Random Probing Security

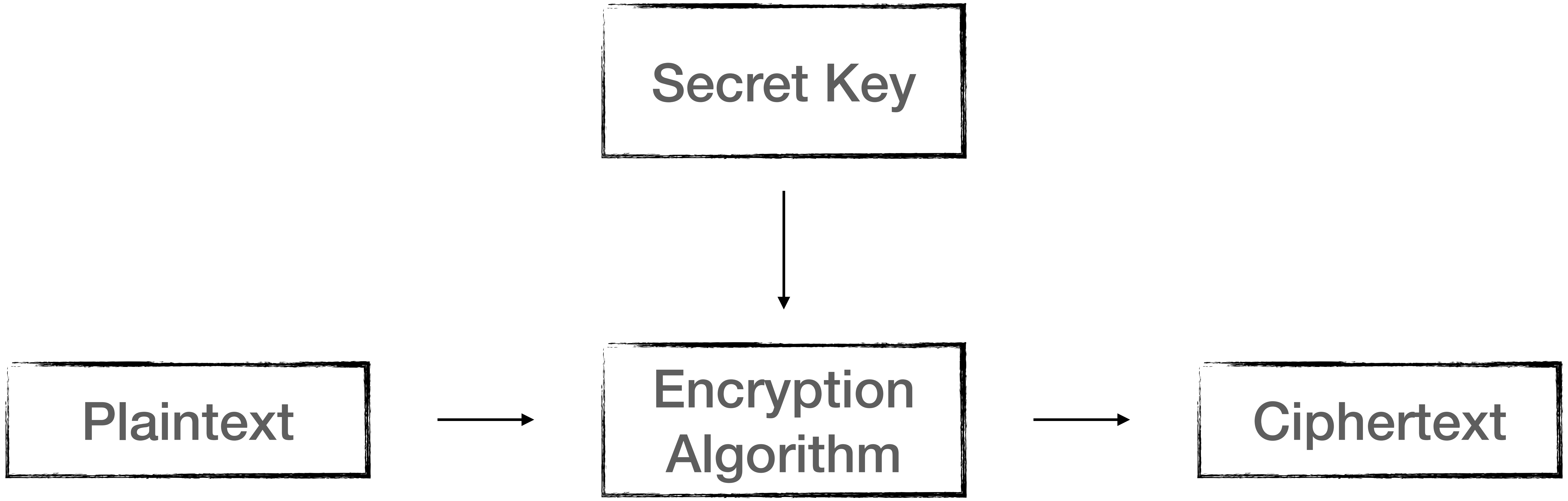
Towards bridging the gap between theory and practice

