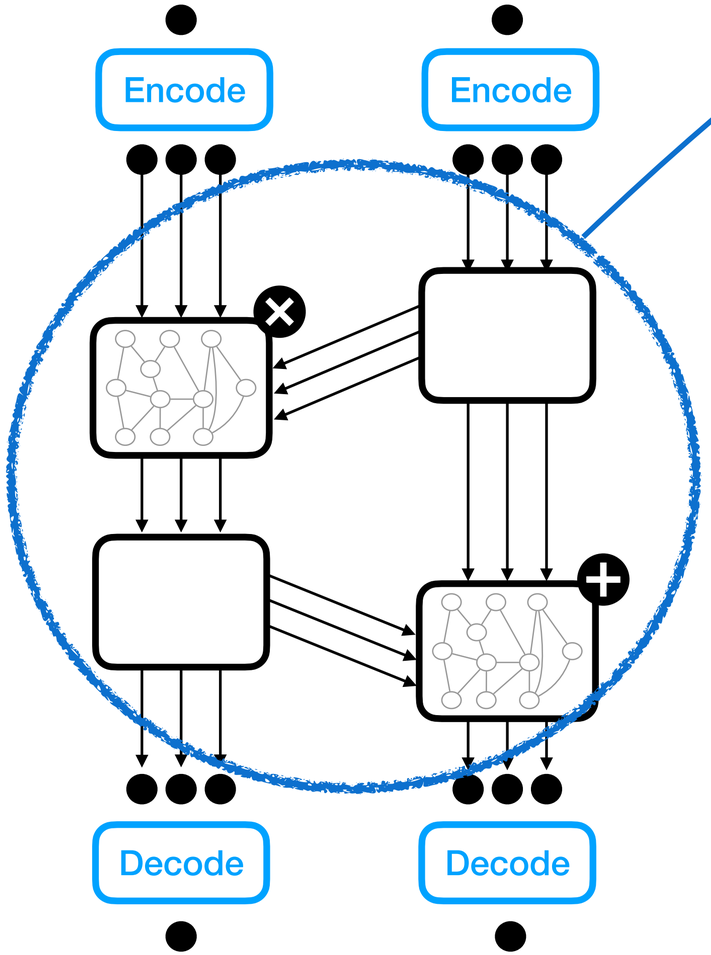
Standard circuit compiler

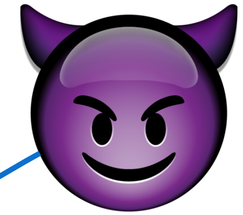
wire $\rightarrow$ n wires (sharing)

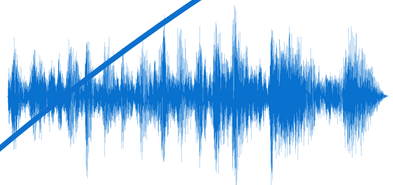
gate $\rightarrow$ gadget



functional equivalence





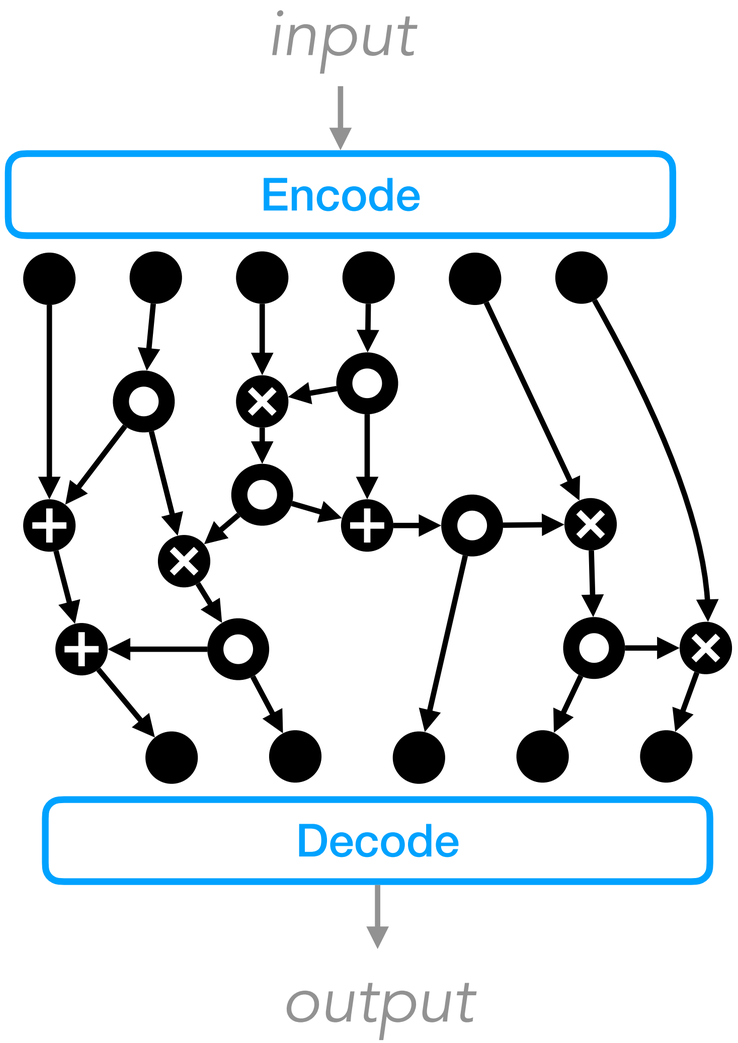


noisy leakage

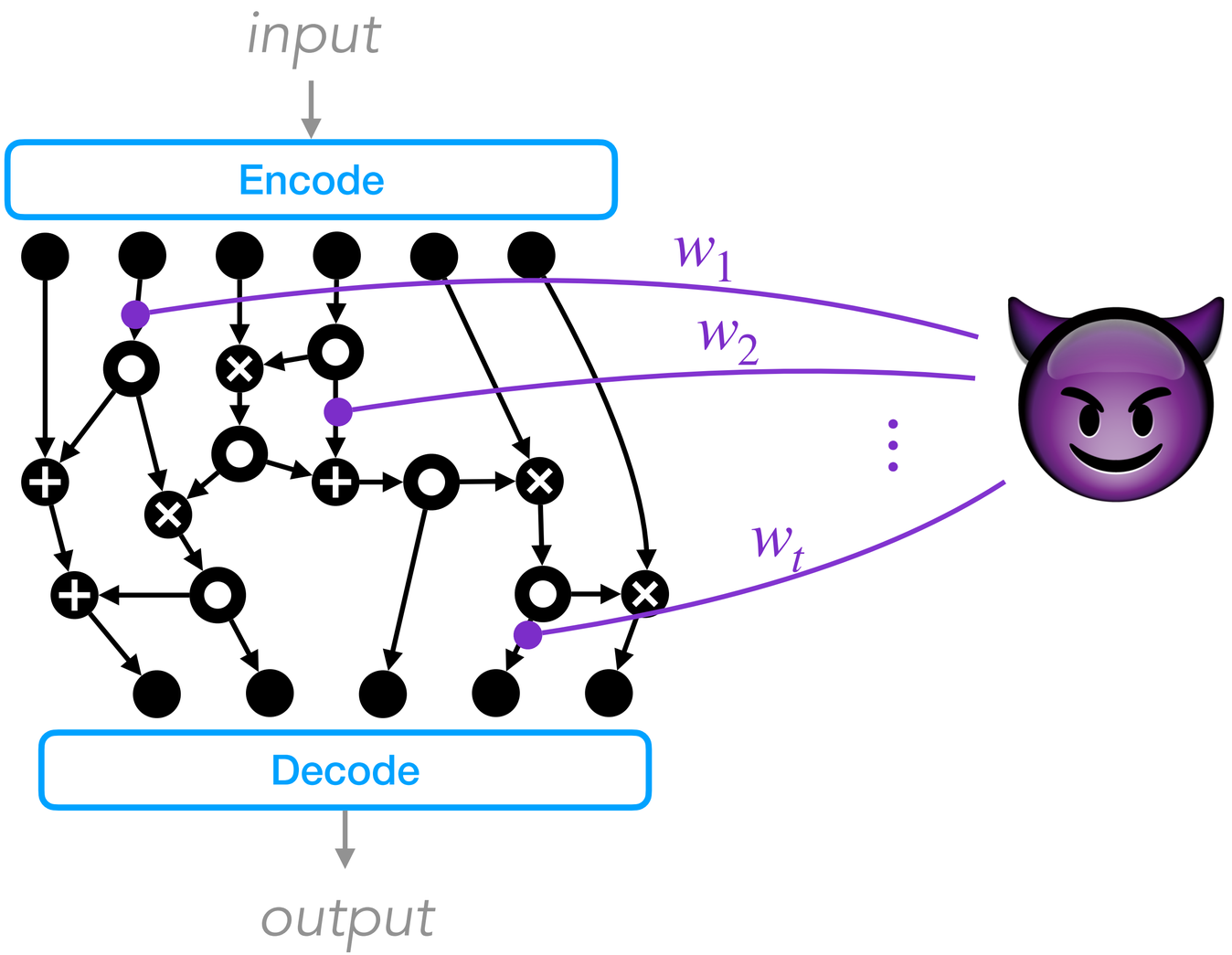
How to prove the security vs. δ-noisy leakage?

Probing models

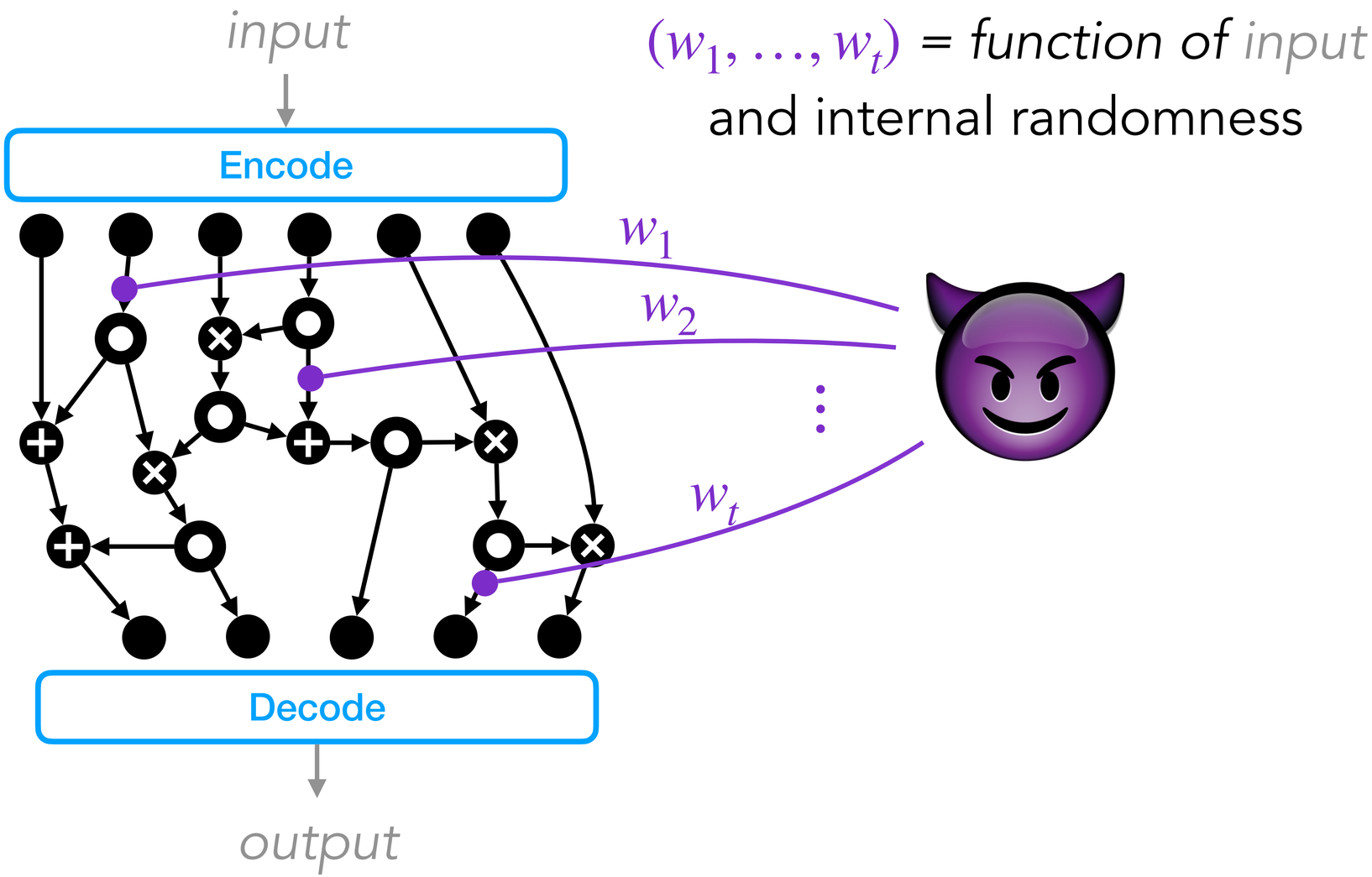
Probing model



Probing model

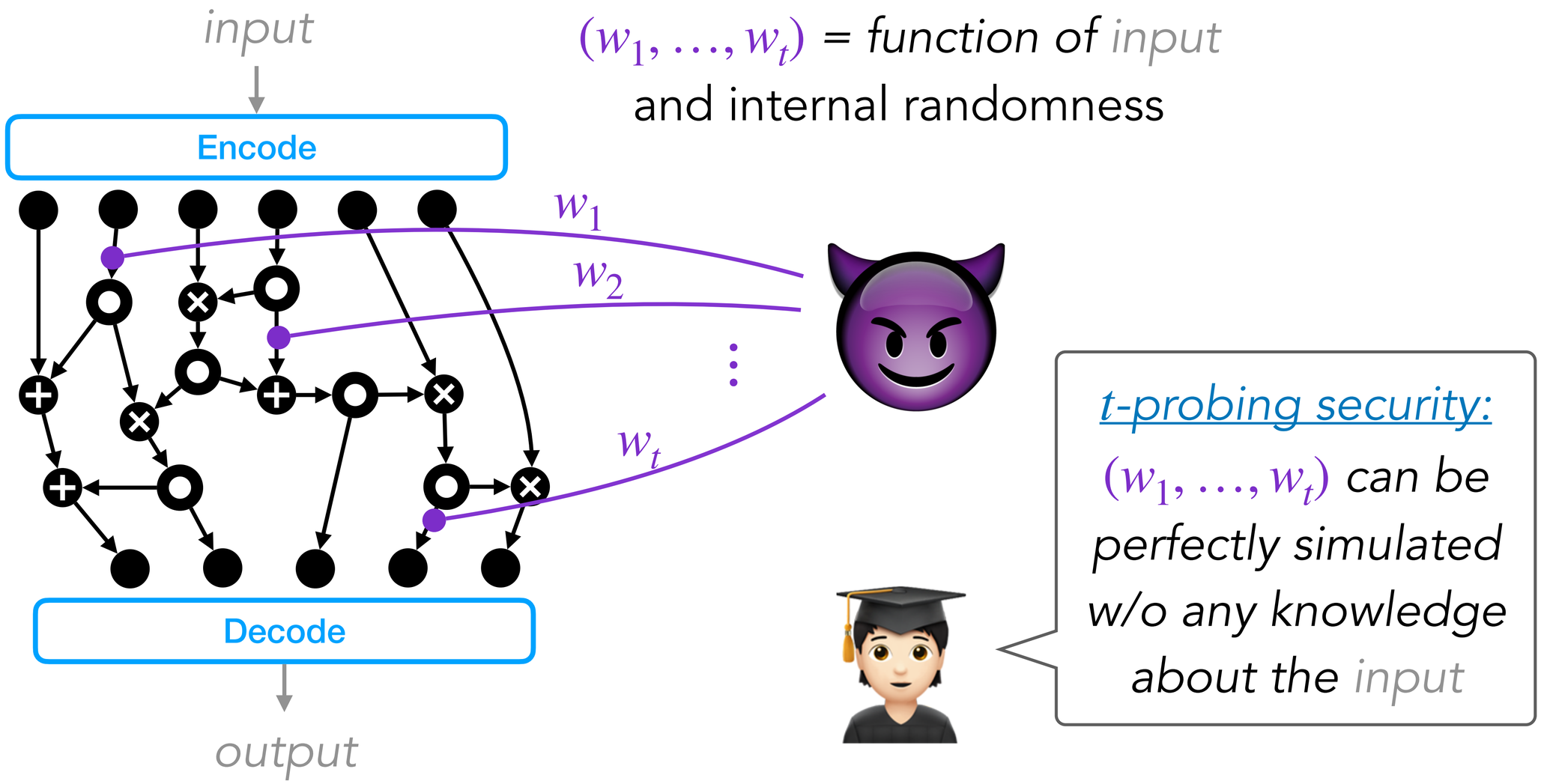


Probing model



$(w_1, \ldots, w_t)$ = *function of input*
and internal randomness

Probing model



$(w_1, \ldots, w_t)$ = *function of input* and internal randomness

t-probing security:

$(w_1, \ldots, w_t)$ *can be perfectly simulated w/o any knowledge about the input*

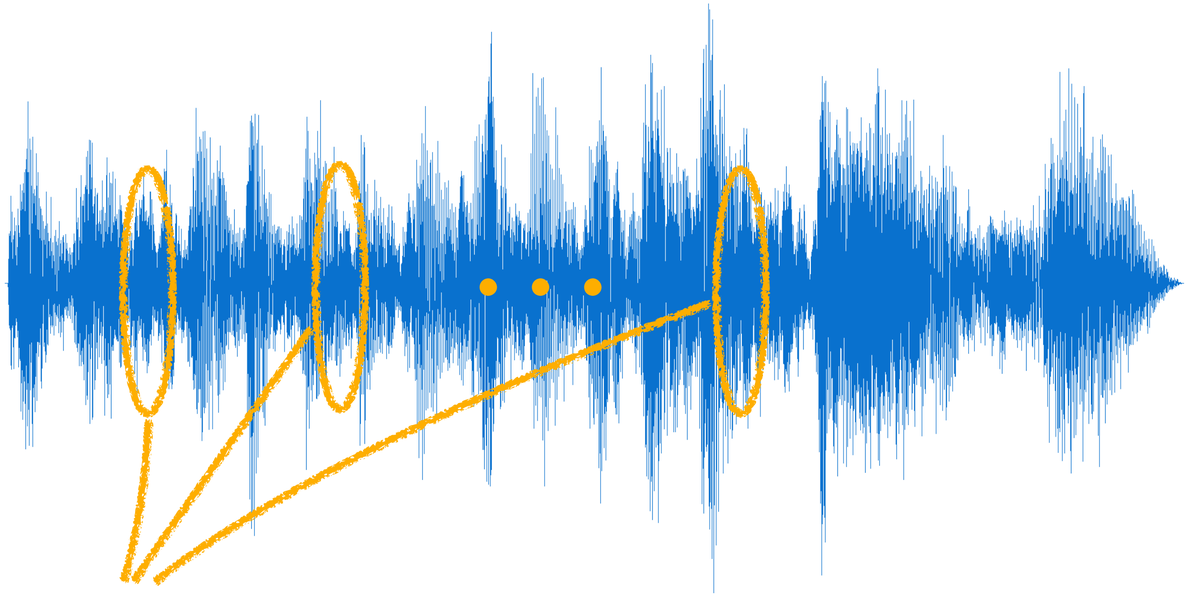
Probing model

t-probing security

⇔

security against

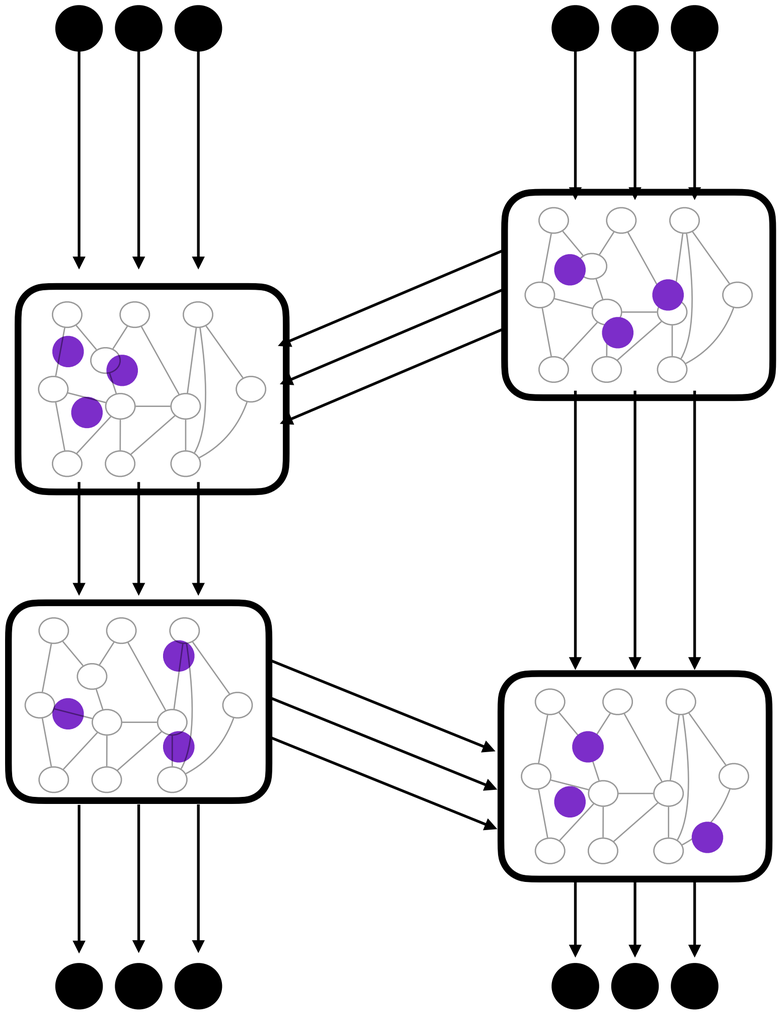
t-order DPA



any t leakage points

independent of the secrets

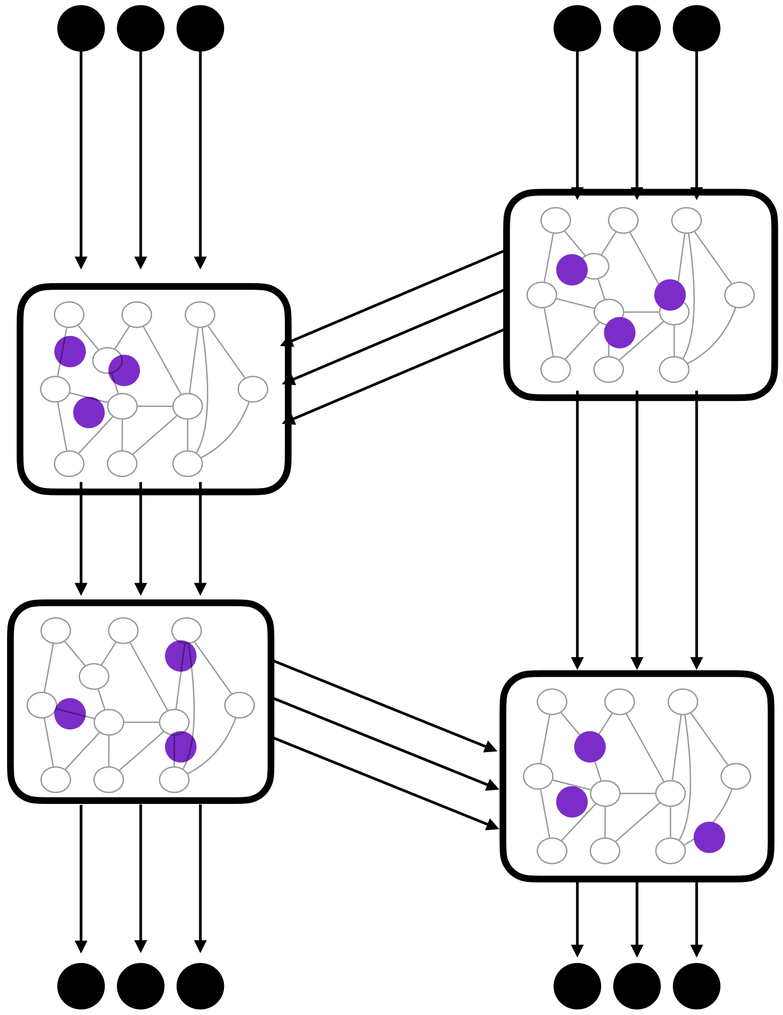
Region probing model



t probes per gadget (or region)

with $t = r \times |G|$

Region probing model



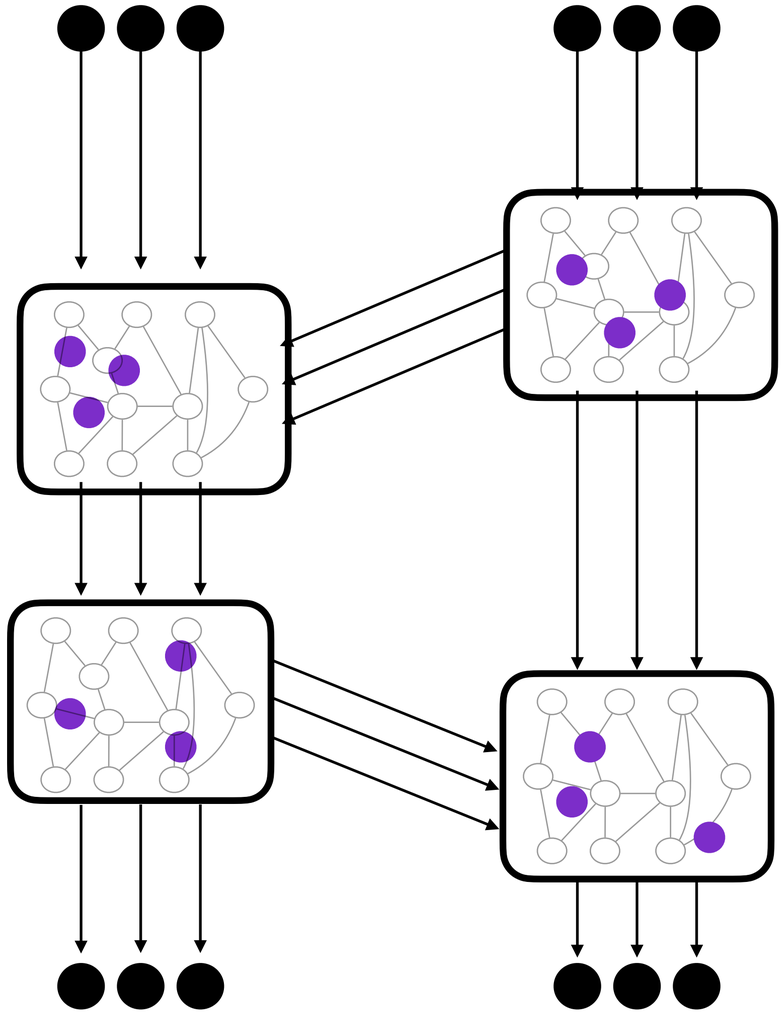
t probes per gadget (or region)

with $t = r \times |G|$

probing rate

number of wires in G

Region probing model



t probes per gadget (or region)

$$\text{with } t = r \times |G|$$

probing rate

number of wires in G

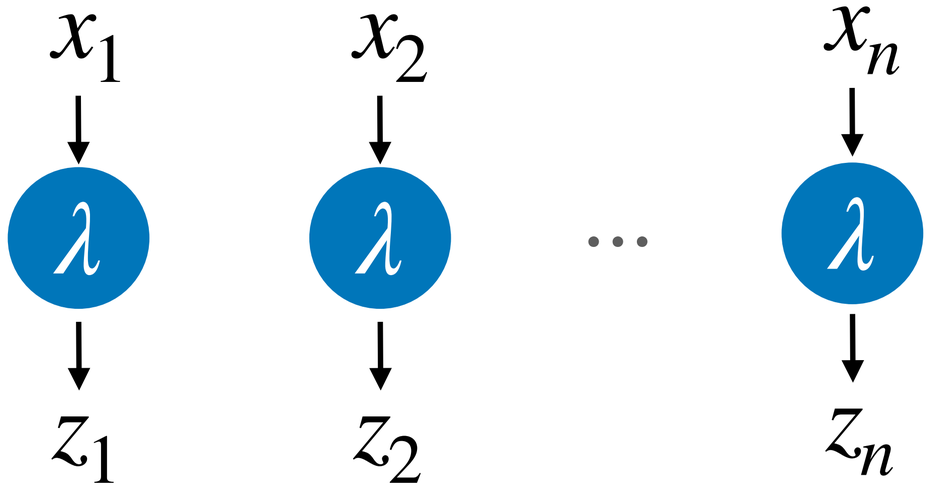
$\Rightarrow$ *r-region probing security*

Security of sharewise gadgets

sharewise gadget
⇒ inherent probing security

$n-1$ probes

= completely random 🎲



ISW multiplication gadget

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}}$$

ISW multiplication gadget

cross-products $\sum_{i,j} x_i y_j$

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}}$$

ISW multiplication gadget

cross-products $\sum_{i,j} x_i y_j$

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} & r_{1,2} & r_{1,3} \\ -r_{1,2} & & r_{2,3} \\ -r_{1,3} & -r_{2,3} & \end{pmatrix}$$

ISW multiplication gadget

cross-products $\sum_{i,j} x_i y_j$

fresh randomness (cancelling out)

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} & r_{1,2} & r_{1,3} \\ -r_{1,2} & & r_{2,3} \\ -r_{1,3} & -r_{2,3} & \end{pmatrix}$$

ISW multiplication gadget

cross-products $\sum_{i,j} x_i y_j$

fresh randomness (cancelling out)

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} & r_{1,2} & r_{1,3} \\ -r_{1,2} & & r_{2,3} \\ -r_{1,3} & -r_{2,3} & \end{pmatrix}$$

row sum →

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

ISW multiplication gadget

cross-products $\sum_{i,j} x_i y_j$

fresh randomness (cancelling out)

$$\begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ & x_2y_2 & x_2y_3 \\ & & x_3y_3 \end{pmatrix} + \begin{pmatrix} & & \\ x_2y_1 & & \\ x_3y_1 & x_3y_2 & \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} & r_{1,2} & r_{1,3} \\ -r_{1,2} & & r_{2,3} \\ -r_{1,3} & -r_{2,3} & \end{pmatrix}$$

row sum → $\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$

t probes ⇒ info on at most t shares

Composition security

⚠️ probing security for gadgets $\not\Rightarrow$ global (region) probing security

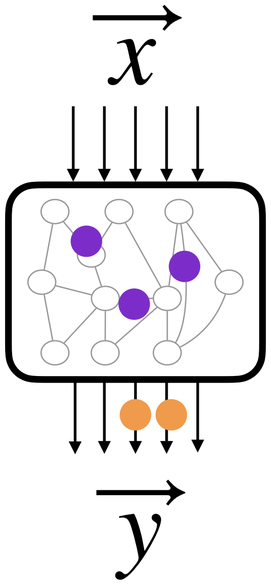
💡 composition security notions

Composition security

⚠️ probing security for gadgets $\not\Rightarrow$ global (region) probing security

💡 composition security notions

Example: strong non-interference (SNI)



t_1 *internal probes*

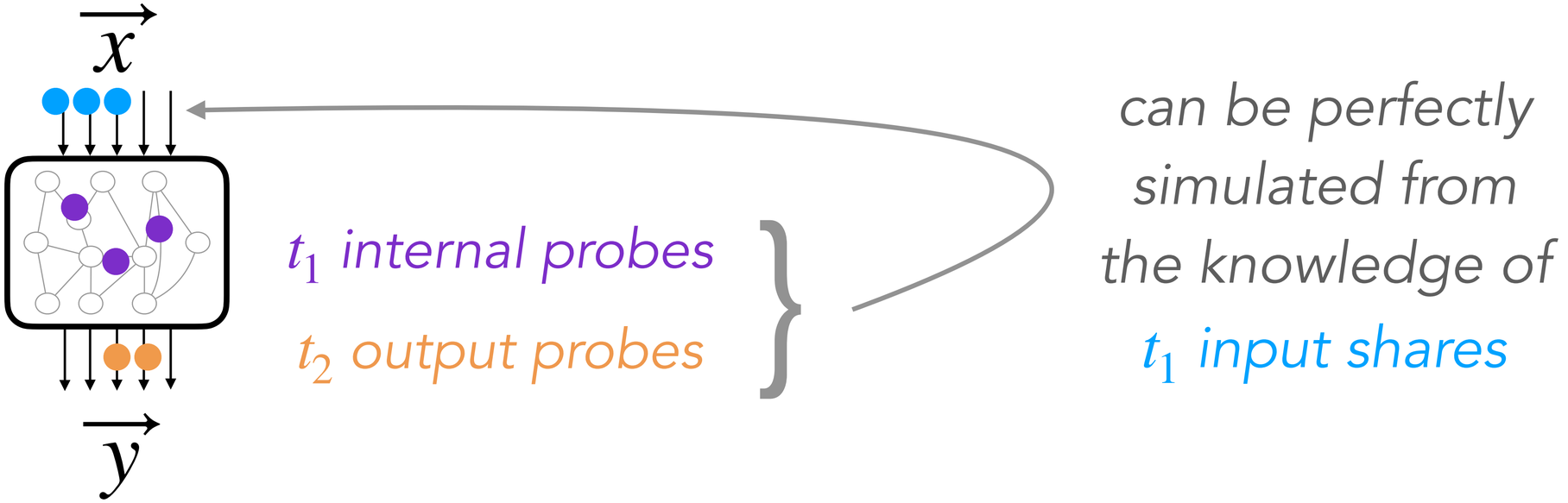
t_2 *output probes*

Composition security

⚠️ probing security for gadgets $\not\Rightarrow$ global (region) probing security