Abdel Taleb

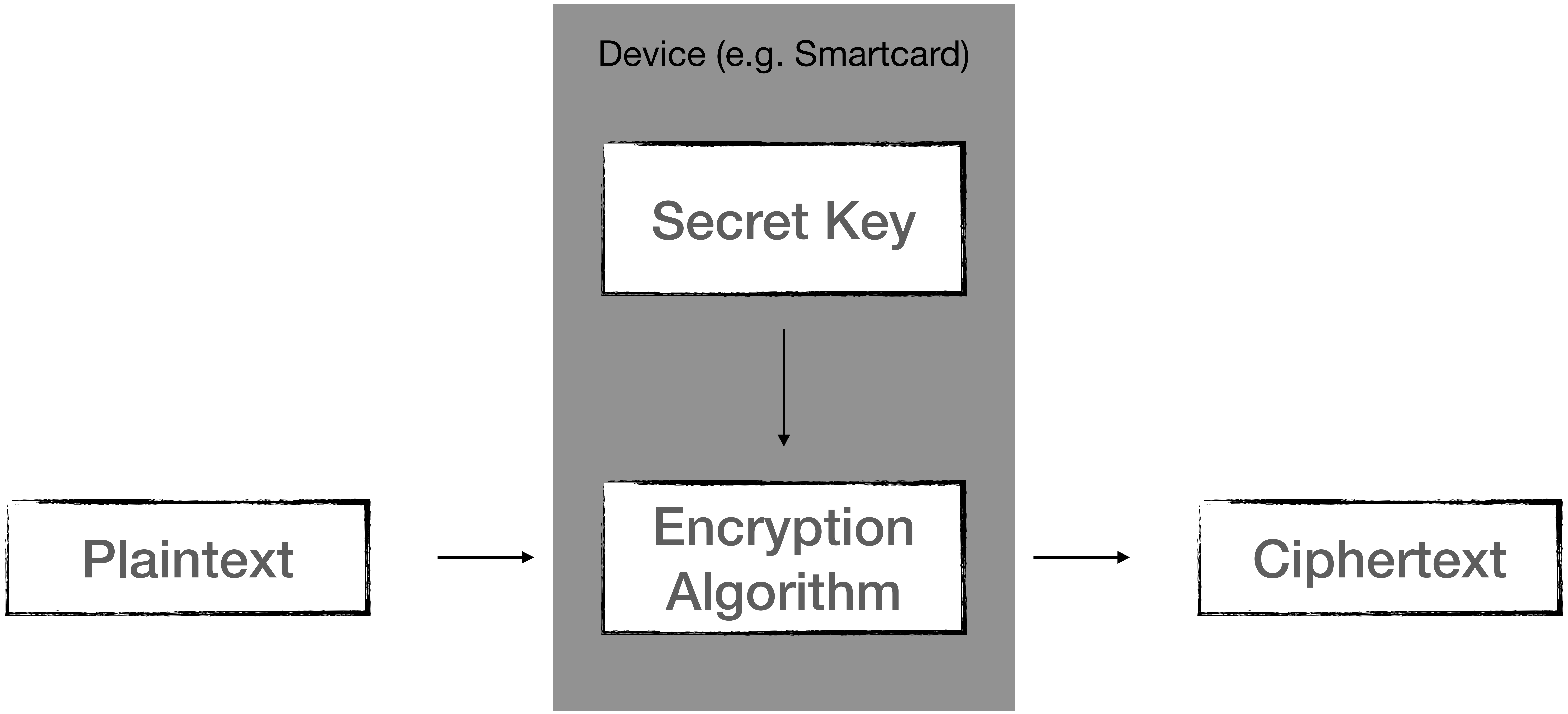
VeriSiCC Seminar

22 - 09 - 2022

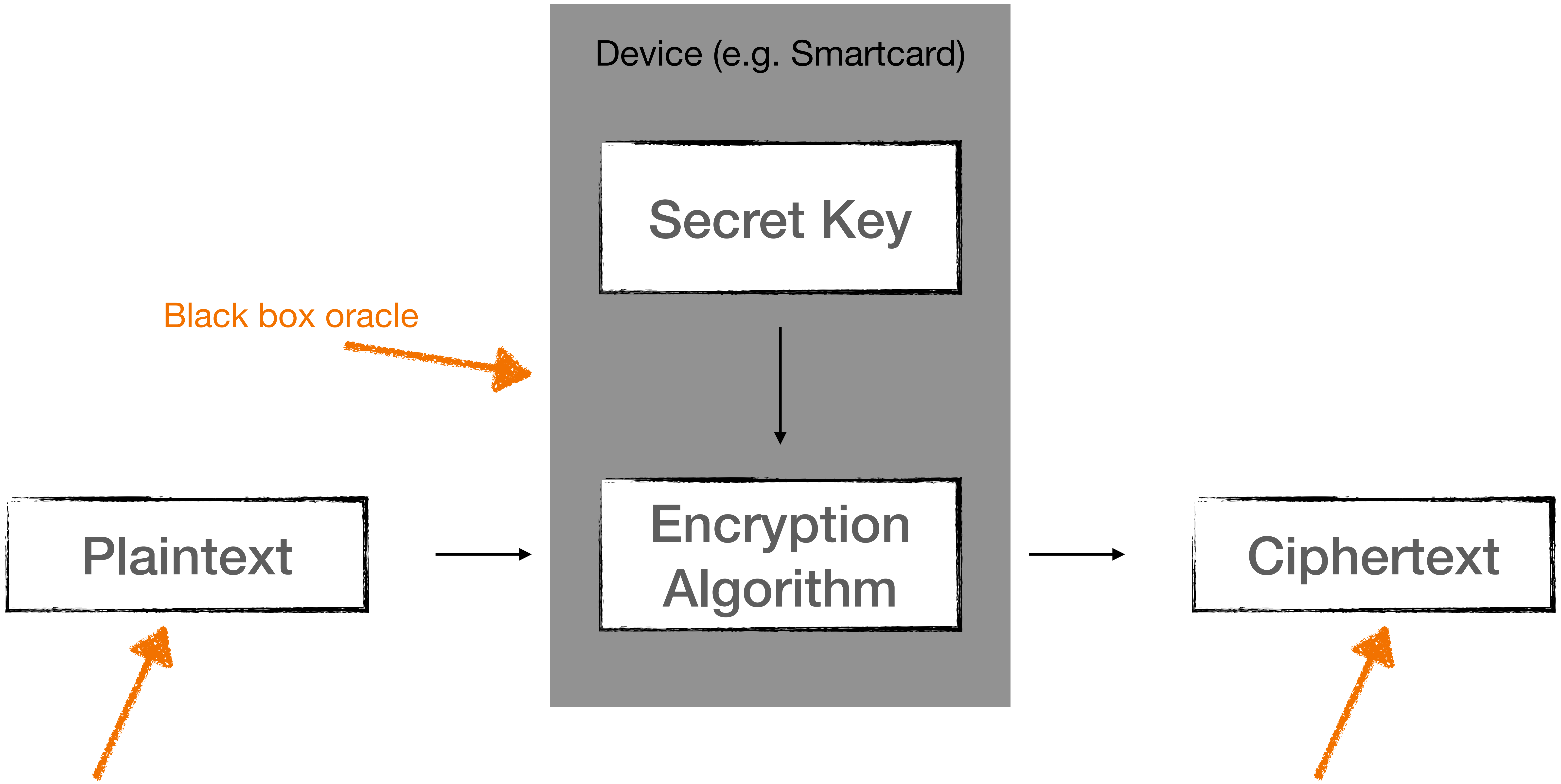
Side-Channel Attacks



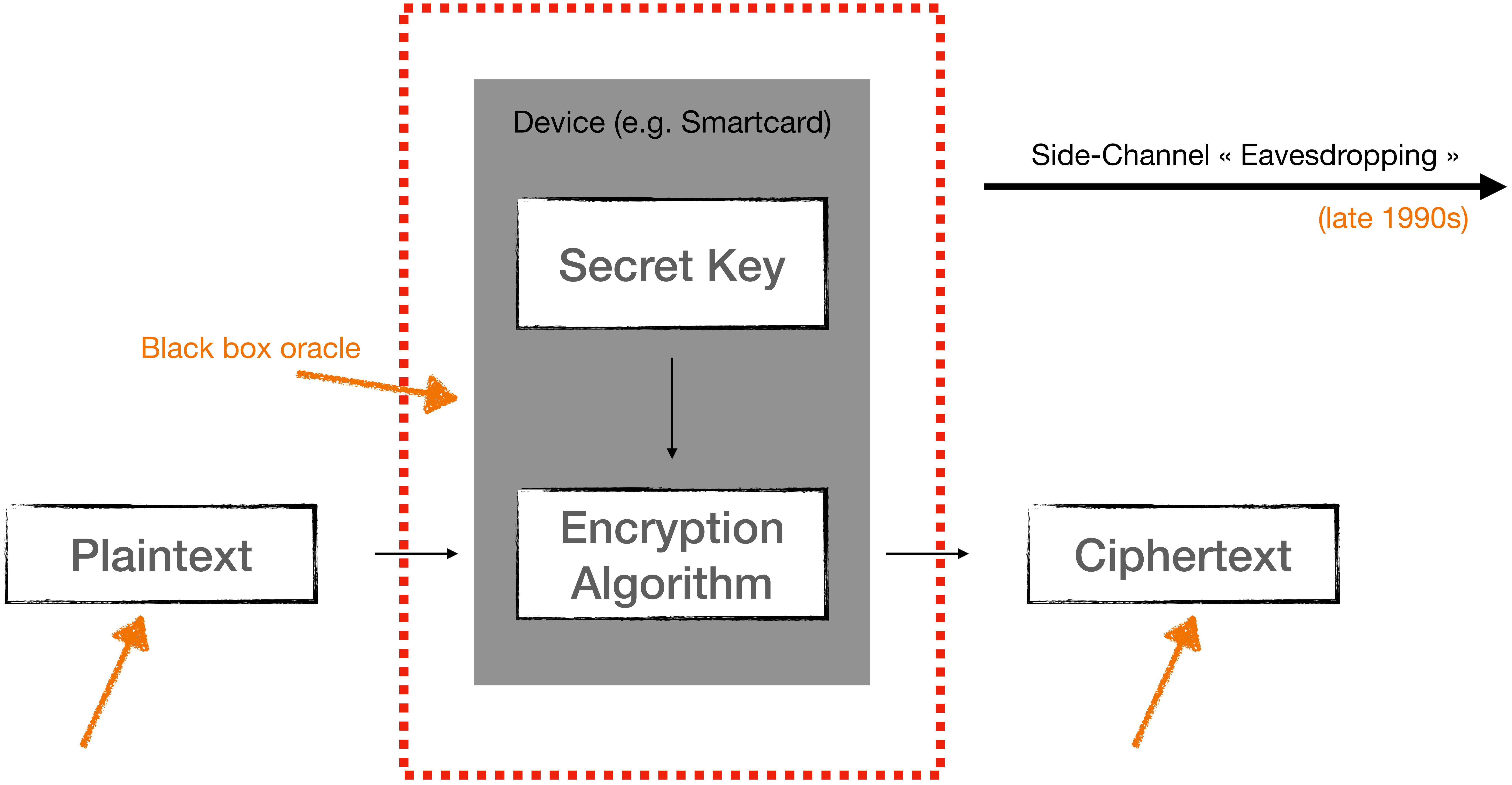
Side-Channel Attacks



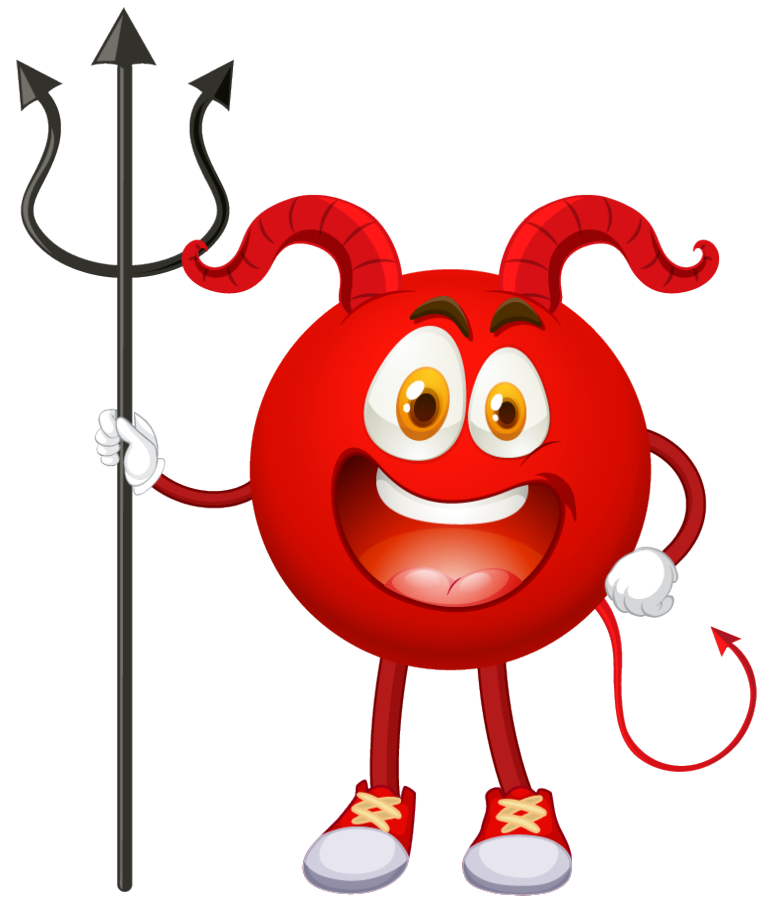
Side-Channel Attacks

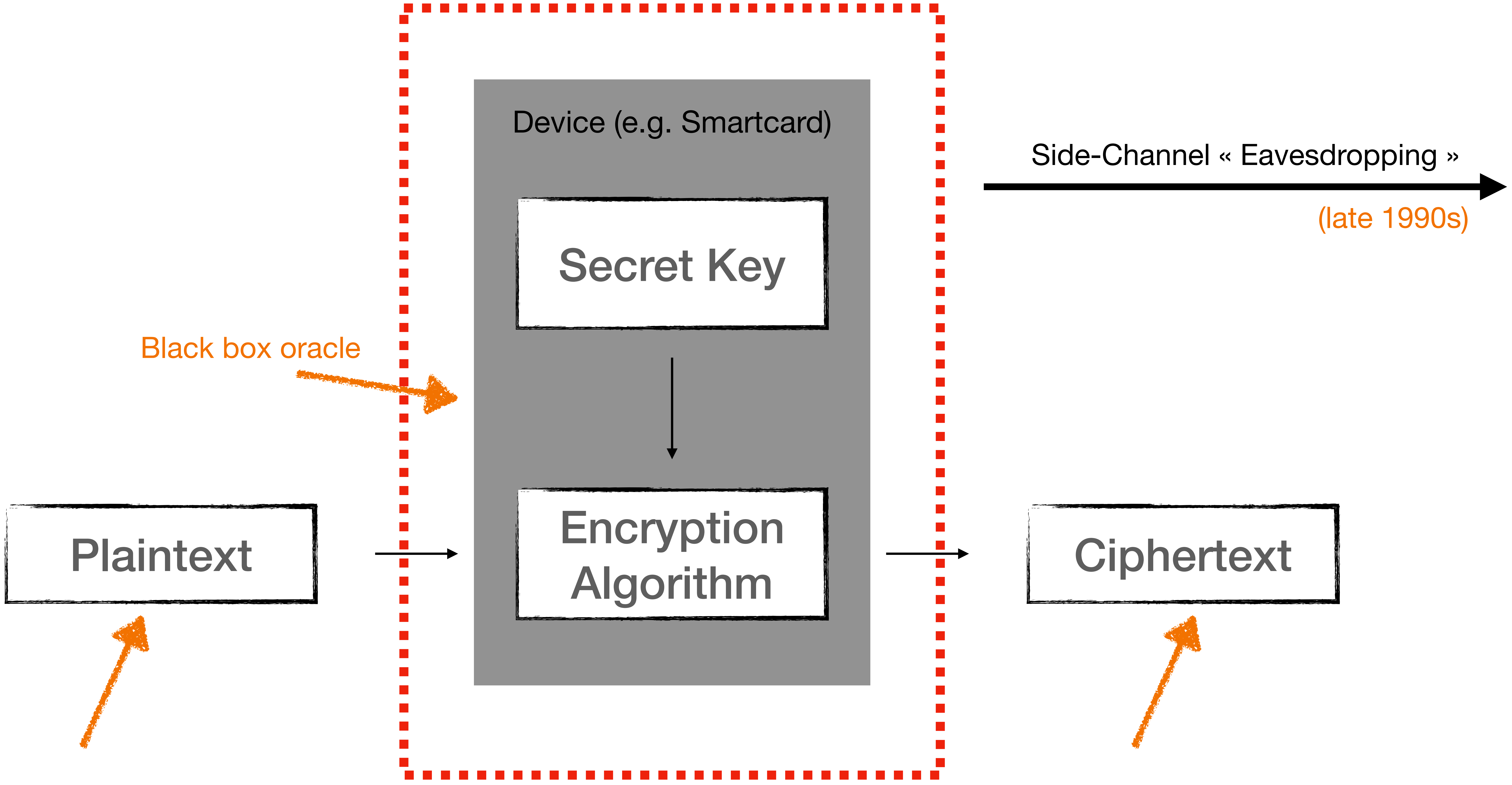


Side-Channel Attacks

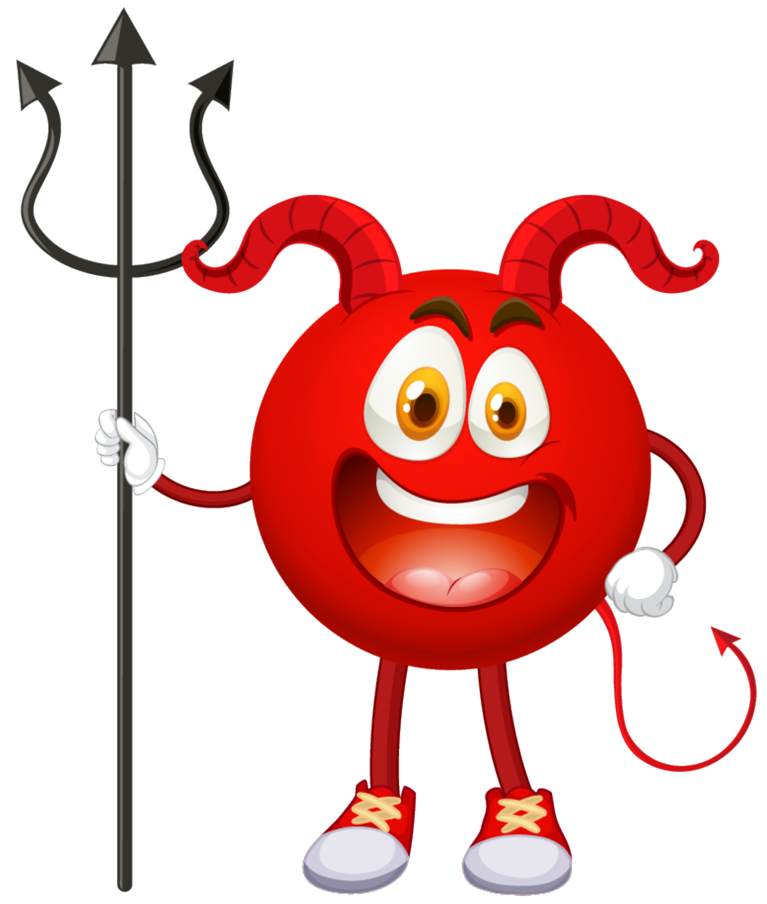


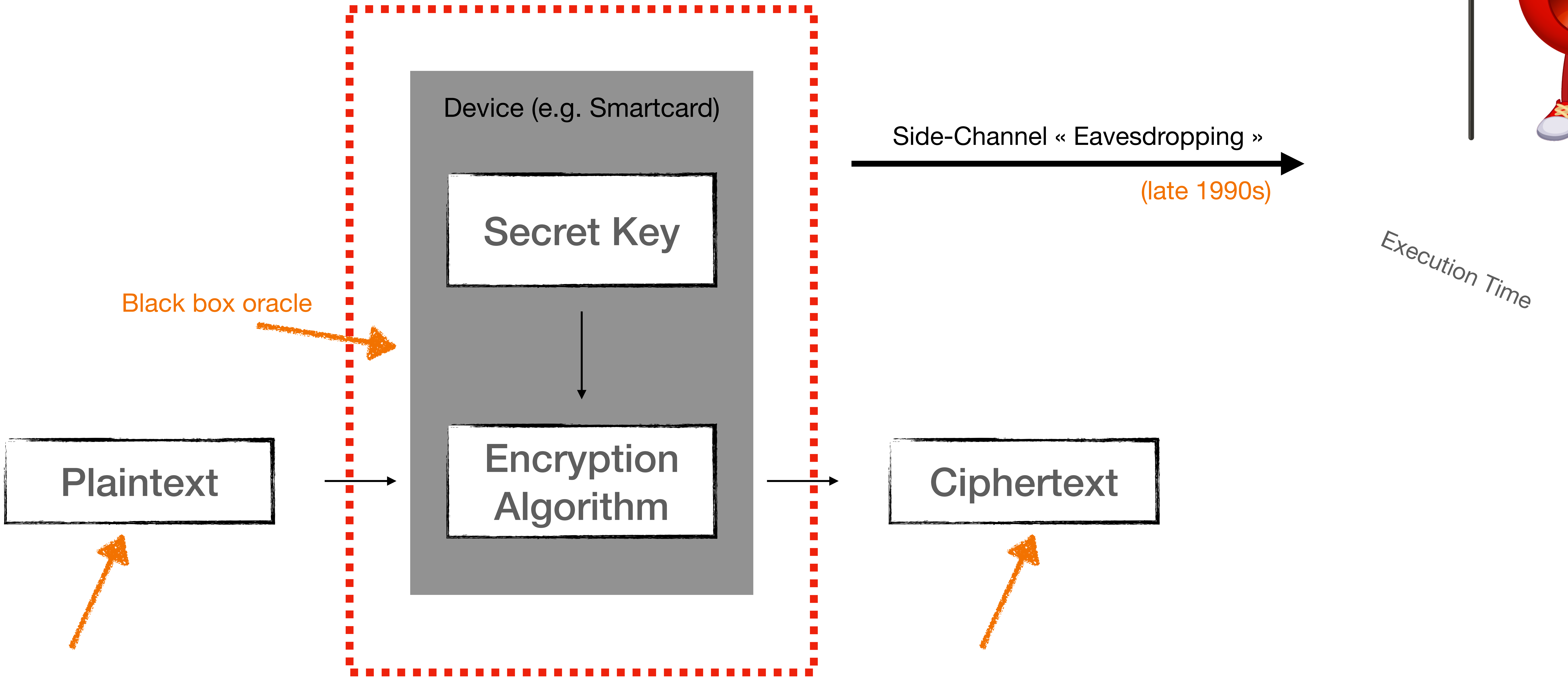
Side-Channel Attacks



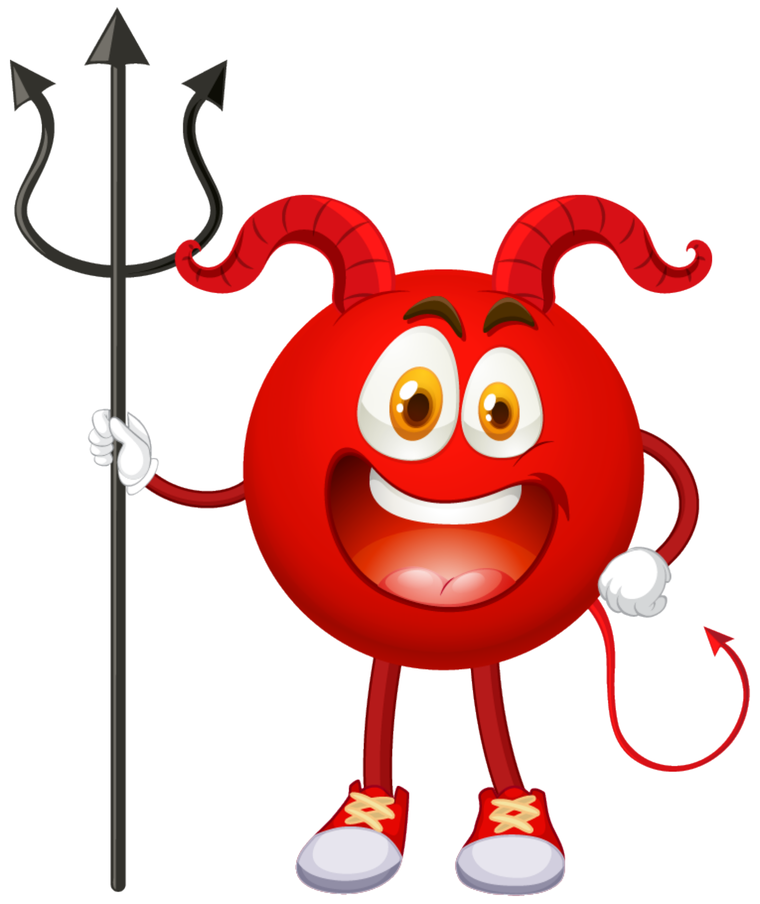


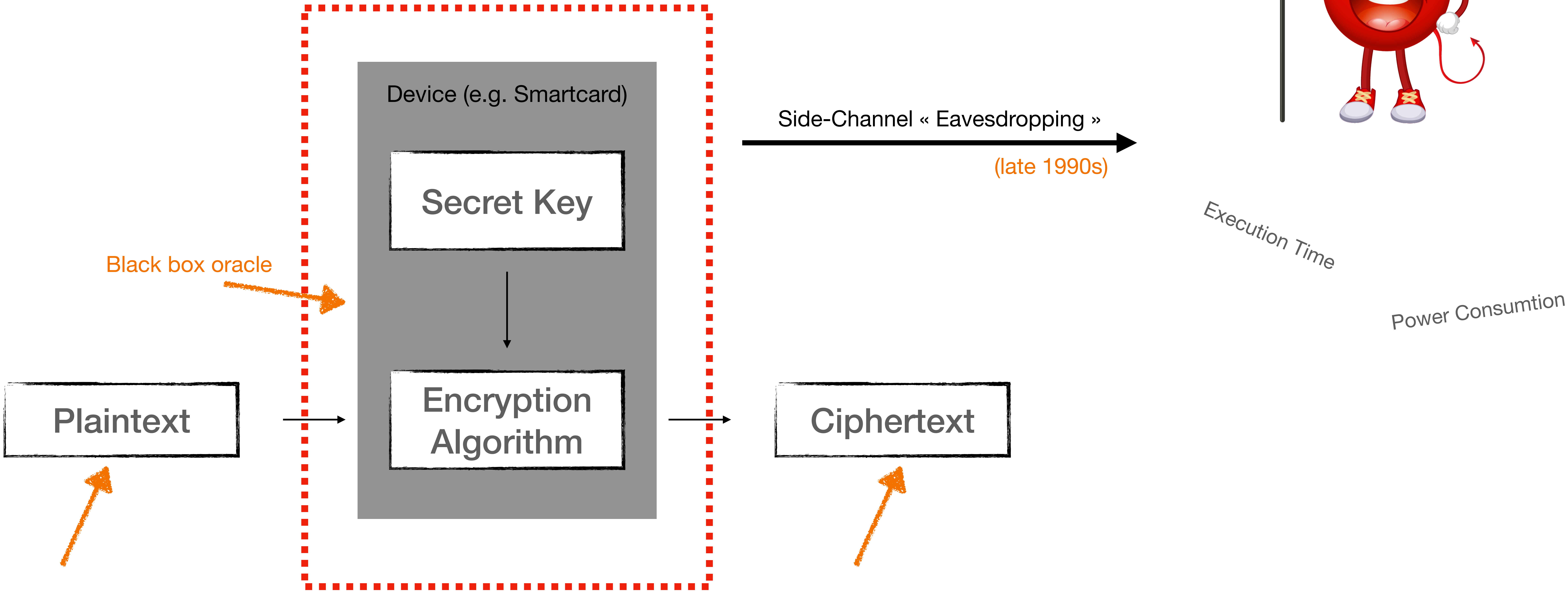
Side-Channel Attacks



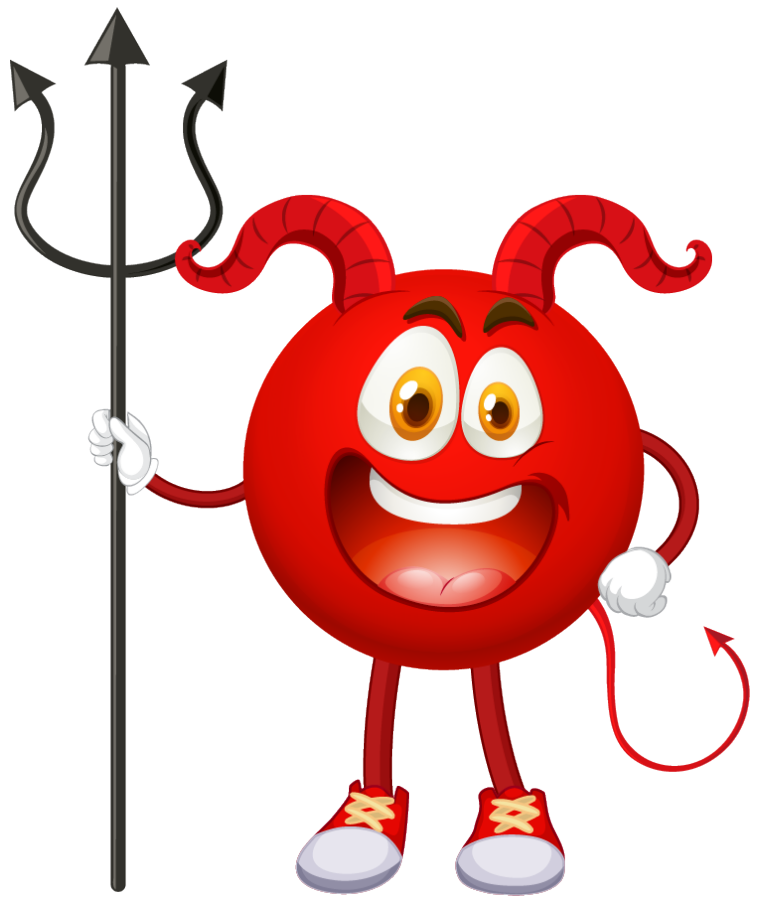


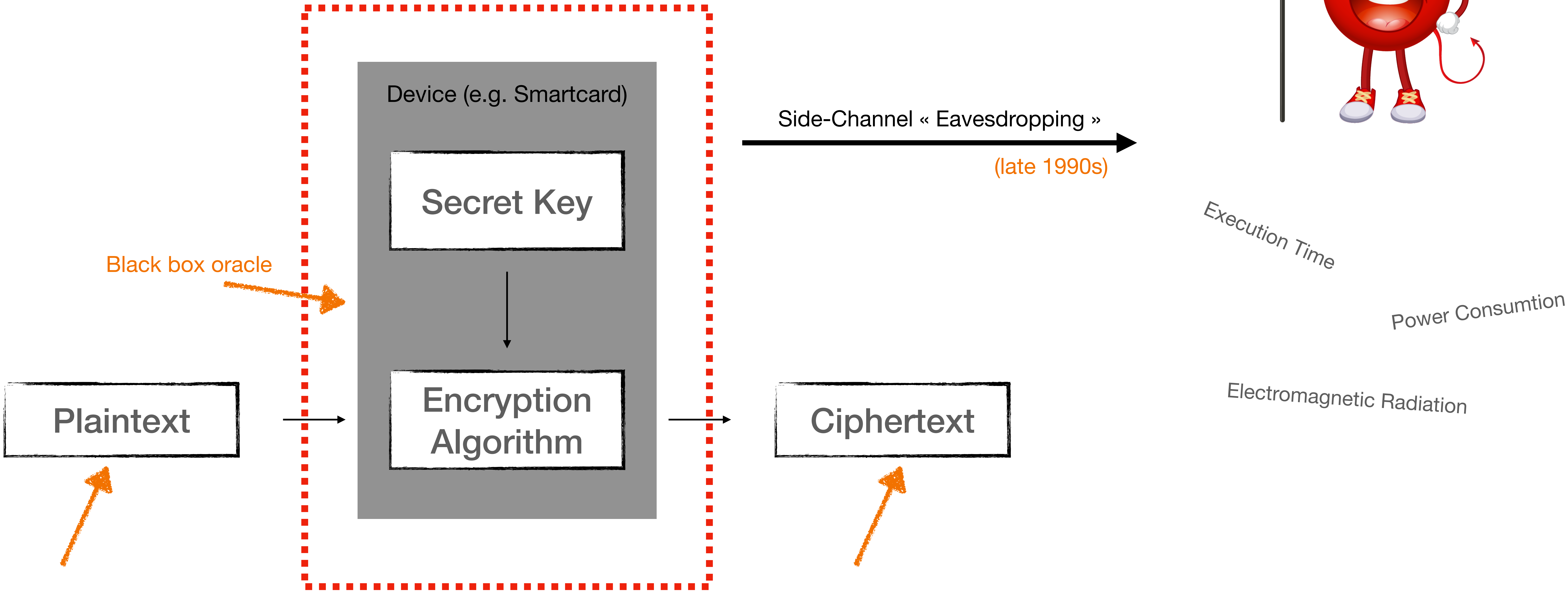
Side-Channel Attacks



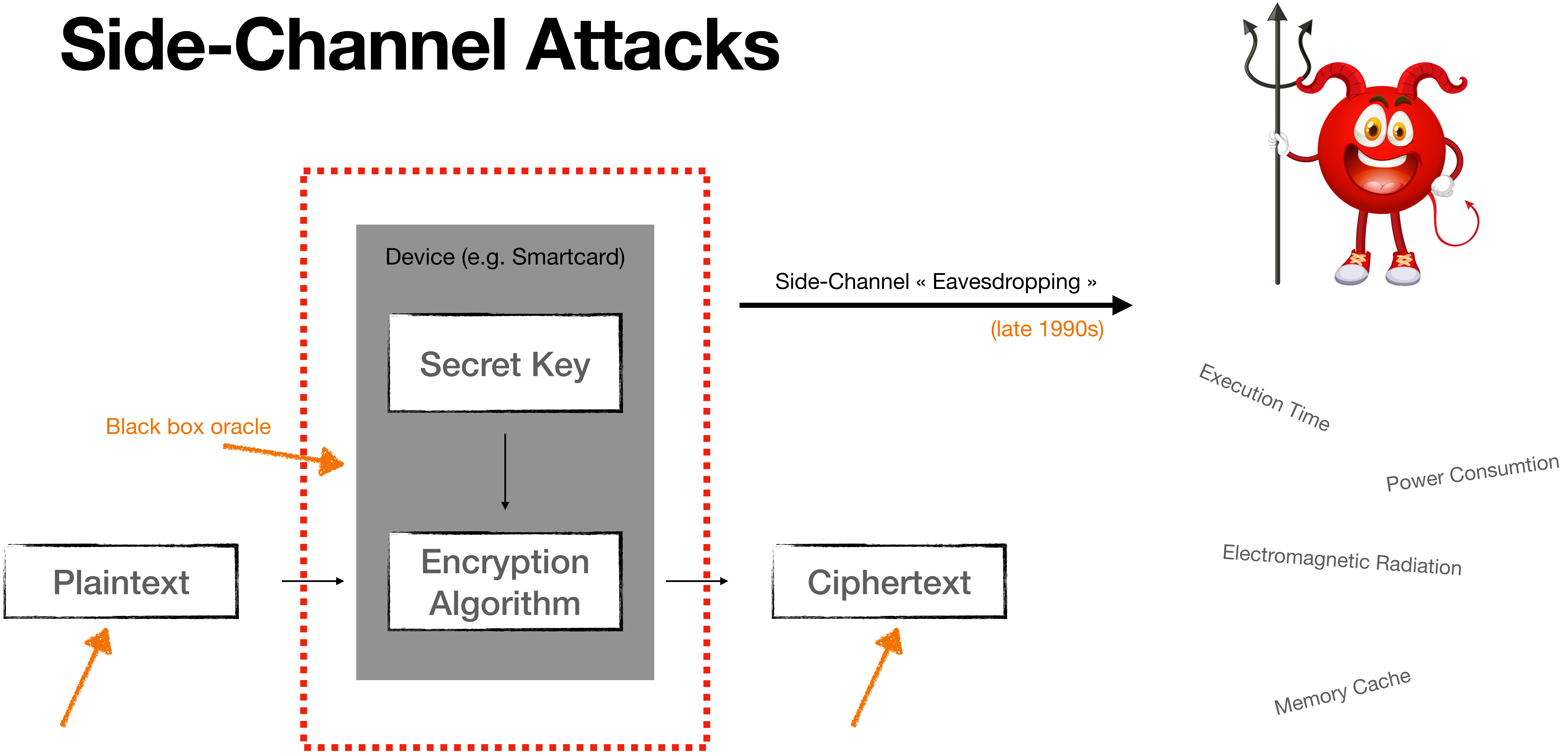


Side-Channel Attacks

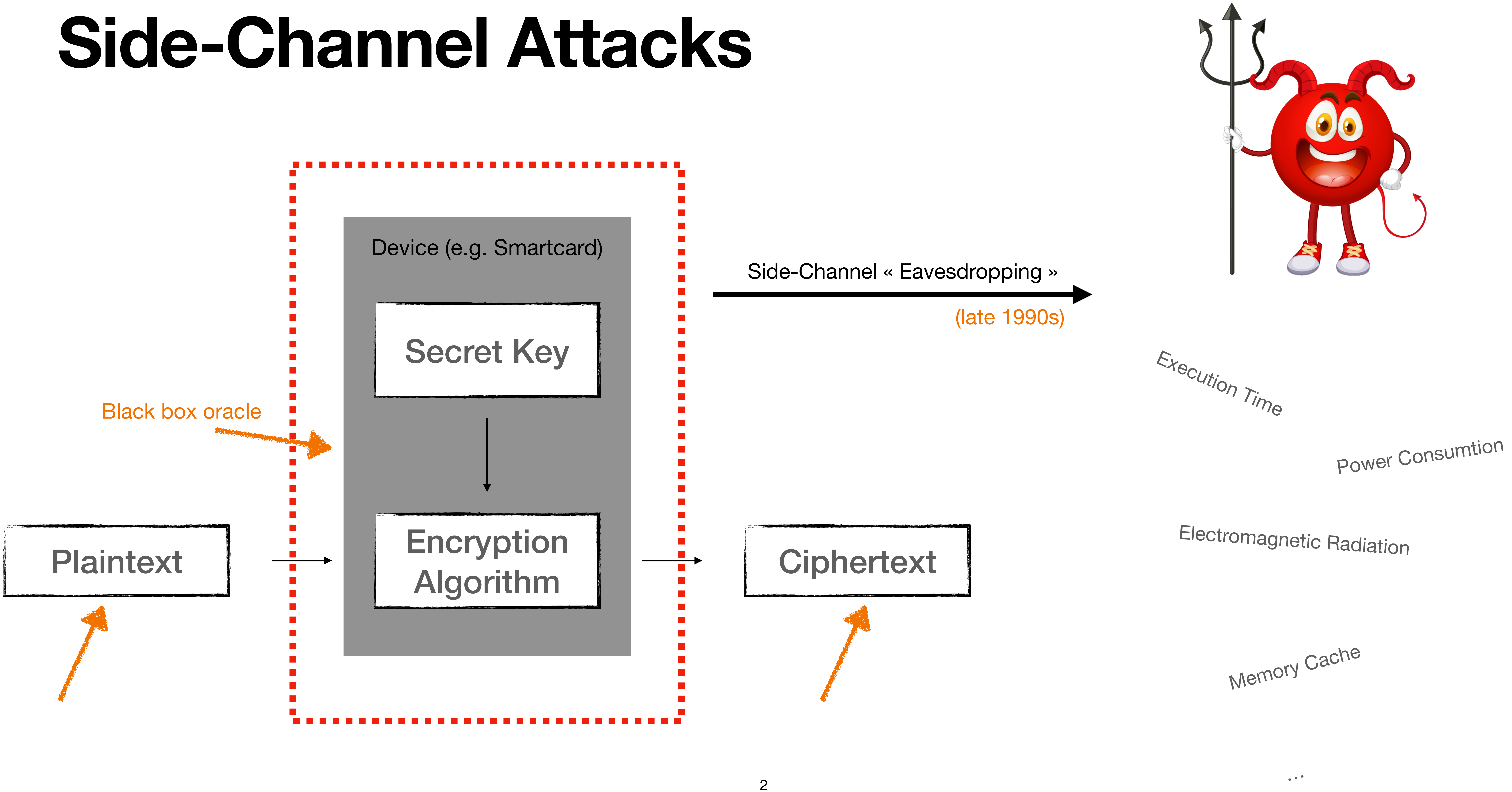




Side-Channel Attacks



Side-Channel Attacks




Device (e.g. Sma
Secret
Black box o
Plaintext
Encryp
Algorit
Power Consumtion
Radiation
ory Cache
...

Countermeasure

Masking *Chari et Al [CRYPTO’99], Goubin and Patarin [CHES’99]*

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \xleftarrow{\$} \mathbb{F}_2$$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$$