💡 composition security notions

Example: strong non-interference (SNI)

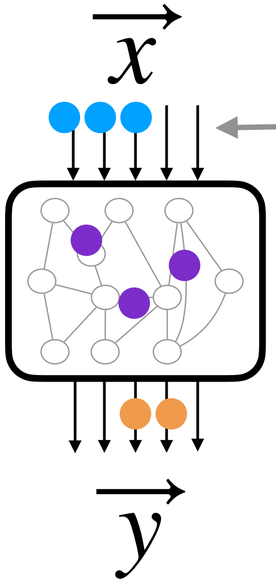


Composition security

⚠️ probing security for gadgets $\not\Rightarrow$ global (region) probing security

💡 composition security notions

Example: strong non-interference (SNI)



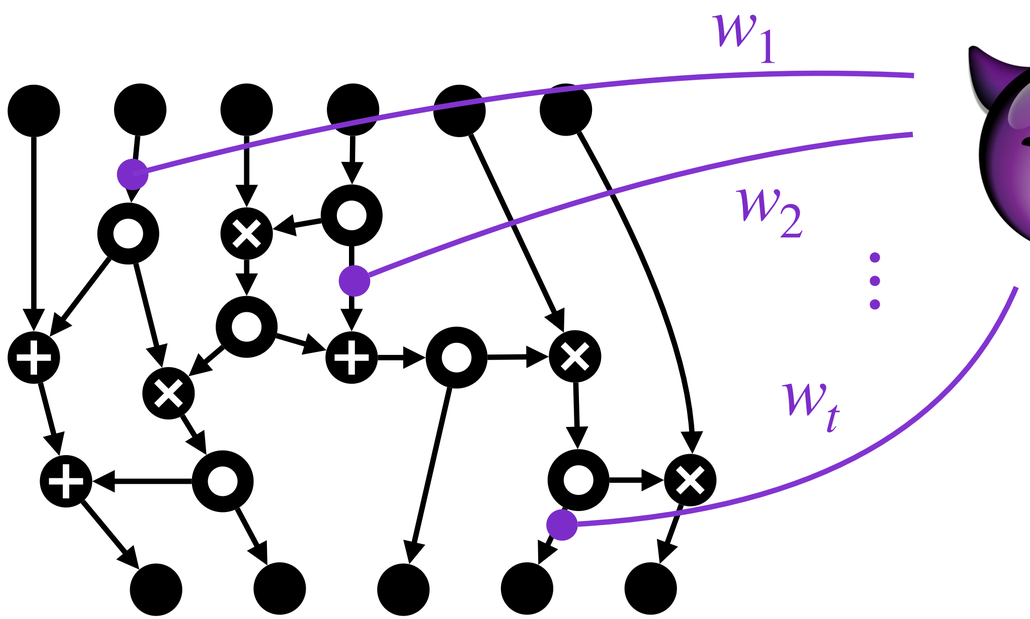
t_1 *internal probes*

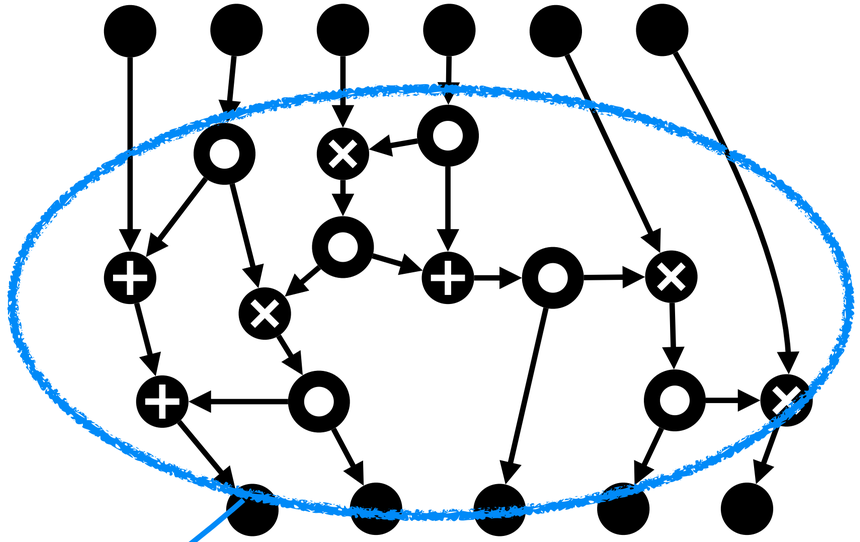
t_2 *output probes*

can be perfectly simulated from the knowledge of t_1 *input shares*

SNI gadgets ⇒ global region probing security

But… wait a minute!

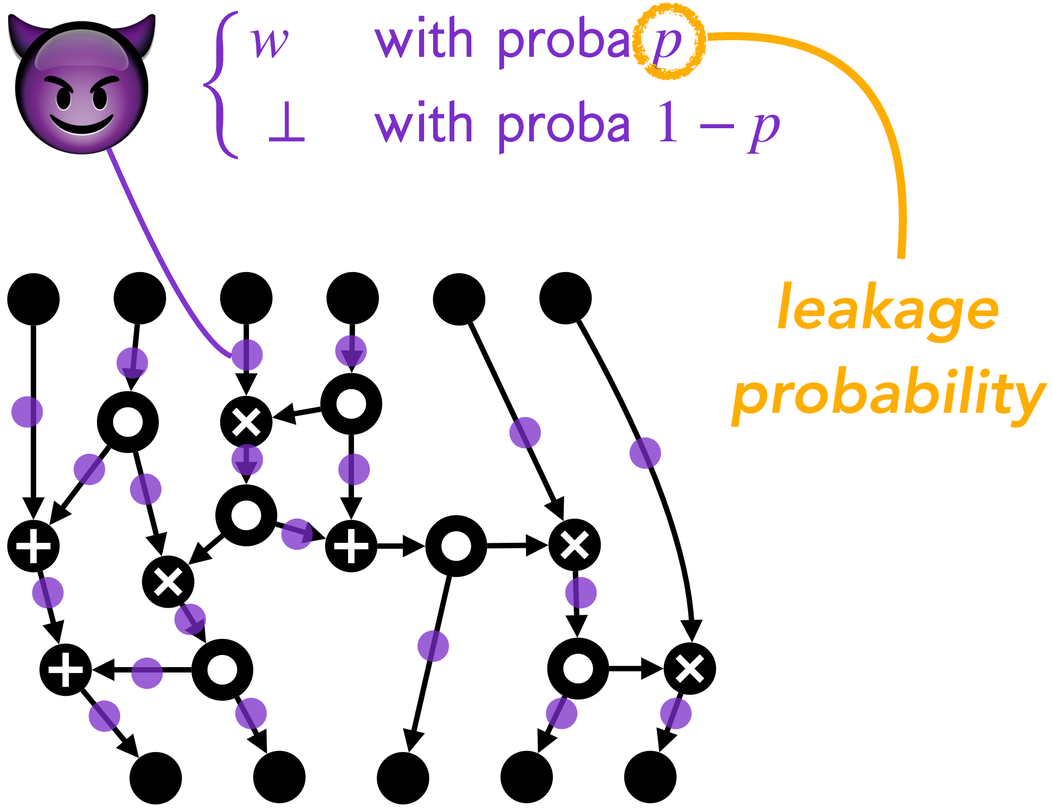




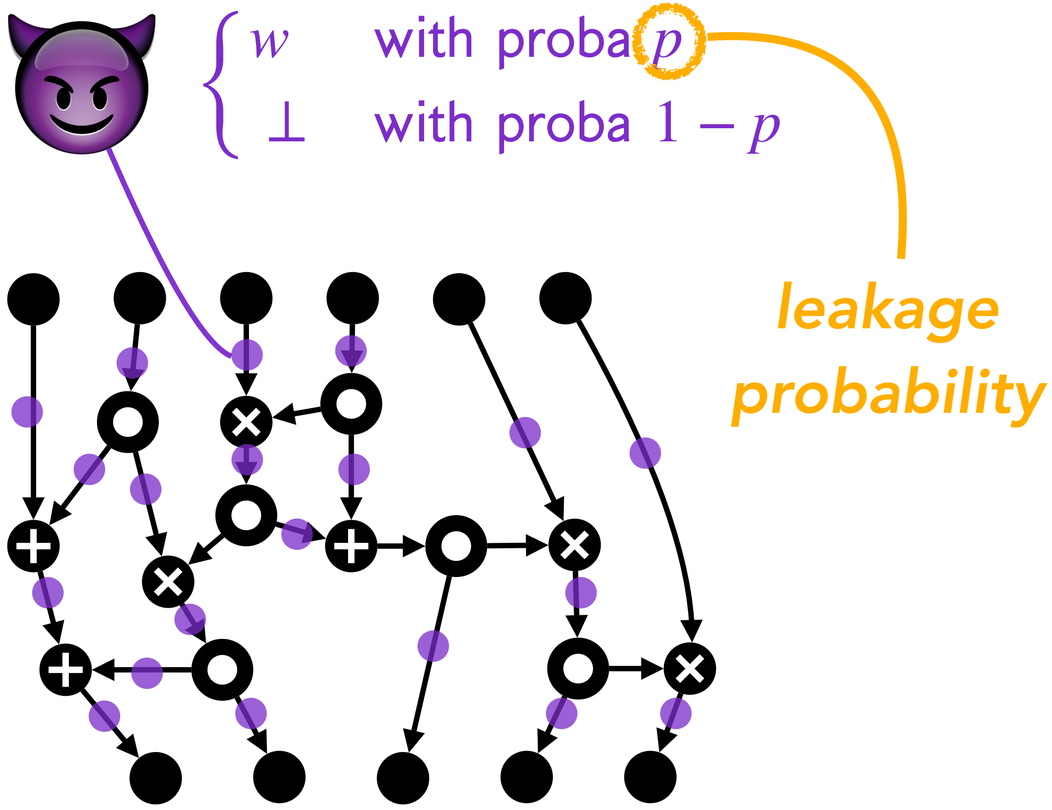
How do I get from probing security …

… to noisy leakage security?!

Random probing model

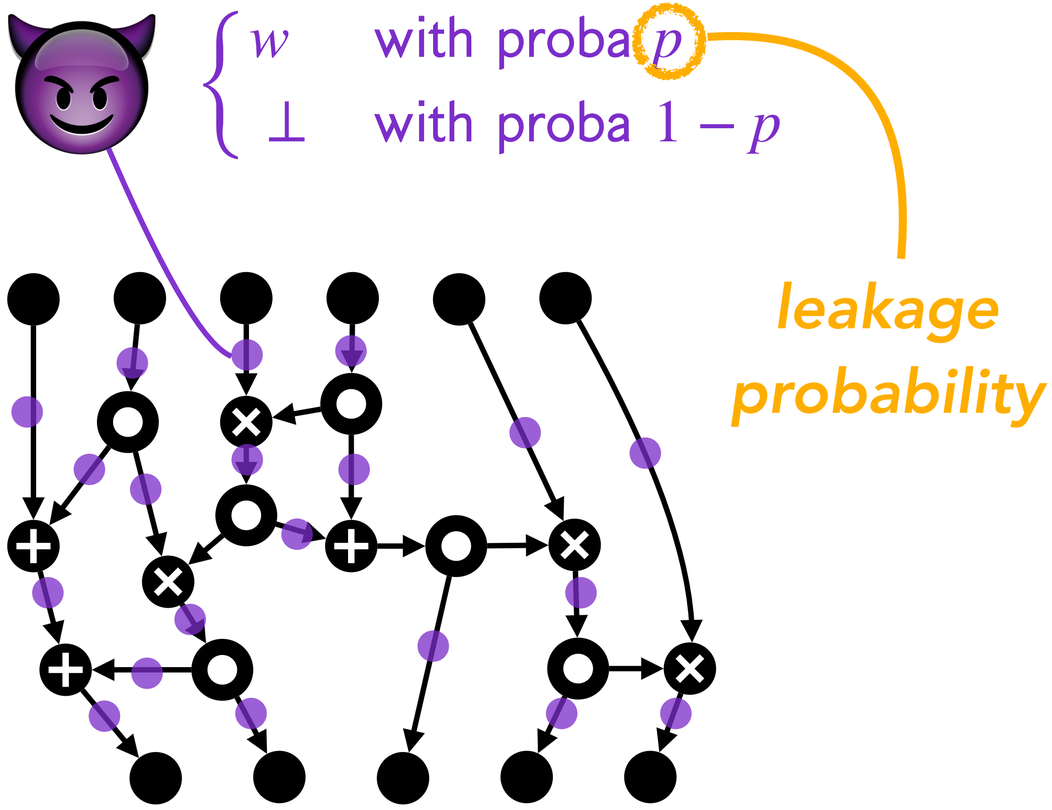


Random probing model



In average $p \cdot |G|$ leaking wires per gadget

Random probing model

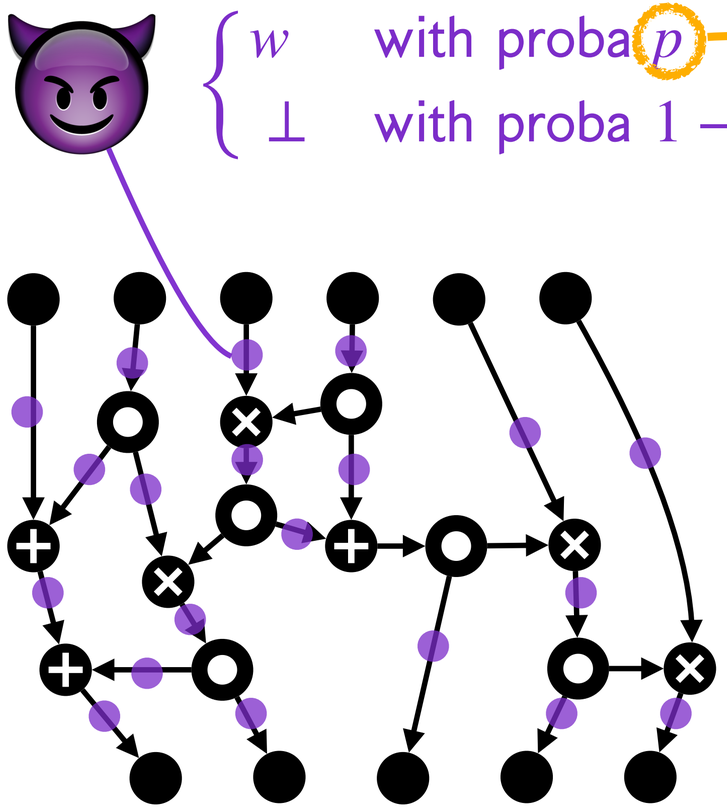


In average $p \cdot |G|$ leaking wires per gadget

Chernoff bound:

$< 2\,p\,|G|$ leaking wires with overwhelming probability

Random probing model



$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$

leakage probability

In average $p \cdot |G|$ leaking wires per gadget

Chernoff bound:

$< 2\,p\,|G|$ leaking wires with overwhelming probability

r-region probing security

$\Rightarrow$ p-random probing security

with $p = \Theta(r)$

Unifying probing and noisy models

Key lemma:

If f is δ-noisy, then $\exists\, f'$ s.t.

$$f(x) = f'(\phi(x))$$

where

$$\phi(x) := \begin{cases} x & \text{with proba } p \leq \delta \cdot |\mathcal{X}| \\ \perp & \text{with proba } 1 - p \end{cases}$$

Unifying probing and noisy models

Key lemma:

If f is δ-noisy, then $\exists\, f'$ s.t.