$$\ldots$$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \stackrel{\$}{\leftarrow} \mathbb{F}_2$$

$$\ldots$$

$$x_{n-1} \stackrel{\$}{\leftarrow} \mathbb{F}_2$$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$$

$$\ldots$$

$$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$$

$$x_n \leftarrow x - x_1 \ldots - x_{n-1}$$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n - 1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b

a

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b

a

b

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b

a

b

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

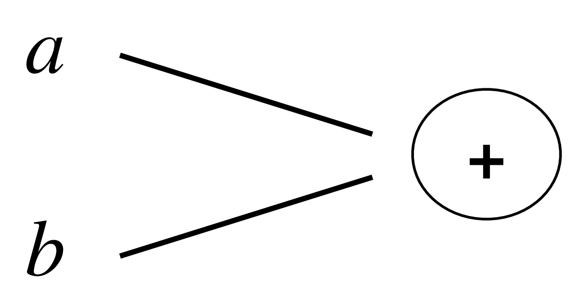
$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

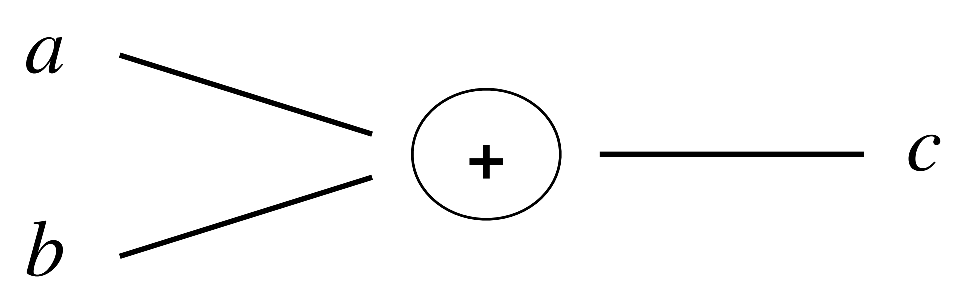
$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \stackrel{\$}{\leftarrow} \mathbb{F}_2$

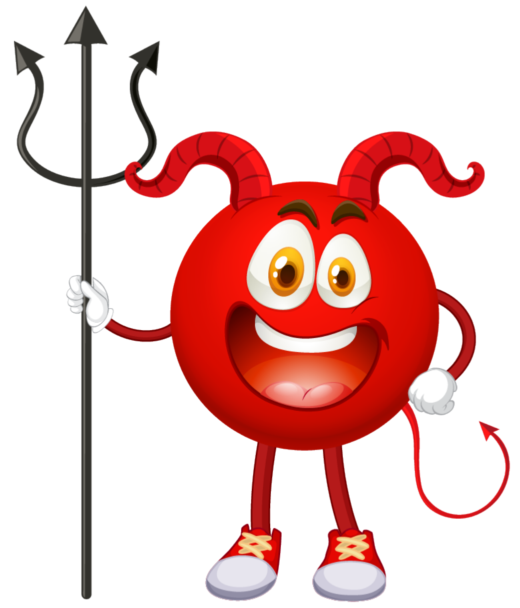
$\ldots$

$x_{n-1} \stackrel{\$}{\leftarrow} \mathbb{F}_2$

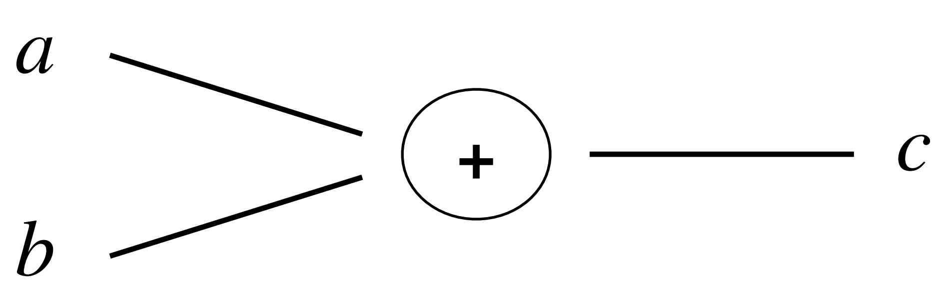
$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

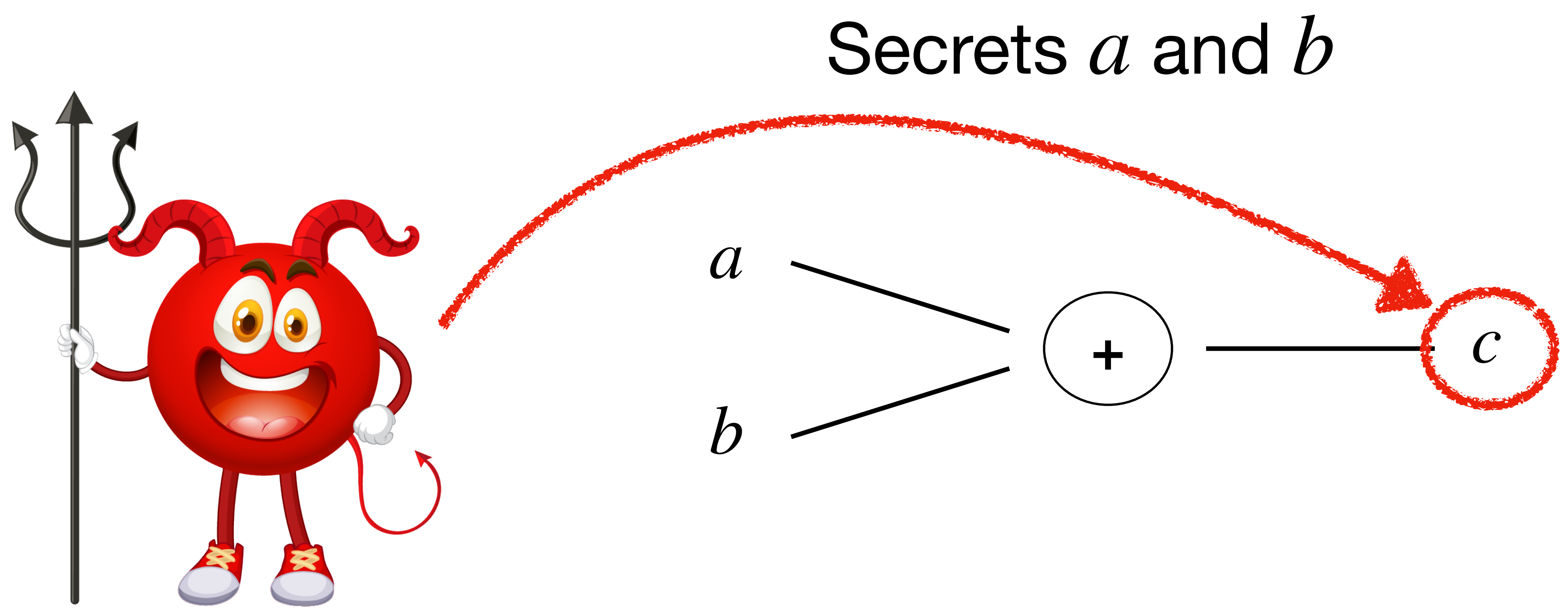
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

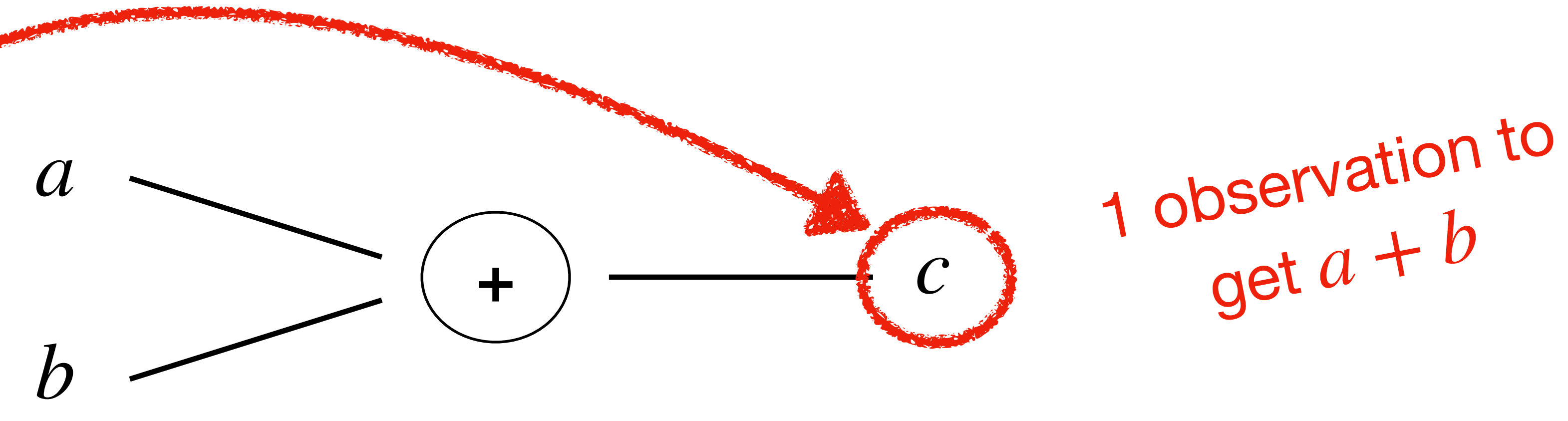
$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

↓ Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

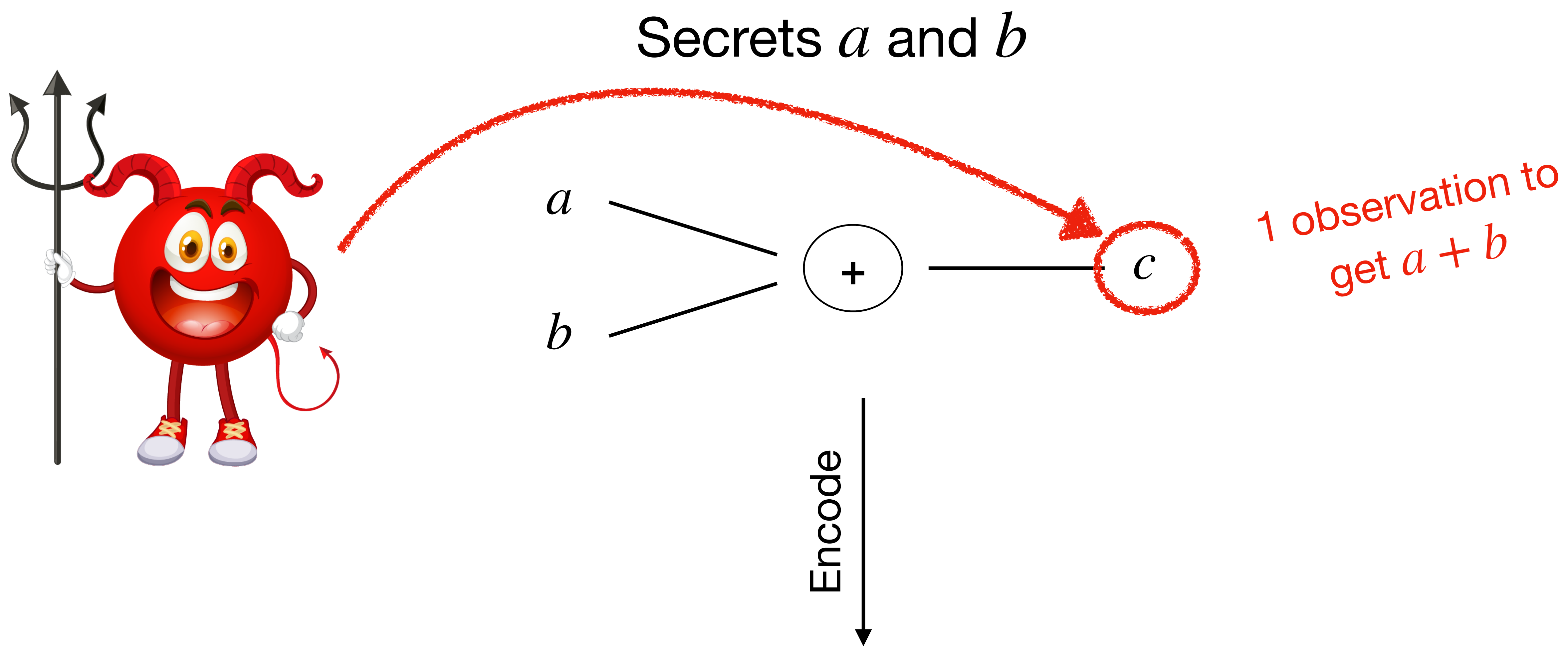
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

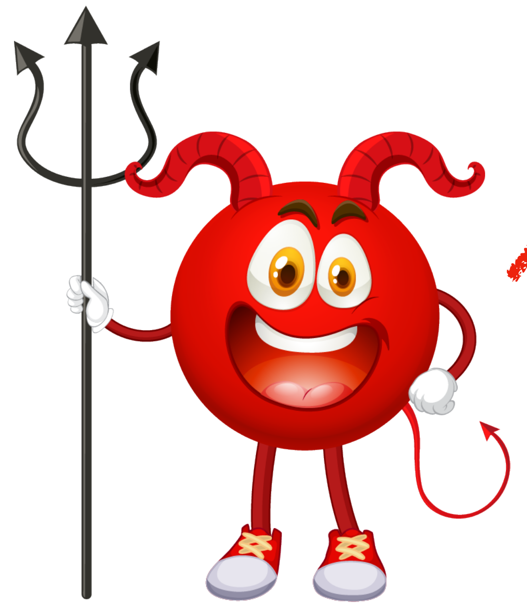
$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Secrets a and b

a, b → + → c

1 observation to get $a + b$

Encode ↓

a_1

b_1

$\ldots$

a_n

b_n

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

$\ldots$

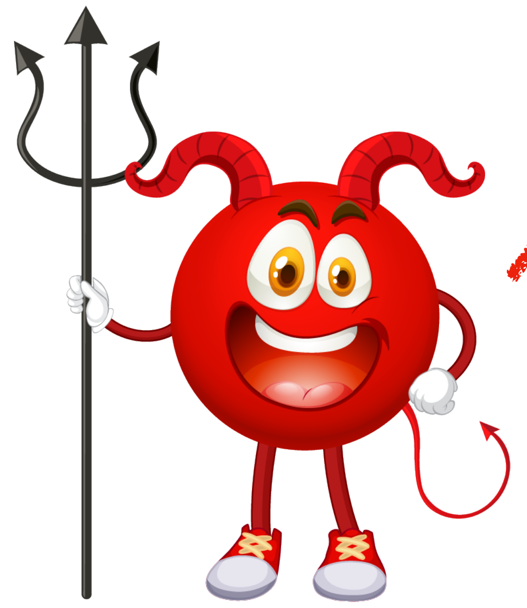
$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

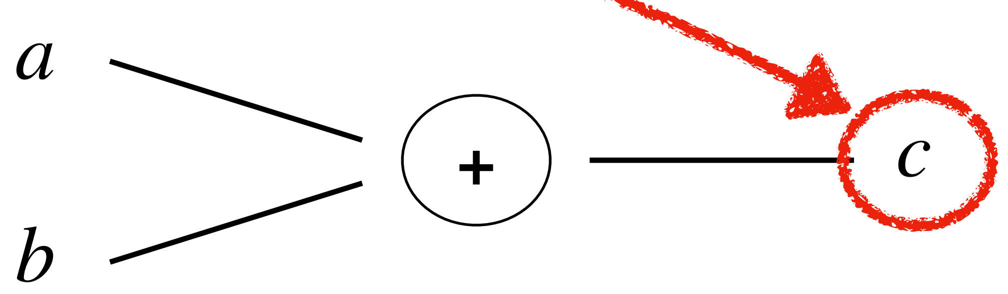
$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b





1 observation to get $a+b$

Encode ↓

a_1

c_1

b_1

$\ldots$

a_n

c_n

b_n

Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

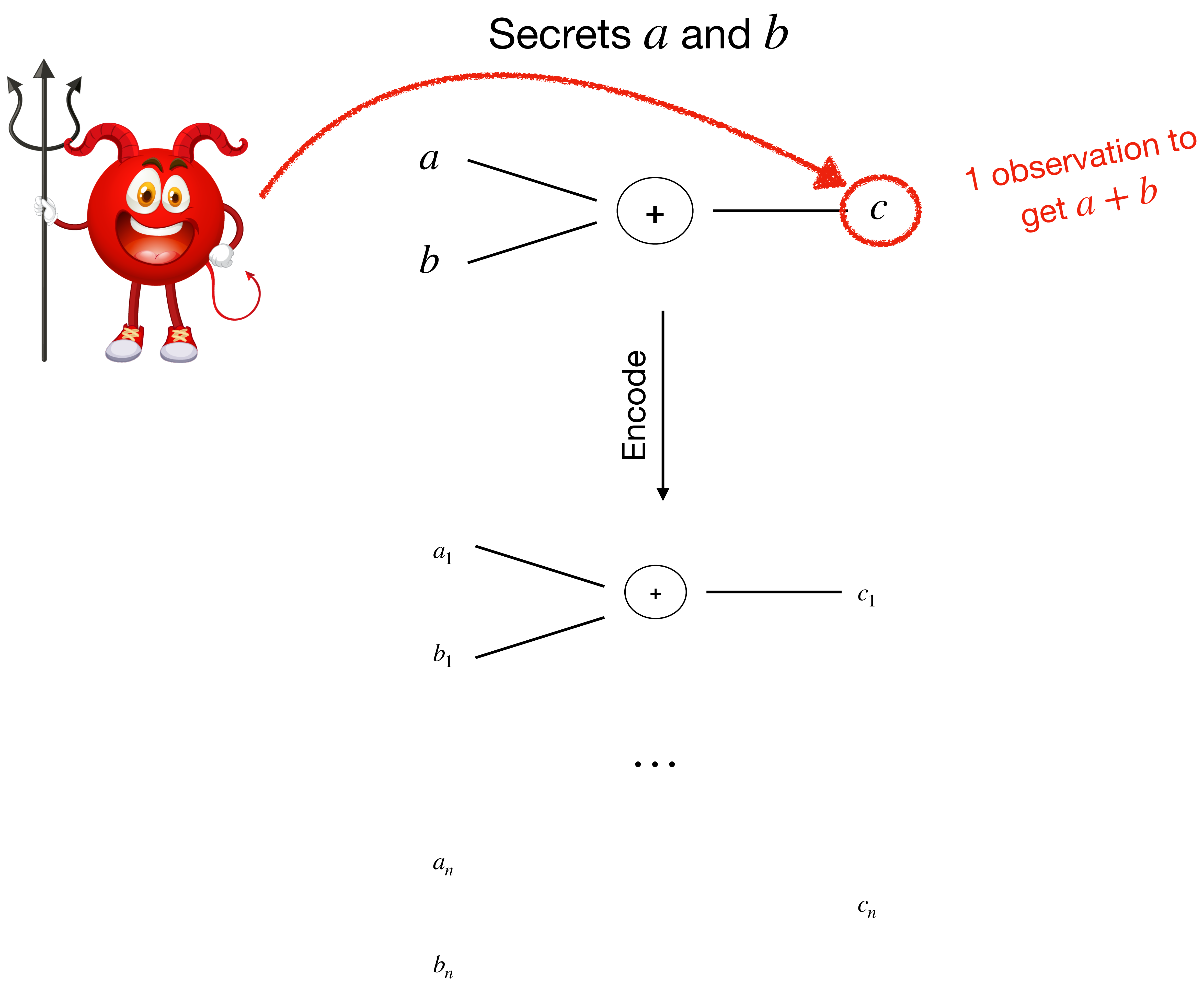
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n - 1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