$$f(x) = f'(\phi(x))$$

where

$$\phi(x) := \begin{cases} x & \text{with proba } p \leq \delta \cdot |\mathcal{X}| \\ \perp & \text{with proba } 1 - p \end{cases}$$

⇒ *δ-noisy leakage can be simulated from p-random probing leakage*

Unifying probing and noisy models

Key lemma:

If f is δ-noisy, then $\exists\, f'$ s.t.

$$f(x) = f'(\phi(x))$$

where

$$\phi(x) := \begin{cases} x & \text{with proba } p \leq \delta \cdot |\mathcal{X}| \\ \perp & \text{with proba } 1 - p \end{cases}$$

⇒ *δ-noisy leakage can be simulated from p-random probing leakage*

Random probing leakage

$\phi(w_1),\ \phi(w_2),\ \ldots\ ,\ \phi(w_N)$

Unifying probing and noisy models

Key lemma:

If f is δ-noisy, then $\exists\, f'$ s.t.

$$f(x) = f'(\phi(x))$$

where

$$\phi(x) := \begin{cases} x & \text{with proba } p \leq \delta \cdot |\mathcal{X}| \\ \perp & \text{with proba } 1 - p \end{cases}$$

⇒ *δ-noisy leakage can be simulated from p-random probing leakage*

Random probing leakage

$$\phi(w_1),\ \phi(w_2),\ \ldots\ ,\ \phi(w_N)$$

Apply $f_1', \ldots, f_N'$

Noisy leakage

$$f_1(w_1),\ f_2(w_2),\ \ldots\ ,\ f_N(w_N)$$

Unifying probing and noisy models

Key lemma:

If f is δ-noisy, then $\exists\, f'$ s.t.

$$f(x) = f'(\phi(x))$$

where

$$\phi(x) := \begin{cases} x & \text{with proba } p \leq \delta \cdot |\mathcal{X}| \\ \perp & \text{with proba } 1-p \end{cases}$$

p-random probing security
$\Rightarrow$ δ-noisy security with $\delta = \Theta(p)$

$\Rightarrow$ *δ-noisy leakage can be simulated from p-random probing leakage*

Random probing leakage

$$\phi(w_1),\ \phi(w_2),\ \ldots\ ,\ \phi(w_N)$$

Apply f_1', …, f_N'

Noisy leakage

$$f_1(w_1),\ f_2(w_2),\ \ldots\ ,\ f_N(w_N)$$

Unifying probing and noisy models

r-region probing security

$\Rightarrow$

p-random probing security

with $p = \Theta(r)$

$\Rightarrow$

δ-noisy leakage security

with $\delta = \Theta(p)$

Unifying probing and noisy models

r-region probing security

$\Rightarrow$

p-random probing security with $p = \Theta(r)$

$\Rightarrow$

δ-noisy leakage security with $\delta = \Theta(p)$

leakage rate

$\begin{cases} 1 = \textit{lot of leakage (low noise)} \\ 0 = \textit{no leakage (infinite noise)} \end{cases}$

Unifying probing and noisy models

r-region probing security

$\Rightarrow$

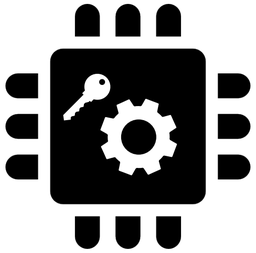
p-random probing security with $p = \Theta(r)$

$\Rightarrow$

δ-noisy leakage security with $\delta = \Theta(p)$

leakage rate

$\begin{cases} 1 = \textit{lot of leakage (low noise)} \\ 0 = \textit{no leakage (infinite noise)} \end{cases}$



the noise / leakage rate depends on the hardware

Unifying probing and noisy models

r-region probing security

$\Rightarrow$

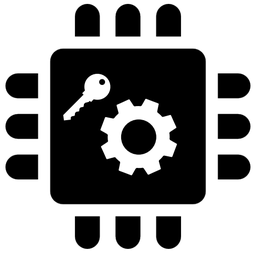
p-random probing security with $p = \Theta(r)$

$\Rightarrow$

δ-noisy leakage security with $\delta = \Theta(p)$

leakage rate

$\begin{cases} 1 = \textit{lot of leakage (low noise)} \\ 0 = \textit{no leakage (infinite noise)} \end{cases}$



the noise / leakage rate depends on the hardware

efficient masking schemes secure vs. constant (high) leakage rate

Secure schemes

State of the art

- State-of-the-art noisy-leakage-secure schemes
  - most schemes with **at least $\mathcal{O}(n^2)$ complexity**
  - a few schemes with $\mathcal{O}(1)$ leakage rate, but **constant not explicit**
- In what follows
  - region probing security in **quasilinear complexity**
  - random probing security with **explicit constant leakage rate**

Security in quasilinear complexity

Quasilinear masking

A $\overrightarrow{v}$-sharing of x

$$\overrightarrow{x} = (x_0, x_1, \ldots, x_{n-1}) \quad \text{s.t.} \quad \langle \overrightarrow{v}, \overrightarrow{x} \rangle = x$$

Quasilinear masking

A $\overrightarrow{v}$-sharing of x

$$\overrightarrow{x} = (x_0, x_1, \ldots, x_{n-1}) \quad \text{s.t.} \quad \langle \overrightarrow{v}, \overrightarrow{x} \rangle = x = \sum_{i=0}^{n-1} x_i \cdot \omega^i$$

$$\overrightarrow{v} = (1, \omega, \omega^2, \ldots, \omega^{n-1}) \quad \text{for} \quad \omega \xleftarrow{\$} \mathbb{F}$$

Quasilinear masking

Polynomial $P_{\vec{x}}(\omega)$
(shares = coefficients)

A $\vec{v}$-sharing of x

$$\vec{x} = (x_0, x_1, \ldots, x_{n-1}) \quad \text{s.t.} \quad \langle \vec{v}, \vec{x} \rangle = x = \sum_{i=0}^{n-1} x_i \cdot \omega^i$$

$$\vec{v} = (1, \omega, \omega^2, \ldots, \omega^{n-1}) \quad \text{for} \quad \omega \xleftarrow{\$} \mathbb{F}$$

Efficient multiplication

- Let $\vec{t}$ such that

$$P_{\vec{t}} = P_{\vec{x}} \cdot P_{\vec{y}}$$

- We get

$$P_{\vec{t}}(\omega) = \sum_{i=0}^{2n-1} t_i \omega^i = x \cdot y$$

Efficient multiplication

- Let $\vec{t}$ such that

$$P_{\vec{t}} = P_{\vec{x}} \cdot P_{\vec{y}}$$

- We get

$$P_{\vec{t}}(\omega) = \sum_{i=0}^{2n-1} t_i \, \omega^i = x \cdot y$$

Efficient multiplication

- Let $\vec{t}$ such that

$$P_{\vec{t}} = P_{\vec{x}} \cdot P_{\vec{y}}$$

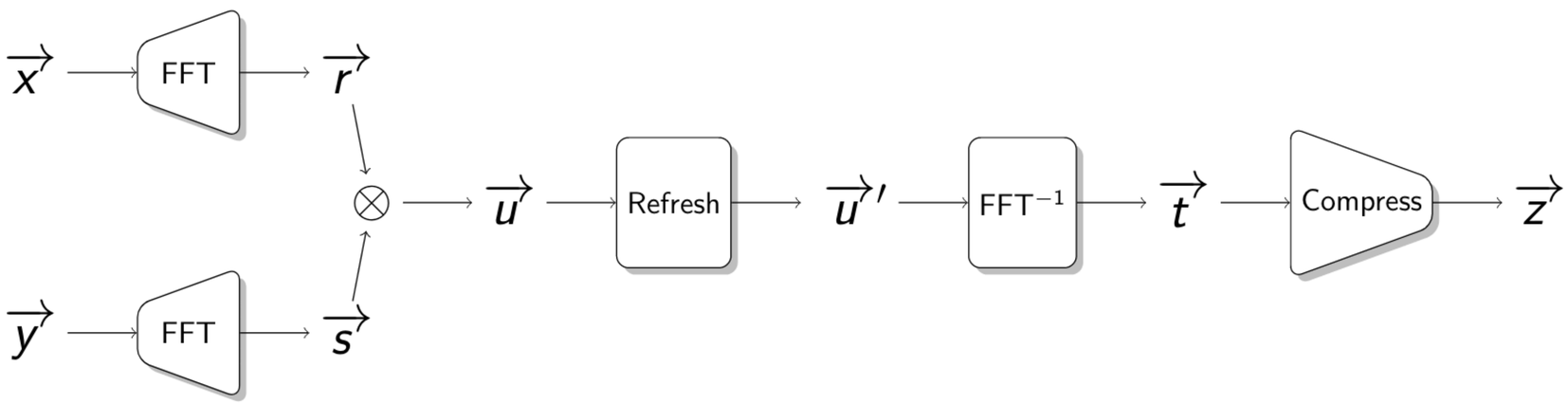
- We get

$$P_{\vec{t}}(\omega) = \sum_{i=0}^{2n-1} t_i \, \omega^i = x \cdot y$$

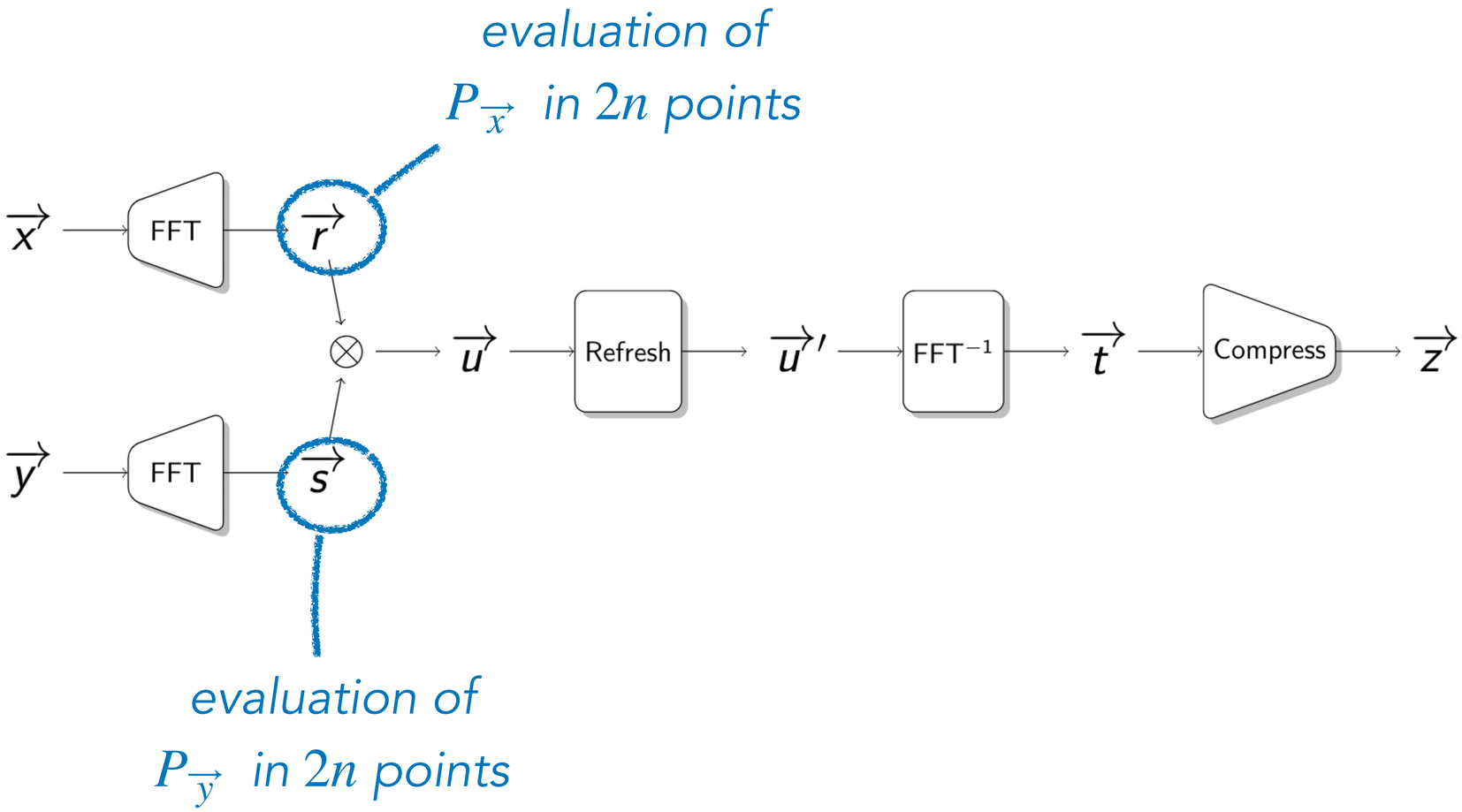
- Compression:

$$\vec{z} = (t_0, \ldots, t_{n-1}) + \omega^n \cdot (t_n, \ldots, t_{2n-1})$$

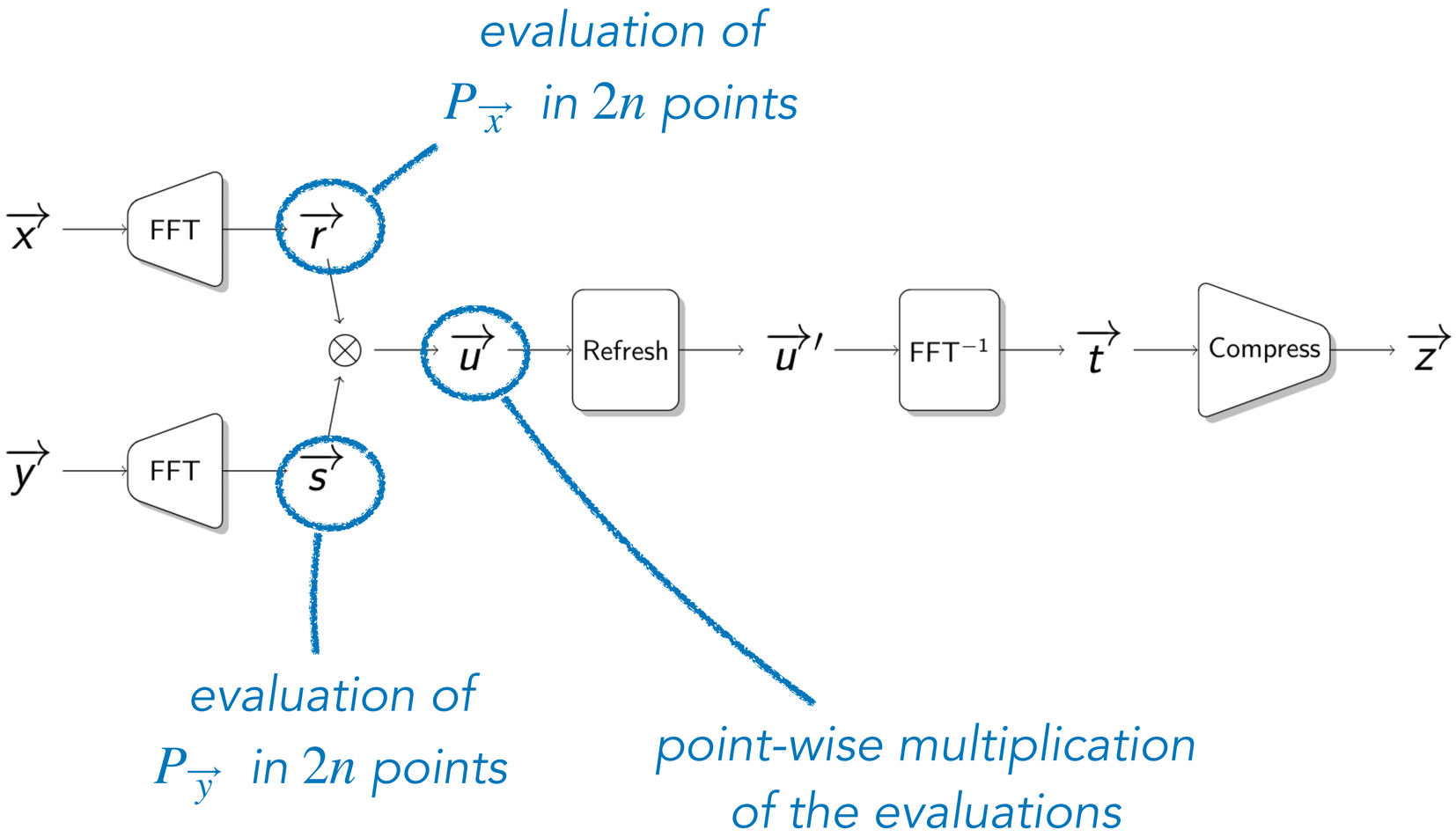
Multiplication gadget



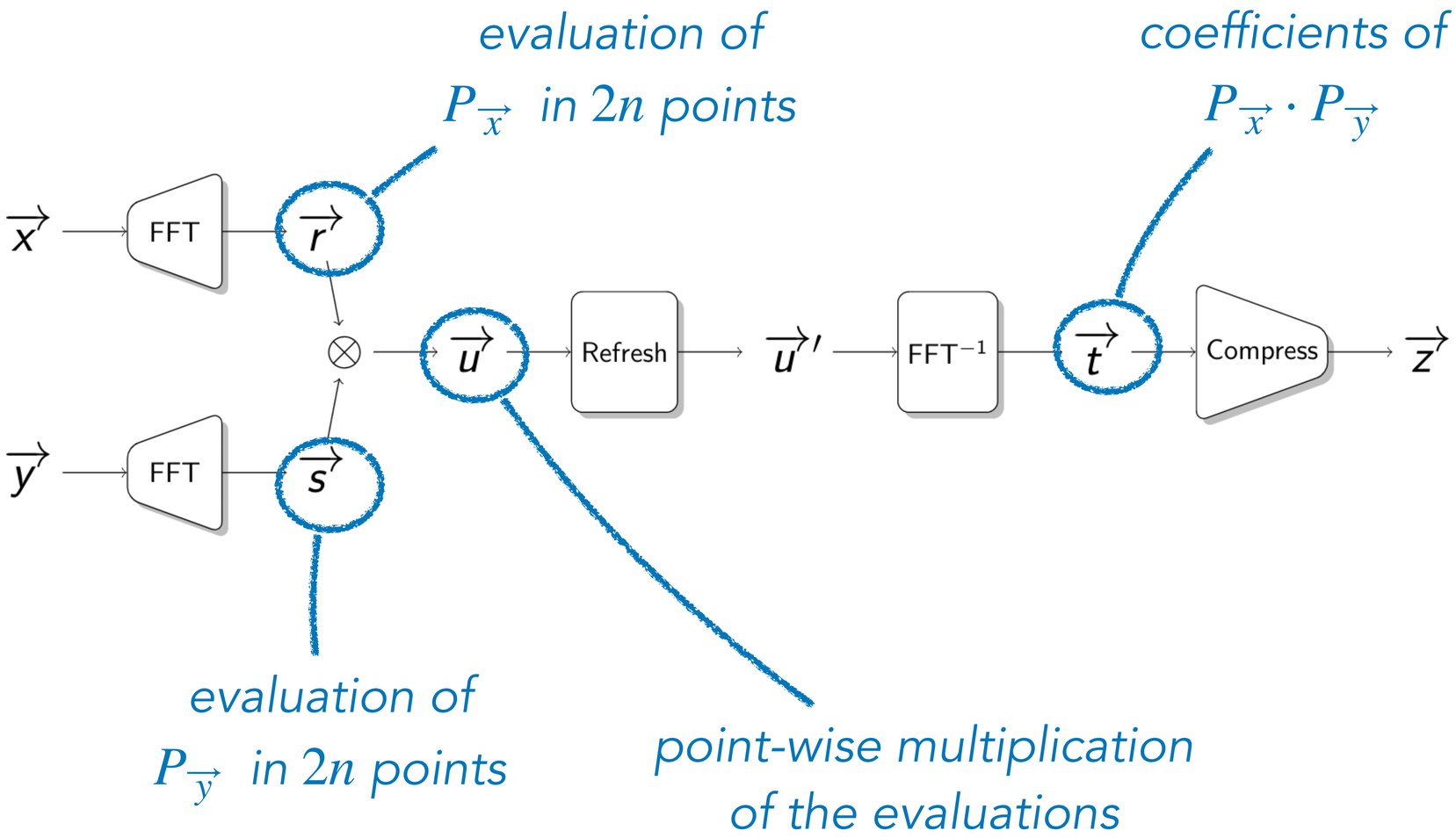
Multiplication gadget



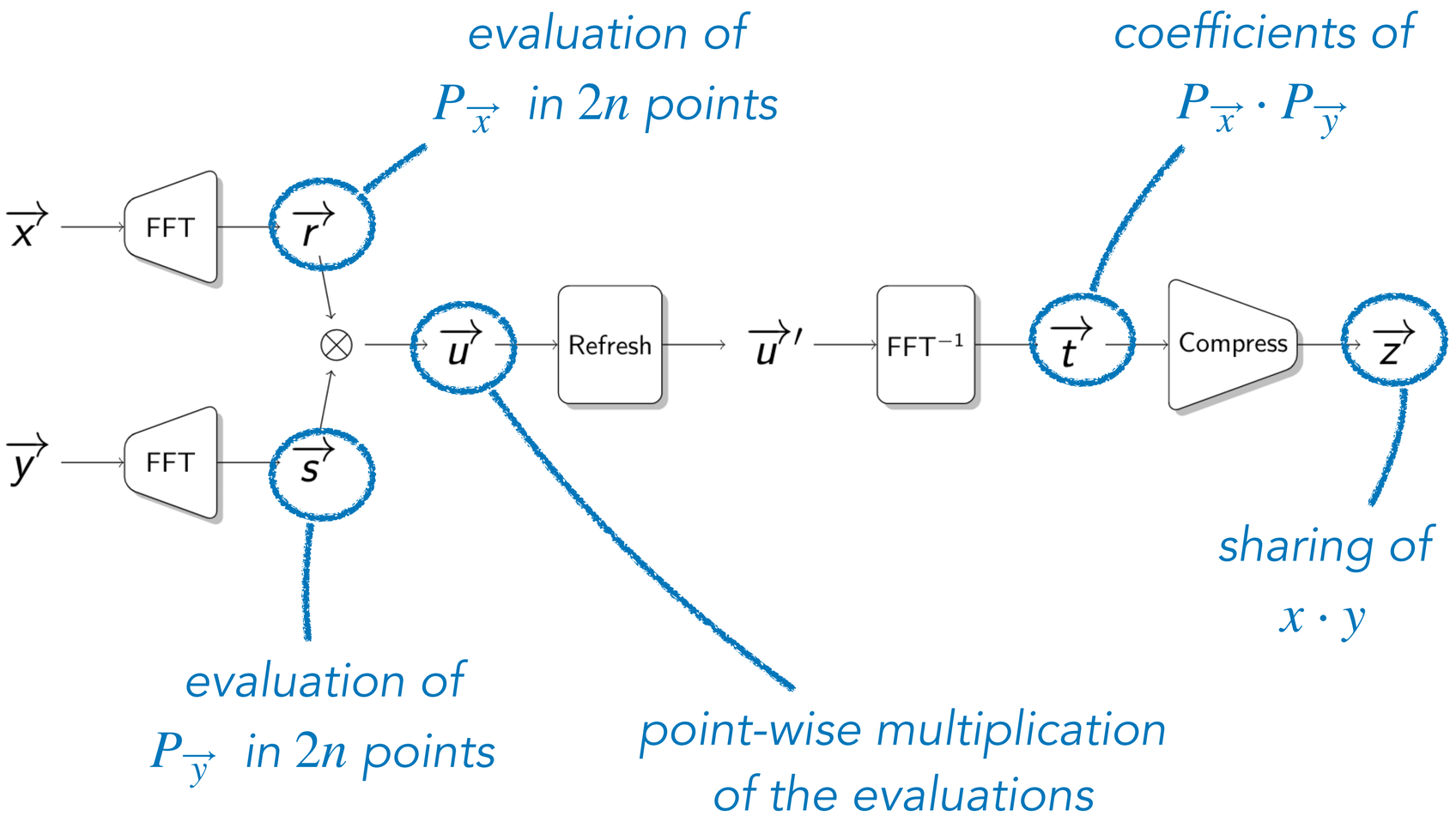
Multiplication gadget



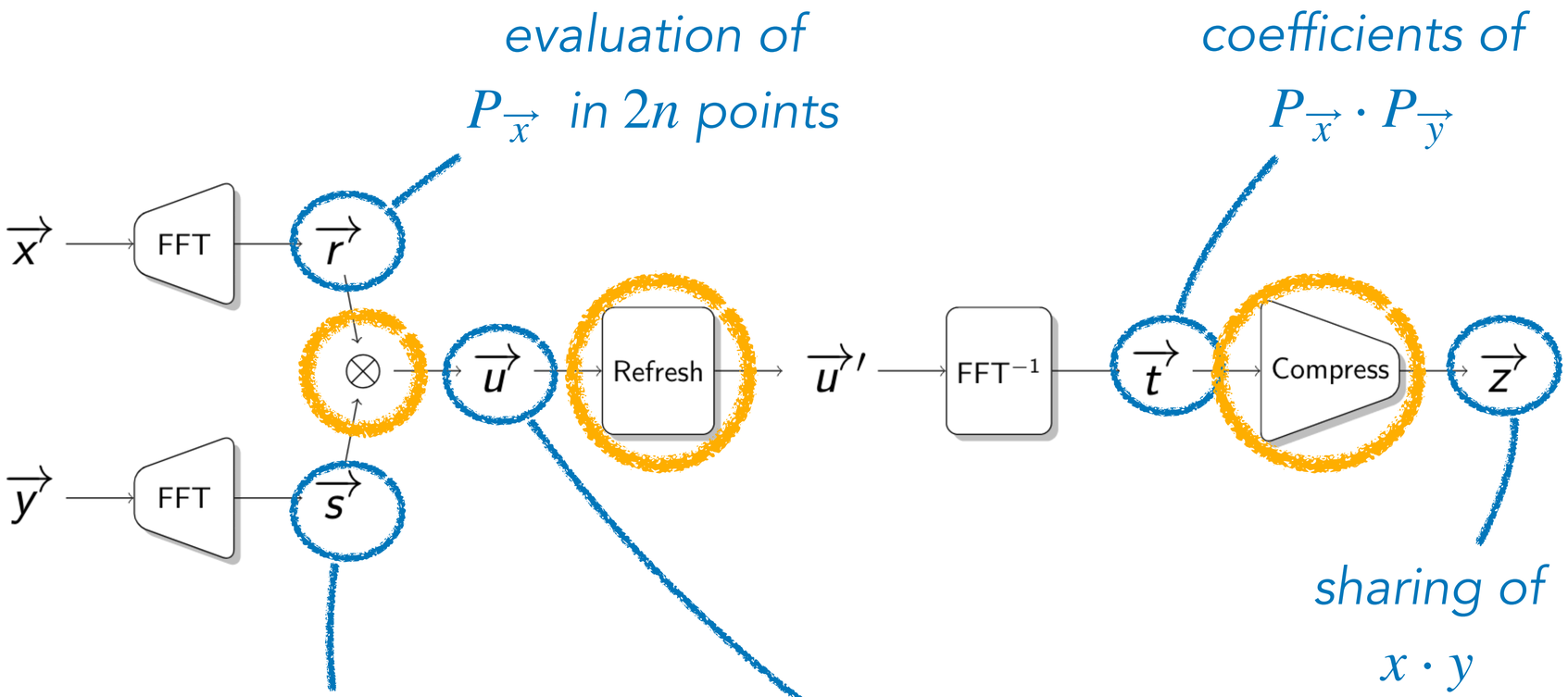
Multiplication gadget



Multiplication gadget



Multiplication gadget



evaluation of $P_{\vec{x}}$ *in* $2n$ *points*

coefficients of $P_{\vec{x}} \cdot P_{\vec{y}}$

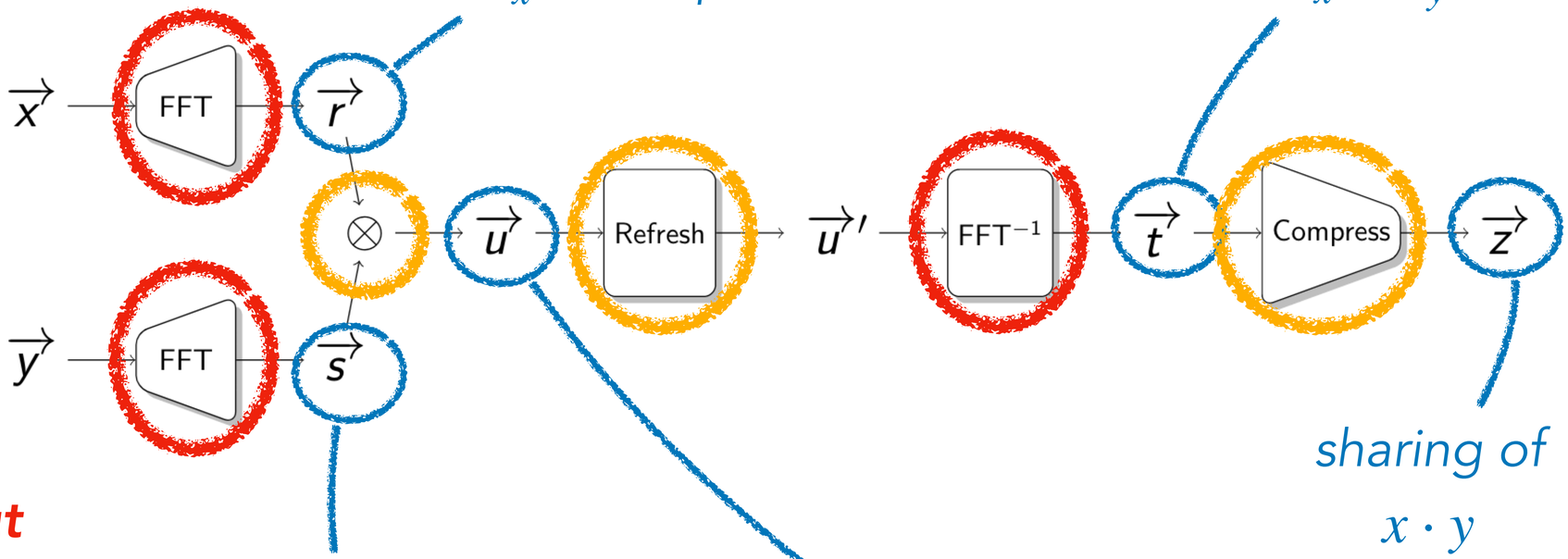
sharing of $x \cdot y$

evaluation of $P_{\vec{y}}$ *in* $2n$ *points*

point-wise multiplication of the evaluations

sharewise operations $\Rightarrow$ ***probing secure***

Multiplication gadget



evaluation of $P_{\vec{x}}$ *in* $2n$ *points*

coefficients of $P_{\vec{x}} \cdot P_{\vec{y}}$

sharing of $x \cdot y$

evaluation of $P_{\vec{y}}$ *in* $2n$ *points*

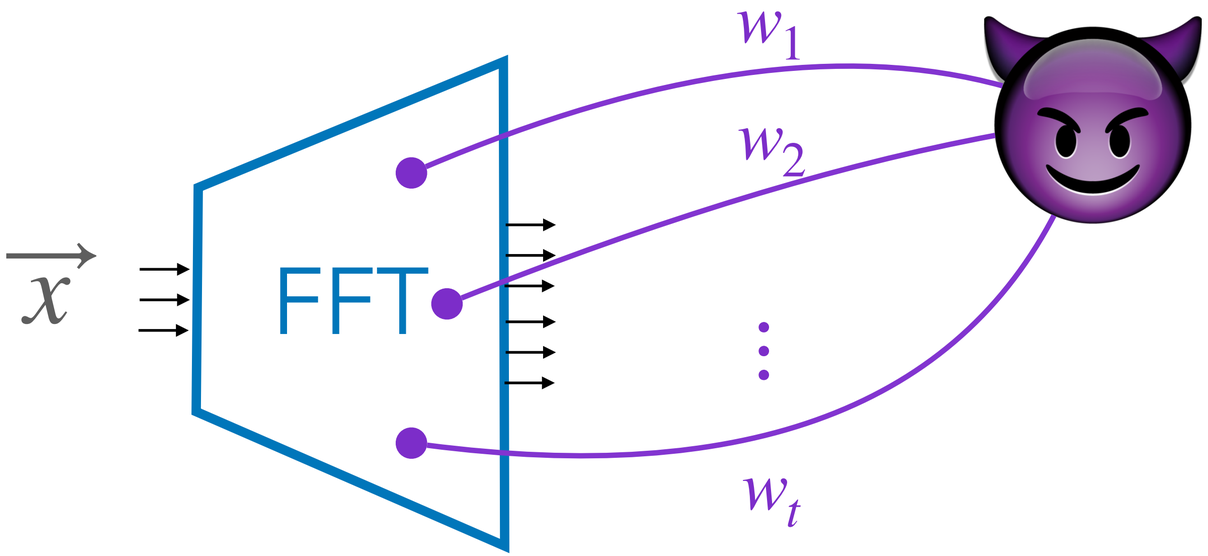
point-wise multiplication of the evaluations

what about the FFT?

🤔

sharewise operations ⇒ ***probing secure***

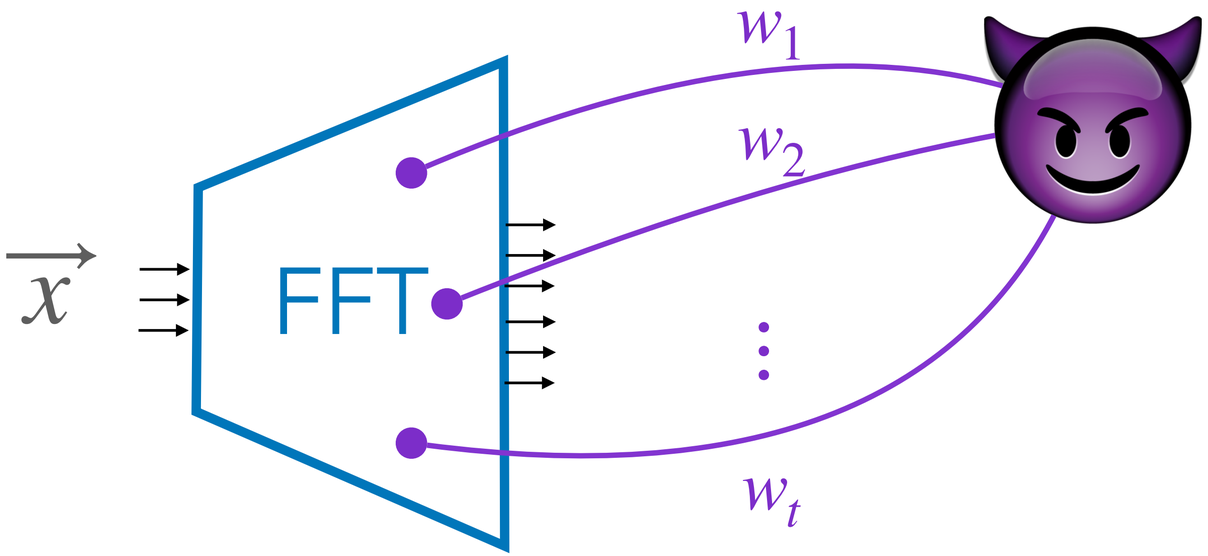
Probing security



💡 *FFT computes linear combinations of the x_i's*

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_t \end{pmatrix} = \begin{bmatrix} A \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix}$$

Probing security



💡 *FFT computes linear combinations of the x_i's*

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_t \end{pmatrix} = \begin{bmatrix} A \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix}$$

Lemma 1

If $\vec{v} = \begin{pmatrix} \omega^0 \\ \omega^1 \\ \vdots \\ \omega^{n-1} \end{pmatrix} \notin \left\langle \begin{bmatrix} A \end{bmatrix} \right\rangle$ then $\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_t \end{pmatrix} \sim \mathcal{U}(\mathbb{F}^t)$

(assuming A full rank wlog)

Probing security

Lemma 2

$\exists$ at most t values of $\omega \in \mathbb{F}$ s.t. $\vec{v} = \begin{pmatrix} \omega^0 \\ \omega^1 \\ \vdots \\ \omega^{n-1} \end{pmatrix} \in \langle \left[\; A \; \right] \rangle$

Probing security

Lemma 2

$\exists$ at most t values of $\omega \in \mathbb{F}$ s.t. $\vec{v} = \begin{pmatrix} \omega^0 \\ \omega^1 \\ \vdots \\ \omega^{n-1} \end{pmatrix} \in \langle \left[\; A \; \right] \rangle$

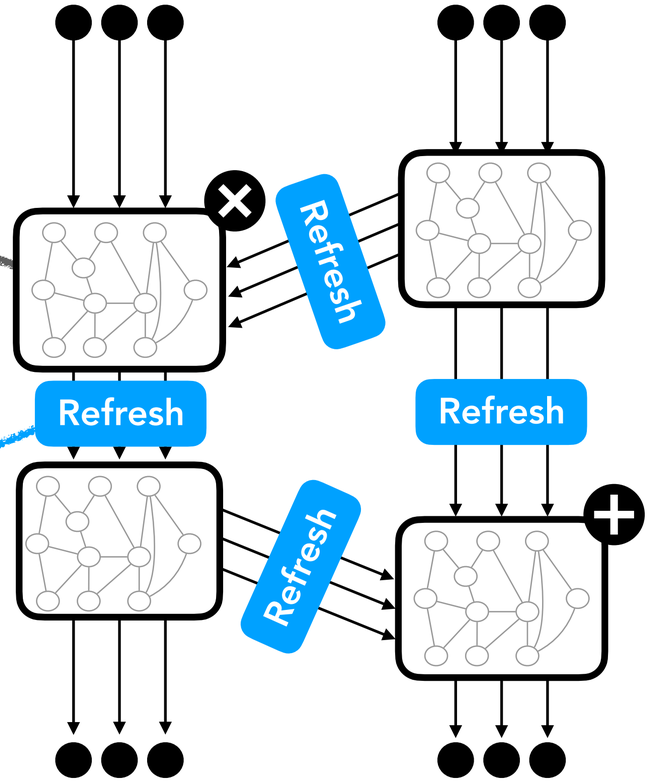
Lemma 1 + Lemma 2

$$P\left[(w_1, \ldots, w_t) \text{ cannot be simulated} \right] \leq \frac{t}{|\mathbb{F}|} < \frac{n}{|\mathbb{F}|}$$

Composition security

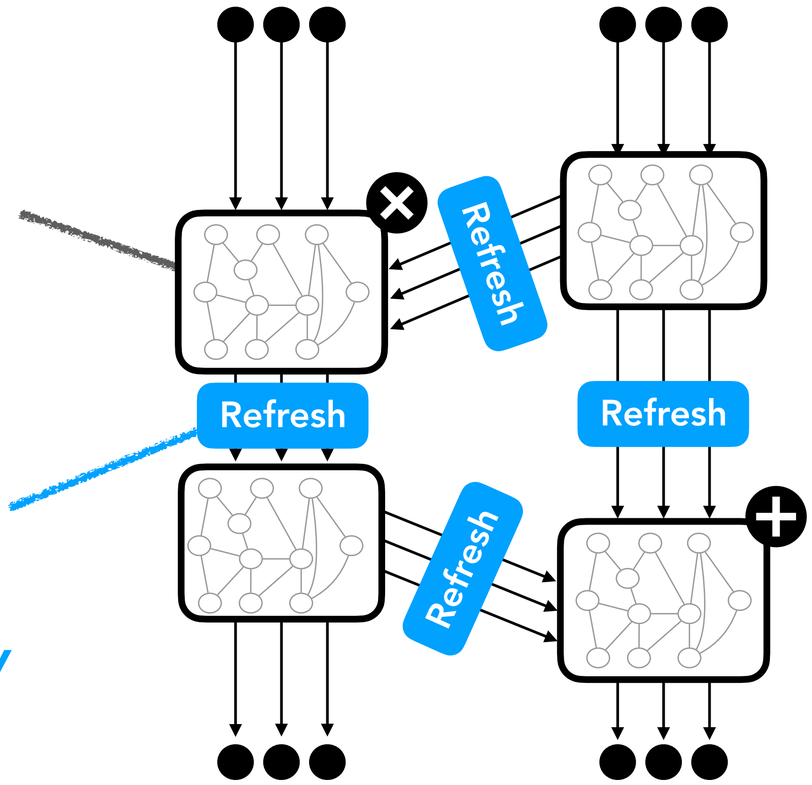
probing security

input-output separation (IOS) property



⇒ region probing security

Composition security



⇒ region probing security

Wrapping up:

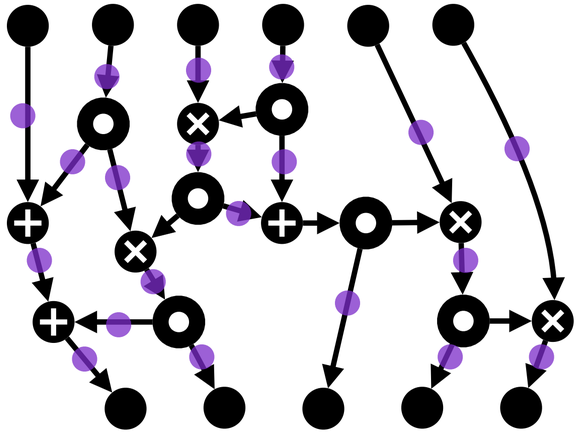
- Gadget complexity: $\Theta(n \log n)$
- Probes per gadget: $\Theta(n)$
- Leakage rate: $\Theta(1/\log n)$

Security with constant leakage rate

Simulation with abort

1. Sample a set of leaking wires

$$W \leftarrow \mathsf{LeakingWires}(\hat{C}, p)$$



$$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$$

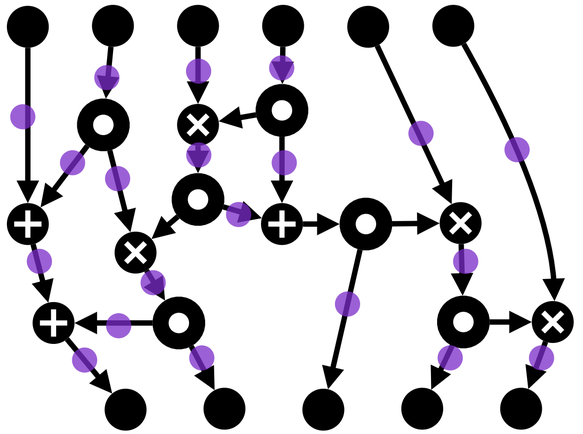
Simulation with abort

1. Sample a set of leaking wires

$$W \leftarrow \mathsf{LeakingWires}(\hat{C}, p)$$

2. Simulate the corresponding wire values

$$\mathsf{Sim} : W \mapsto \begin{cases} \text{perfect simulation} \\ \perp \text{ (abort)} \end{cases}$$



$$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$$

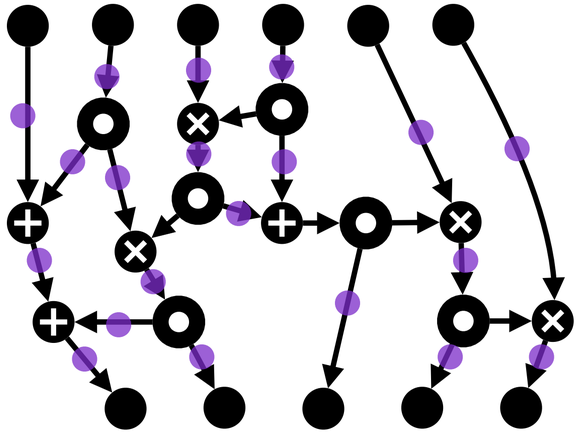
Simulation with abort

1. Sample a set of leaking wires

$$W \leftarrow \mathsf{LeakingWires}(\hat{C}, p)$$

2. Simulate the corresponding wire values

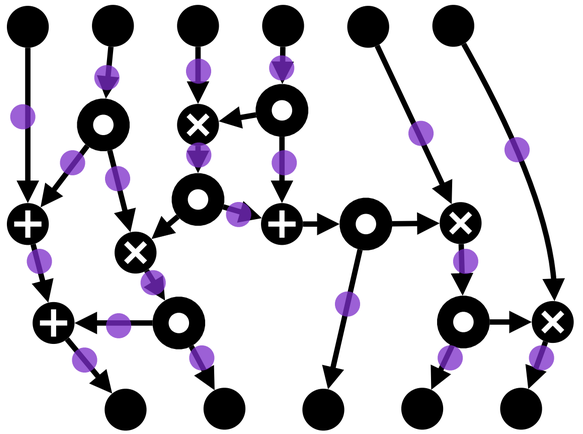
$$\mathsf{Sim} : W \mapsto \begin{cases} \text{perfect simulation} \\ \perp \text{ (abort)} \end{cases}$$



$$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$$

$$\delta_W = \begin{cases} 1 & \text{if } \mathsf{Sim}(W) = \perp \\ 0 & \text{otherwise} \end{cases}$$

Simulation with abort



1. Sample a set of leaking wires

$$W \leftarrow \mathsf{LeakingWires}(\hat{C}, p)$$

2. Simulate the corresponding wire values

$$\mathsf{Sim} : W \mapsto \begin{cases} \text{perfect simulation} \\ \perp \text{ (abort)} \end{cases}$$

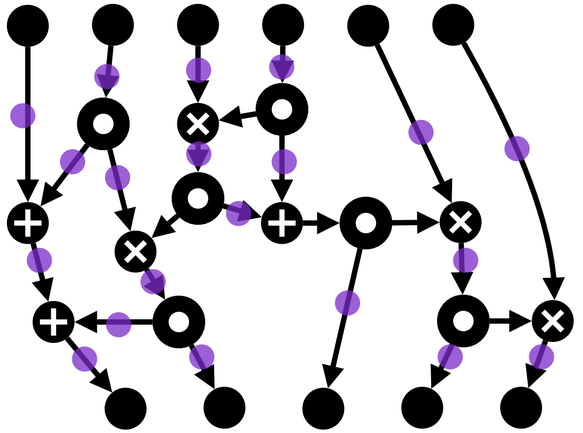
$$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$$

$$\delta_W = \begin{cases} 1 & \text{if } \mathsf{Sim}(W) = \perp \\ 0 & \text{otherwise} \end{cases}$$

- Failure probability

$$f(p) = \sum_W \delta_W p^{|W|}(1-p)^{s-|W|}$$

Simulation with abort



$$\begin{cases} w & \text{with proba } p \\ \perp & \text{with proba } 1-p \end{cases}$$

1. Sample a set of leaking wires

$$W \leftarrow \mathsf{LeakingWires}(\hat{C}, p)$$

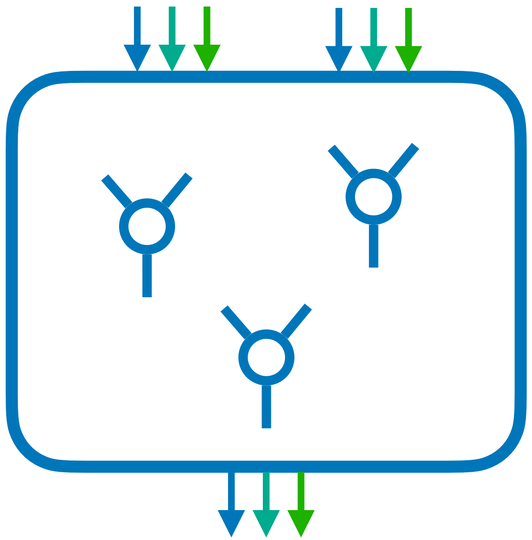
2. Simulate the corresponding wire values

$$\mathsf{Sim} : W \mapsto \begin{cases} \text{perfect simulation} \\ \perp \text{ (abort)} \end{cases} \qquad \delta_W = \begin{cases} 1 & \text{if } \mathsf{Sim}(W) = \perp \\ 0 & \text{otherwise} \end{cases}$$