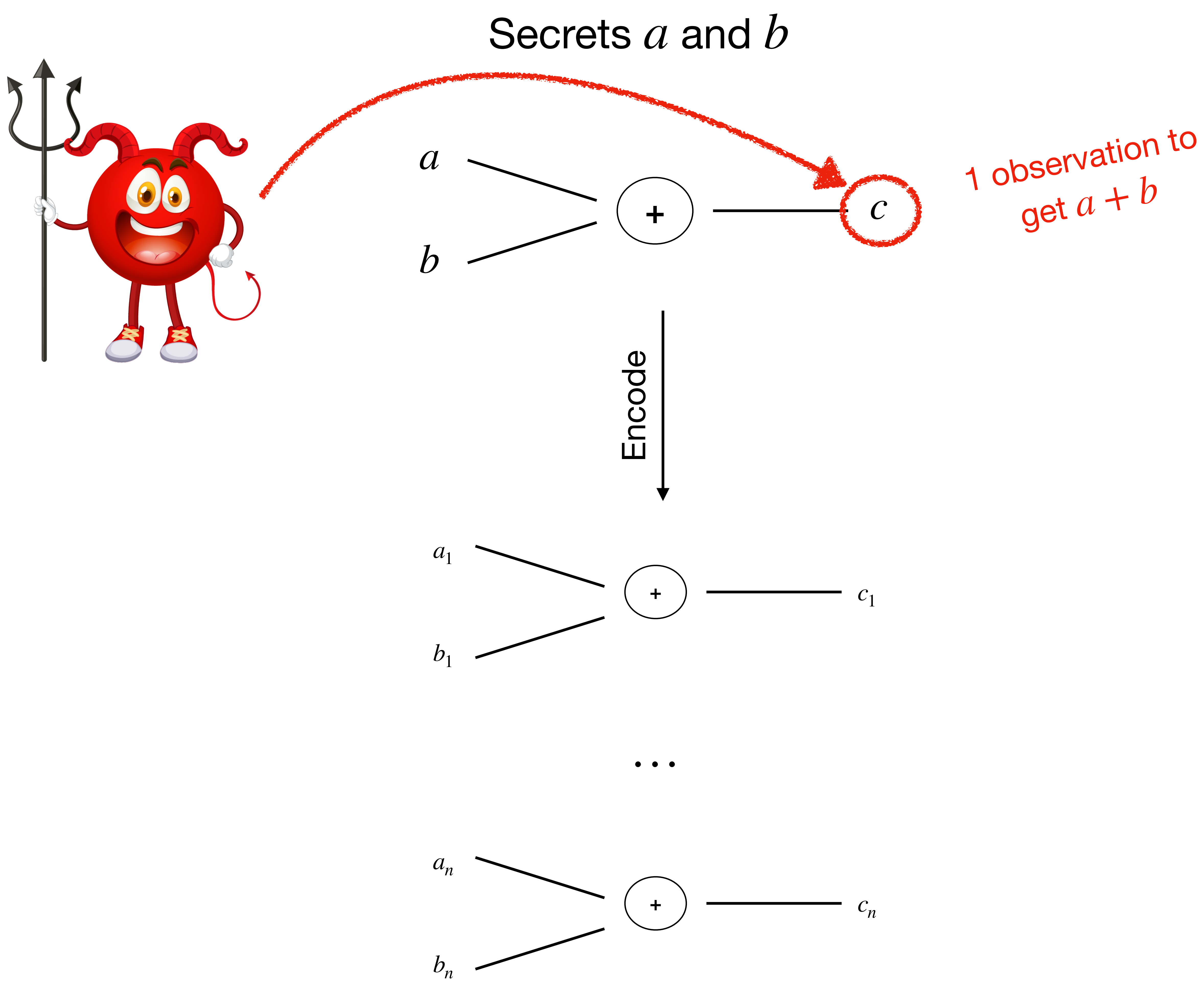
$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

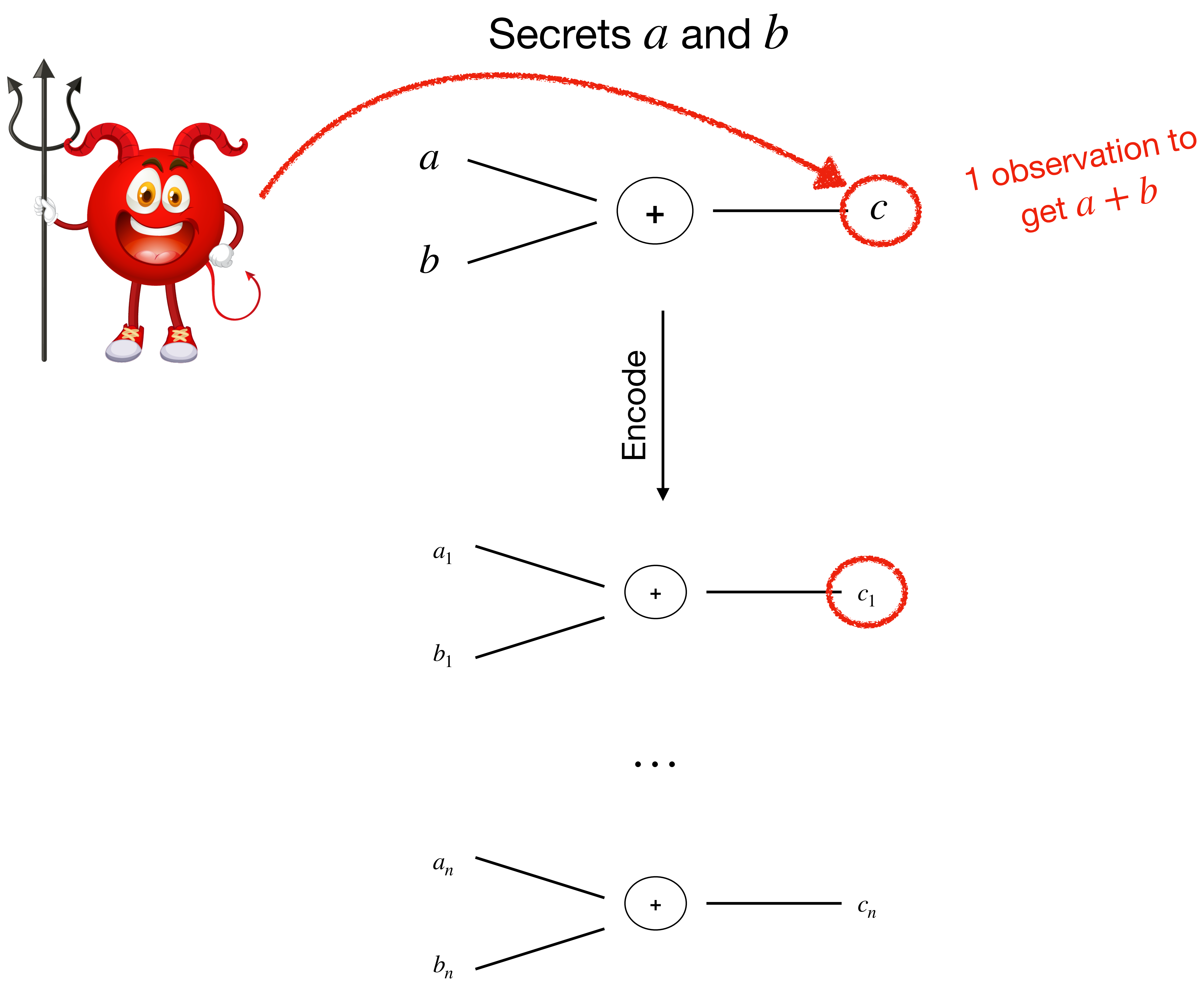
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