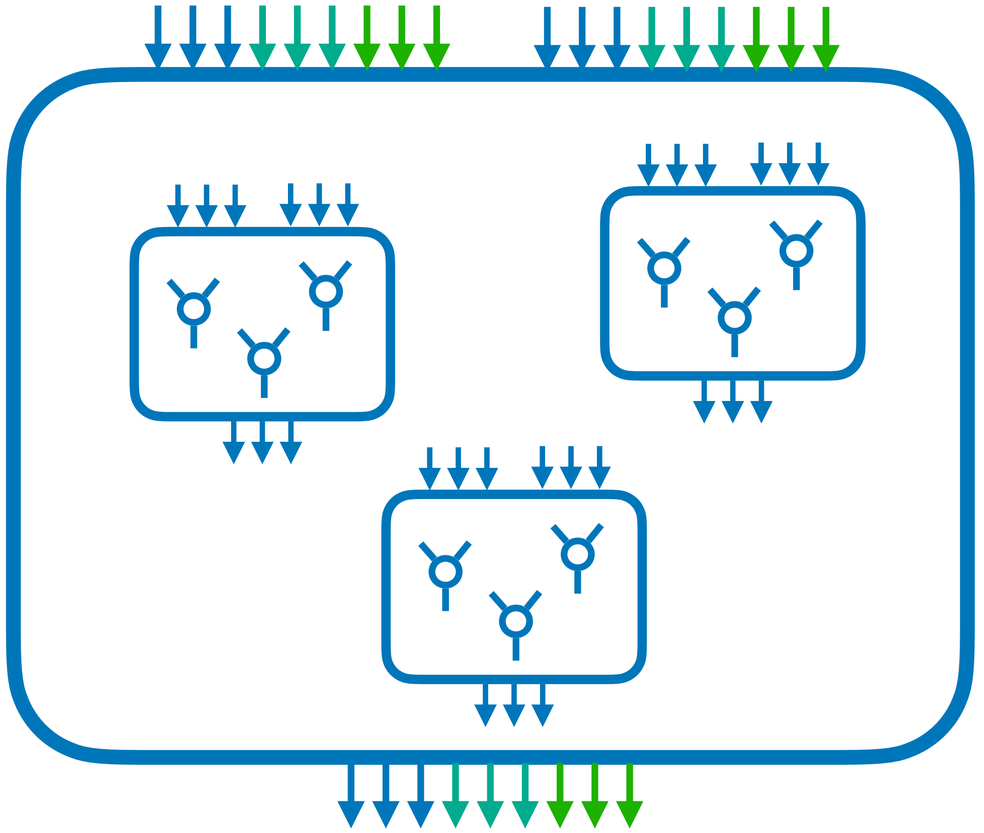
- Failure probability

$$f(p) = \sum_W \delta_W p^{|W|}(1-p)^{s-|W|} \leq \sum_i c_i \, p^i$$

The expansion strategy

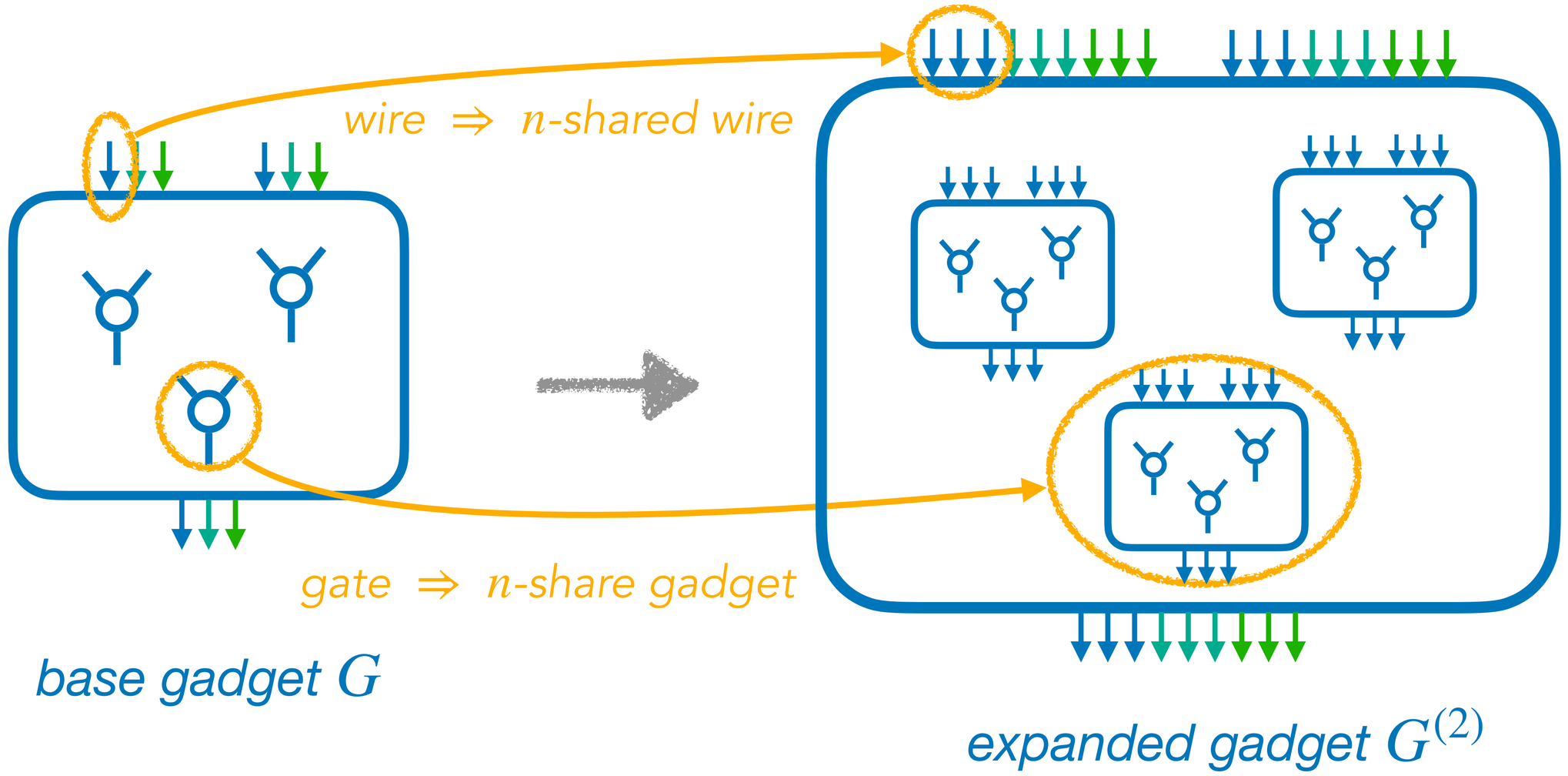


base gadget G



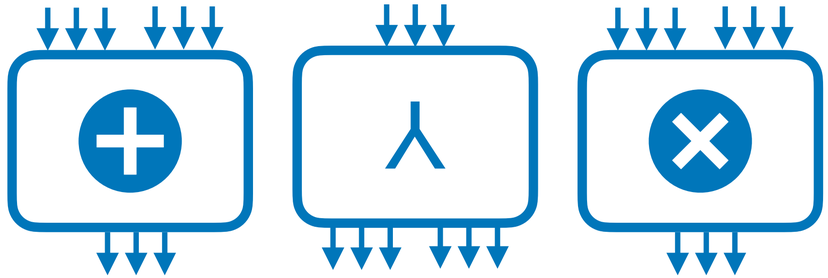
expanded gadget $G^{(2)}$

The expansion strategy



The expansion strategy

Idea: bootstrap constant-size (small) gadgets

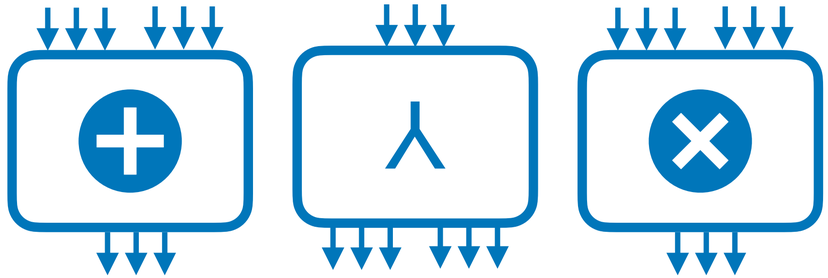


base gadgets

$$\{G\} \to \{G^{(2)}\} \to \cdots \to \{G^{(k)}\}$$

The expansion strategy

Idea: bootstrap constant-size (small) gadgets



base gadgets

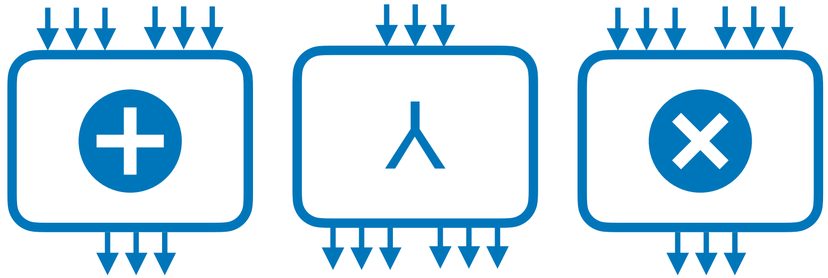
$$\{G\} \to \{G^{(2)}\} \to \cdots \to \{G^{(k)}\}$$

Goal: amplification of random probing security

$$p \longrightarrow f(p)$$

The expansion strategy

Idea: bootstrap constant-size (small) gadgets



base gadgets

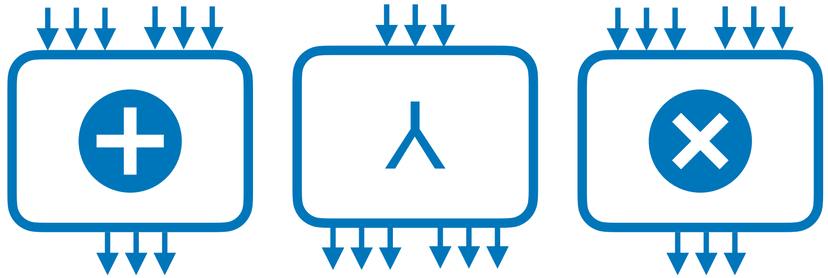
$$\{G\} \to \{G^{(2)}\} \to \cdots \to \{G^{(k)}\}$$

Goal: amplification of random probing security

$$p \longrightarrow f(p) \longrightarrow f(f(p))$$

The expansion strategy

Idea: bootstrap constant-size (small) gadgets



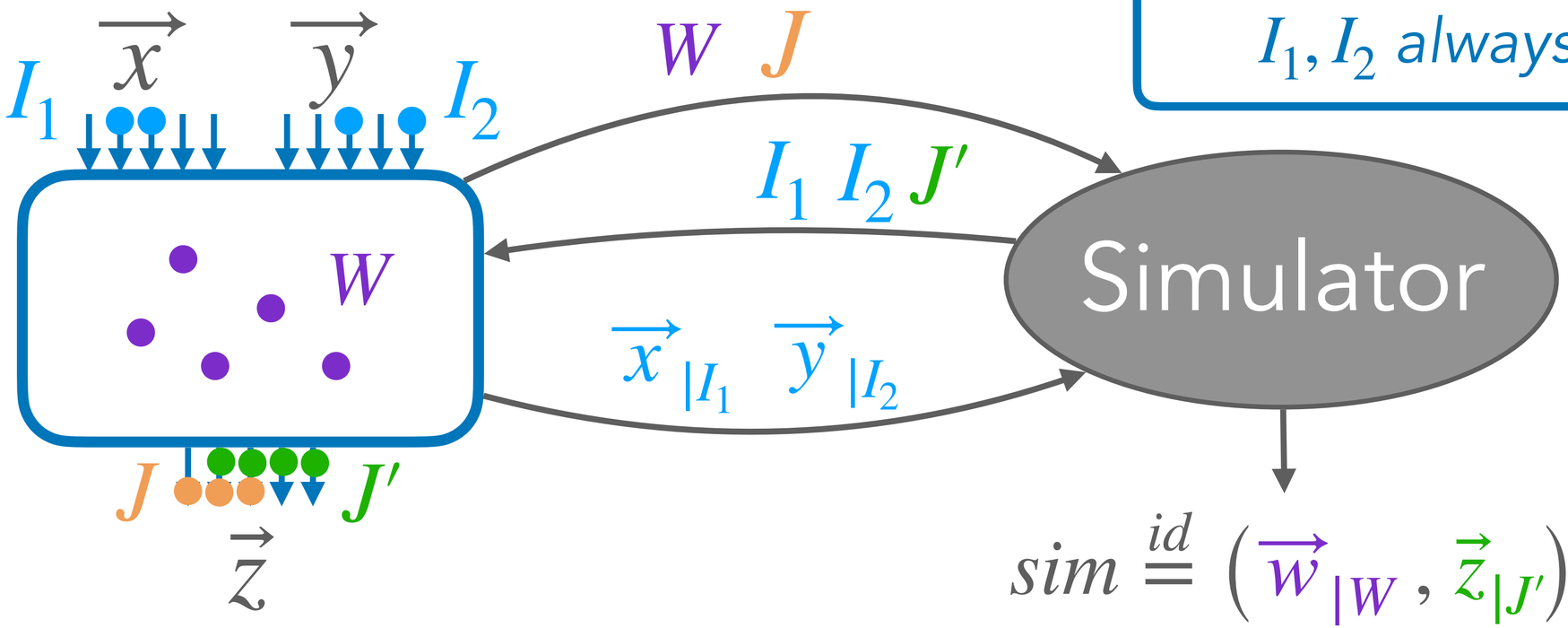
base gadgets

$$\{G\} \to \{G^{(2)}\} \to \cdots \to \{G^{(k)}\}$$

Goal: amplification of random probing security

$$p \longrightarrow f(p) \longrightarrow f(f(p)) \longrightarrow \cdots \longrightarrow f^{(k)}(p)$$

Random probing expandability (RPE)



I_1, I_2 *always exist*

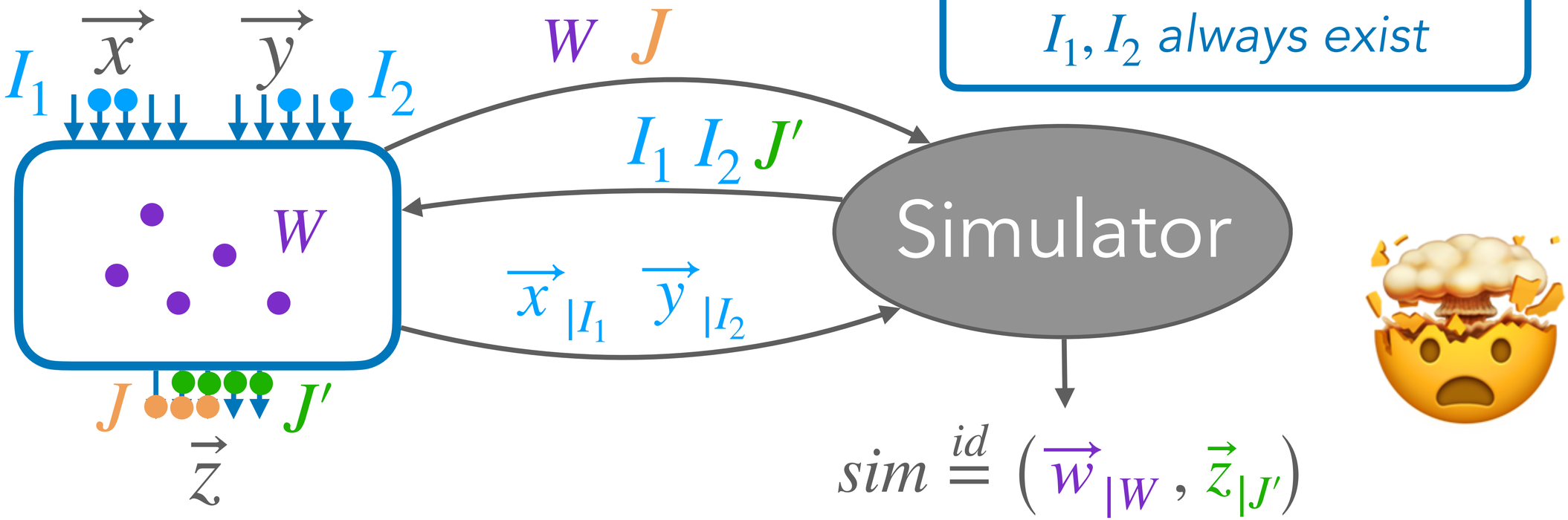
RPE threshold t: $|J| \leq t$,

$(|I_1| > t$ *or* $|I_2| > t)$ = *simulation failure*

if $|J| > t$, *sim. can choose*

J' *s.t.* $|J'| = n - 1$

Random probing expandability (RPE)



I_1, I_2 *always exist*

RPE threshold t: $|J| \leq t$,

$(|I_1| > t$ *or* $|I_2| > t)$ = *simulation failure*

if $|J| > t$, *sim. can choose*

J' *s.t.* $|J'| = n - 1$

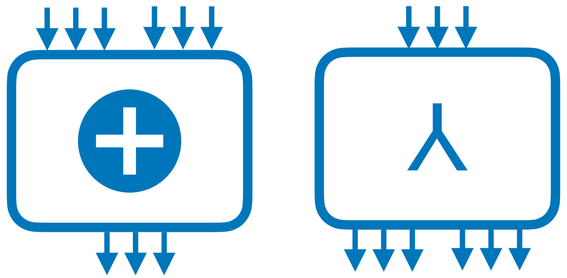
Random probing expandability (RPE)

Base gadgets $\{G\}$ f-RPE $\Rightarrow$ expanded gadgets $\{G^{(2)}\}$ $f^{(2)}$-RPE

$\Rightarrow$ expanded gadgets $\{G^{(k)}\}$ $f^{(k)}$-RPE

$f^{(k)}(p)$ simulation security vs.
p-random probing leakage

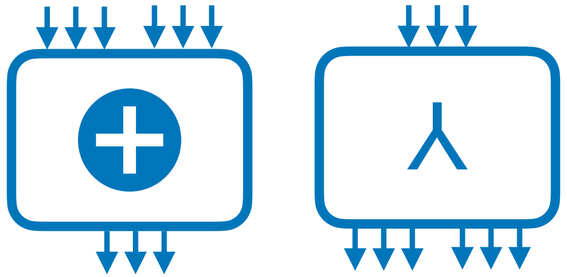
Generic constructions



$$G_{\oplus}(\overrightarrow{x}, \overrightarrow{y}) = G_{\mathsf{R}}(\overrightarrow{x}) + G_{\mathsf{R}}(\overrightarrow{y})$$

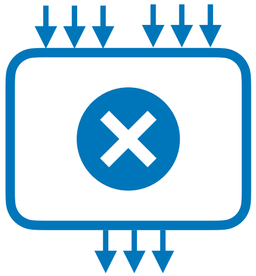
$$G_{\curlywedge}(\overrightarrow{x}) = (G_{\mathsf{R}}(\overrightarrow{x}), G_{\mathsf{R}}(\overrightarrow{x}))$$

Generic constructions



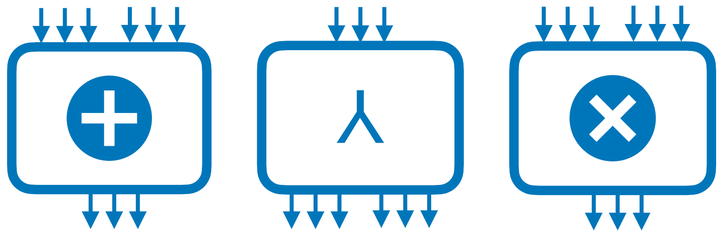
$$G_{\oplus}(\overrightarrow{x}, \overrightarrow{y}) = G_{\mathsf{R}}(\overrightarrow{x}) + G_{\mathsf{R}}(\overrightarrow{y})$$

$$G_{\curlywedge}(\overrightarrow{x}) = (G_{\mathsf{R}}(\overrightarrow{x}), G_{\mathsf{R}}(\overrightarrow{x}))$$



$$G_{\otimes}(\overrightarrow{x}, \overrightarrow{y}) \mapsto \begin{pmatrix} x_1 \cdot G_{\mathsf{R}}(\overrightarrow{y}) \\ x_2 \cdot G_{\mathsf{R}}(\overrightarrow{y}) \\ \vdots \\ x_n \cdot G_{\mathsf{R}}(\overrightarrow{y}) \end{pmatrix} + \text{ greedy use of randomness}$$

Efficient instantiations

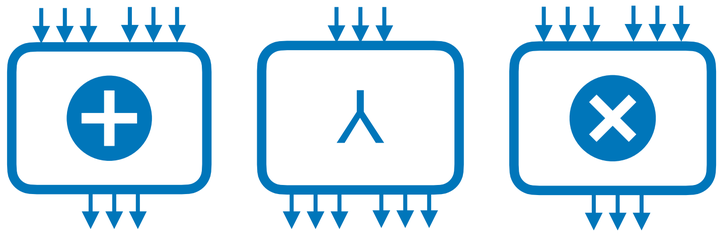


constant-size (small) gadgets

Your gadget is f-RPE

IronMask

Efficient instantiations



constant-size (small) gadgets

IronMask

Your gadget is f-RPE

Maximum tolerated leakage probability

$p_{max} \in [0,1)$ such that $f(p_{max}) < p_{max}$

Efficient instantiations

3-share gadgets

$$\left.\begin{aligned} G_{\mathsf{R}} : z_1 &\leftarrow r_1 + x_1 \\ z_2 &\leftarrow r_2 + x_2 \\ z_3 &\leftarrow (r_1 + r_2) + x_3 \end{aligned}\right\} \Rightarrow \quad \mathcal{O}(|C|\kappa^{3.9}) \ , \quad p_{max} = 2^{-7.5}$$

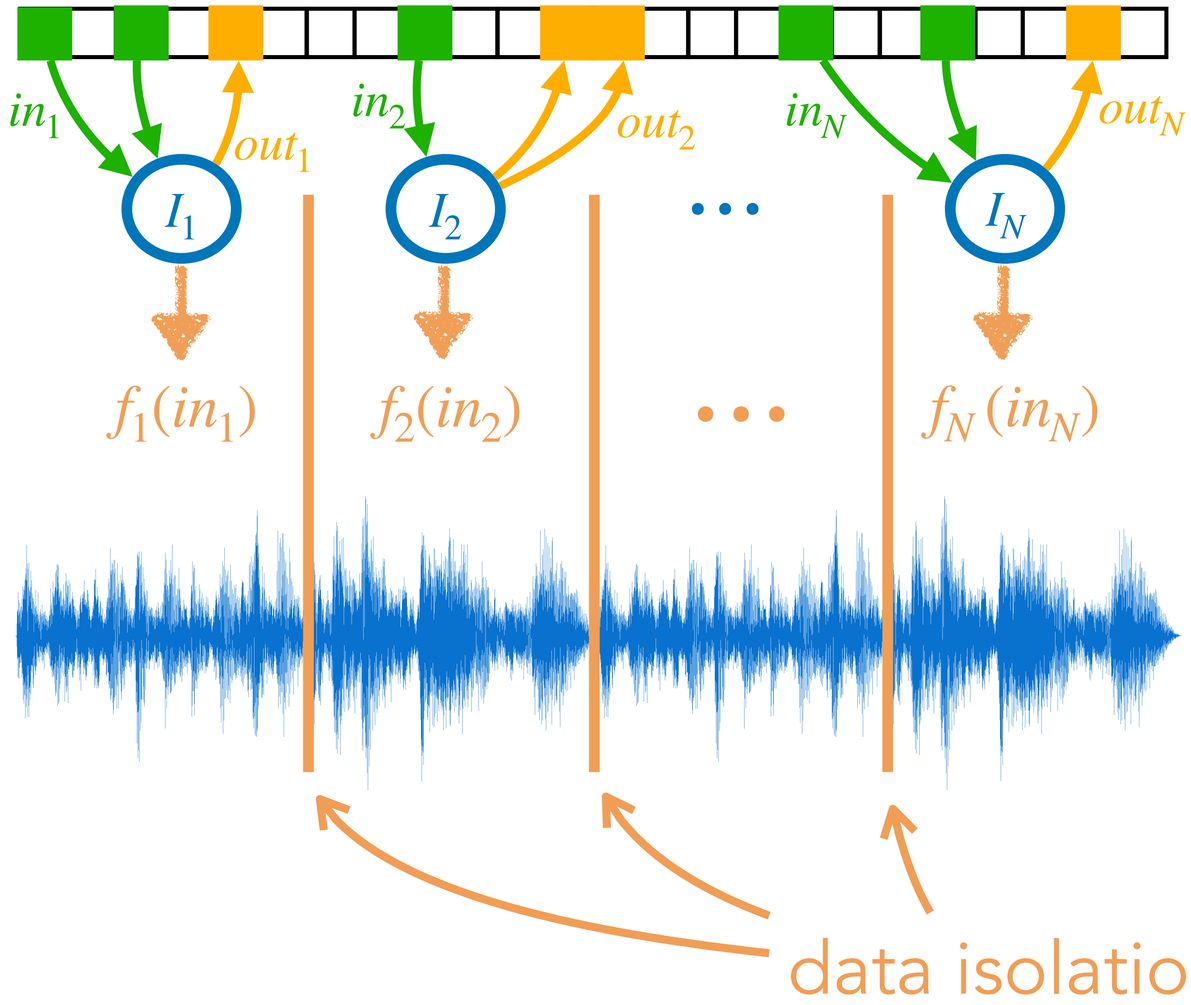
5-share gadgets

$$\left.\begin{aligned} G_{\mathsf{R}} : z_1 &\leftarrow (r_1 + r_2) + x_1 \\ z_2 &\leftarrow (r_2 + r_3) + x_2 \\ z_3 &\leftarrow (r_3 + r_4) + x_3 \\ z_4 &\leftarrow (r_4 + r_5) + x_4 \\ z_5 &\leftarrow (r_5 + r_1) + x_5 \end{aligned}\right\} \Rightarrow \quad \mathcal{O}(|C|\kappa^{3.2}) \ , \quad p_{max} = 2^{-12}$$

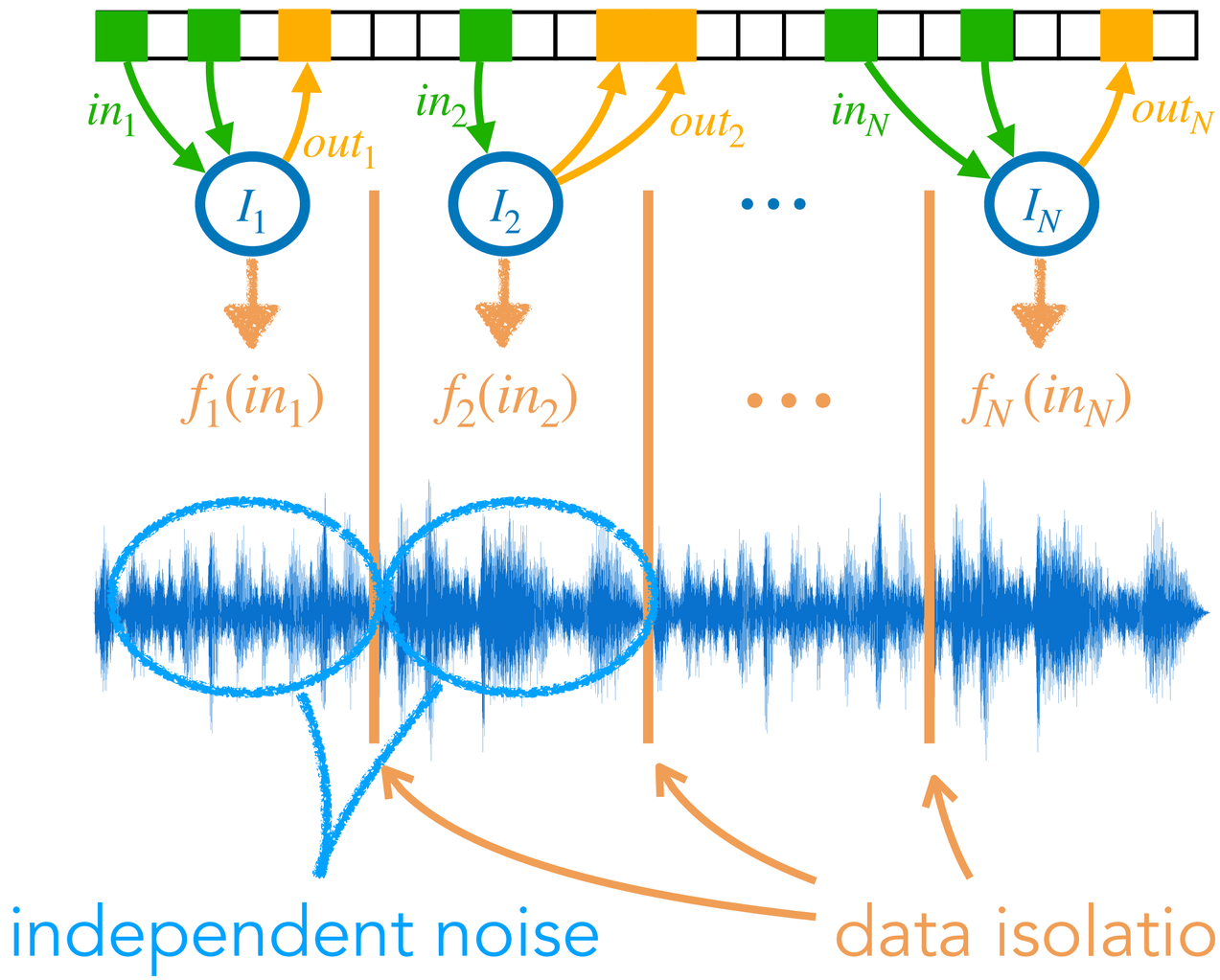
Conclusion

- [x] Provable security against side-channel attacks
- [ ] (Fully) bridging theory and practice

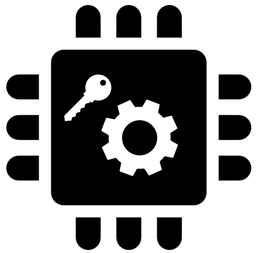
Physical assumptions



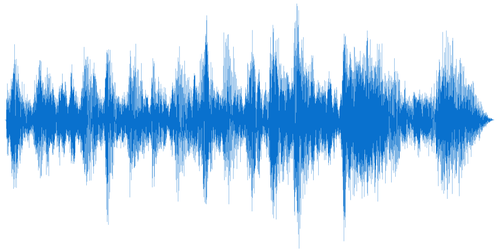
Physical assumptions



Noise parameters



device /
implementation



leakage
distribution

δ

noise
parameter

p

leakage
probability

Performances

3-share gadgets

$$G_R : z_1 \leftarrow r_1 + x_1$$
$$z_2 \leftarrow r_2 + x_2$$
$$z_3 \leftarrow (r_1 + r_2) + x_3$$

5-share gadgets

$$G_R : z_1 \leftarrow (r_1 + r_2) + x_1$$
$$z_2 \leftarrow (r_2 + r_3) + x_2$$
$$z_3 \leftarrow (r_3 + r_4) + x_3$$
$$z_4 \leftarrow (r_4 + r_5) + x_4$$
$$z_5 \leftarrow (r_5 + r_1) + x_5$$

$\Rightarrow$ $\mathcal{O}(|C|\kappa^{3.9})$ $p_{max} = 2^{-7.5}$

→ improved complexity

$\Rightarrow$ $\mathcal{O}(|C|\kappa^{3.2})$ $p_{max} = 2^{-12}$

→ optimised implementations

VeriSiCC Seminar 2022

Verification and Generation of Side-Channel Countermeasures

September 22, 2022, Paris

https://cryptoexperts.com/verisicc/seminaire-2022.html

Thank you!

References

- **Differential Power Analysis** — Paul C. Kocher, Joshua Jaffe, Benjamin Jun: Differential Power Analysis. CRYPTO 1999

- **Masking / soundness of masking with noise** — Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi: Towards Sound Approaches to Counteract Power-Analysis Attacks. CRYPTO 1999

- **Masking applied to DES** — Louis Goubin, Jacques Patarin: DES and Differential Power Analysis (The "Duplication" Method). CHES 1999

- **Probing model / ISW scheme** — Yuval Ishai, Amit Sahai, David A. Wagner: Private Circuits: Securing Hardware against Probing Attacks. CRYPTO 2003

- **“Only computation leaks” model** — Silvio Micali, Leonid Reyzin: Physically Observable Cryptography. TCC 2004

References

- **Noisy leakage model** — Emmanuel Prouff, Matthieu Rivain: Masking against Side-Channel Attacks: A Formal Security Proof. EUROCRYPT 2013

- **Unifying probing and noisy models** — Alexandre Duc, Stefan Dziembowski, Sebastian Faust: Unifying Leakage Models: From Probing Attacks to Noisy Leakage. EUROCRYPT 2014

- **Composition security for masking** — Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, Rébecca Zucchini: Strong Non-Interference and Type-Directed Higher-Order Masking. CCS 2016

- **Random probing expansion strategy** — Prabhanjan Ananth, Yuval Ishai, Amit Sahai: Private Circuits: A Modular Approach. CRYPTO 2018

References

- **Quasilinear masking**
  - Dahmun Goudarzi, Antoine Joux, Matthieu Rivain: How to Securely Compute with Noisy Leakage in Quasilinear Complexity. ASIACRYPT 2018
  - Dahmun Goudarzi, Thomas Prest, Matthieu Rivain, Damien Vergnaud: Probing Security through Input-Output Separation and Revisited Quasilinear Masking. IACR TCHES 2021

- **Random probing expandability**
  - Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb: Random Probing Security: Verification, Composition, Expansion and New Constructions. CRYPTO 2020
  - Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb: On the Power of Expansion: More Efficient Constructions in the Random Probing Model. EUROCRYPT 2021

References

- **IronMask tool** — Sonia Belaïd, Darius Mercadier, Matthieu Rivain, Abdul Rahman Taleb: IronMask: Versatile Verification of Masking Security. IEEE S&P 2022