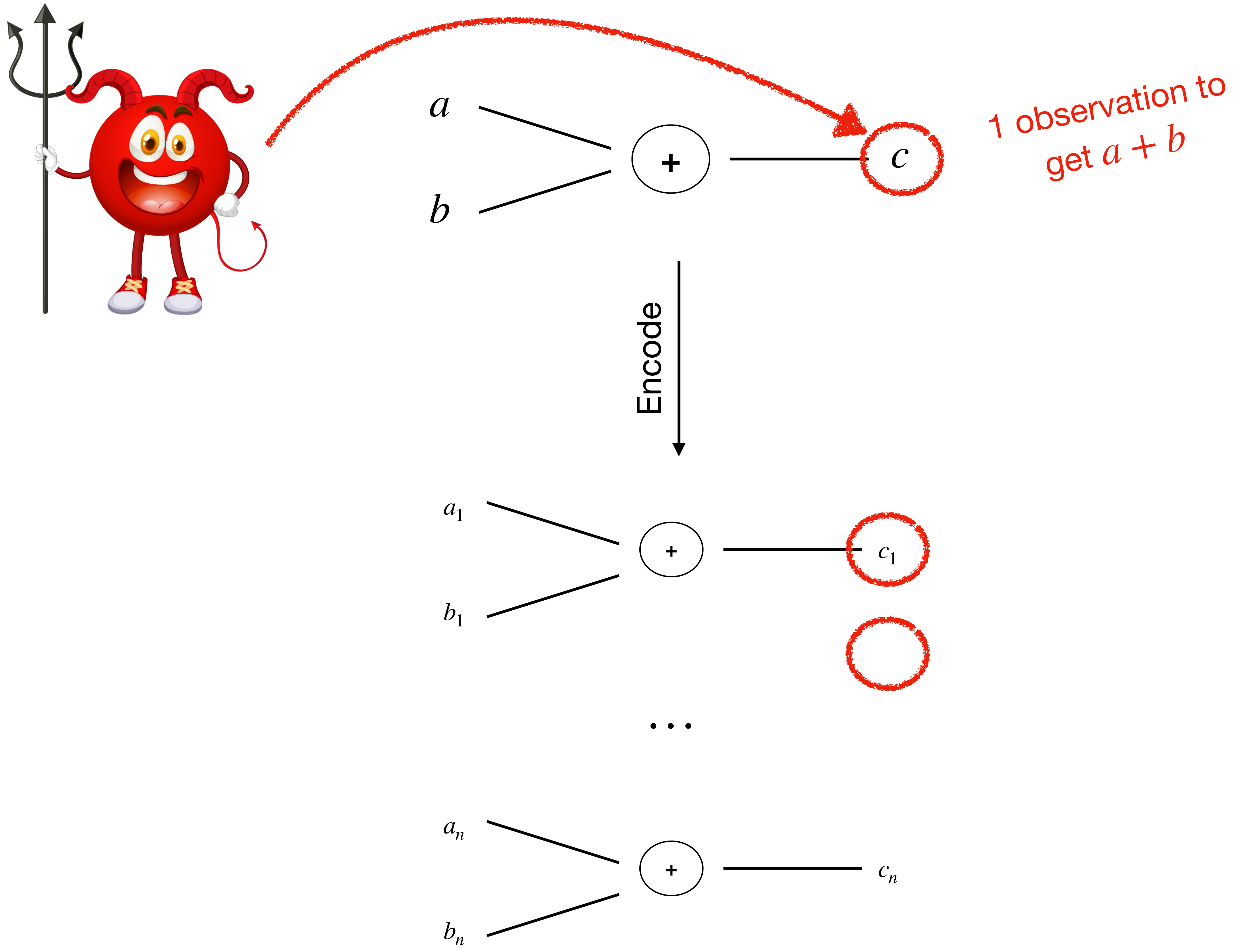
$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n - 1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$$

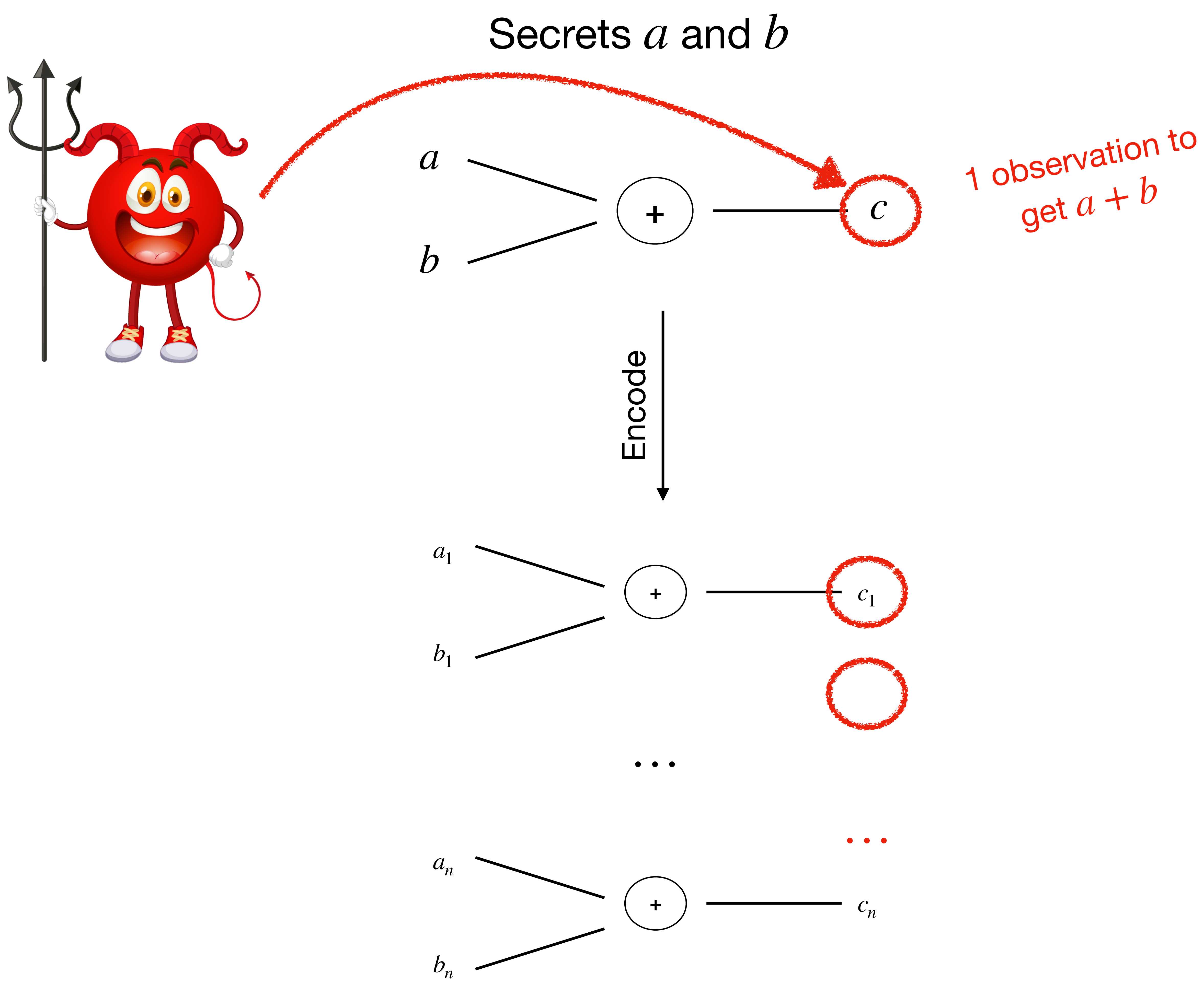
$$\ldots$$

$$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$$

$n-1$ random values

secret recombination

$$x_n \leftarrow x - x_1 \ldots - x_{n-1}$$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

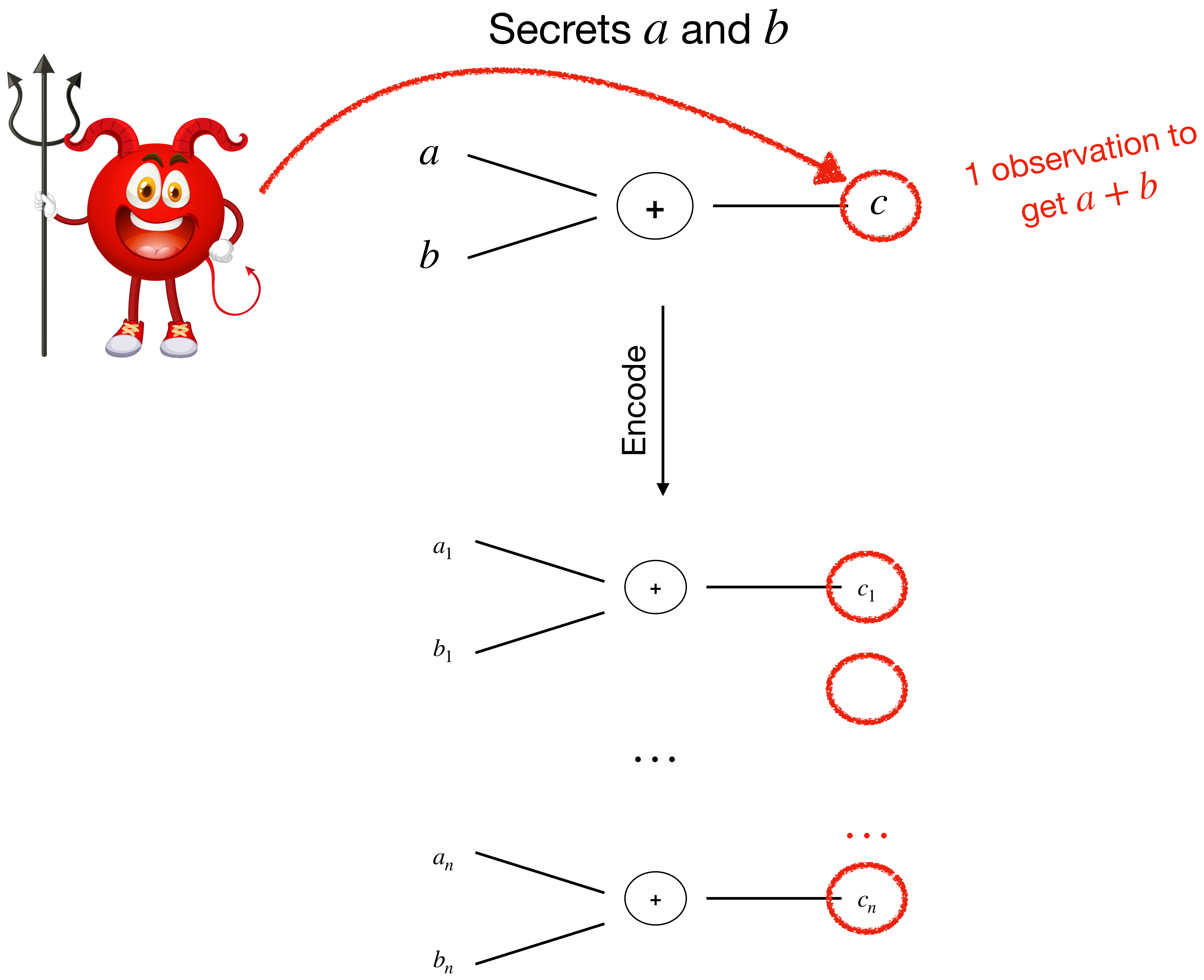
$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

↓ Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

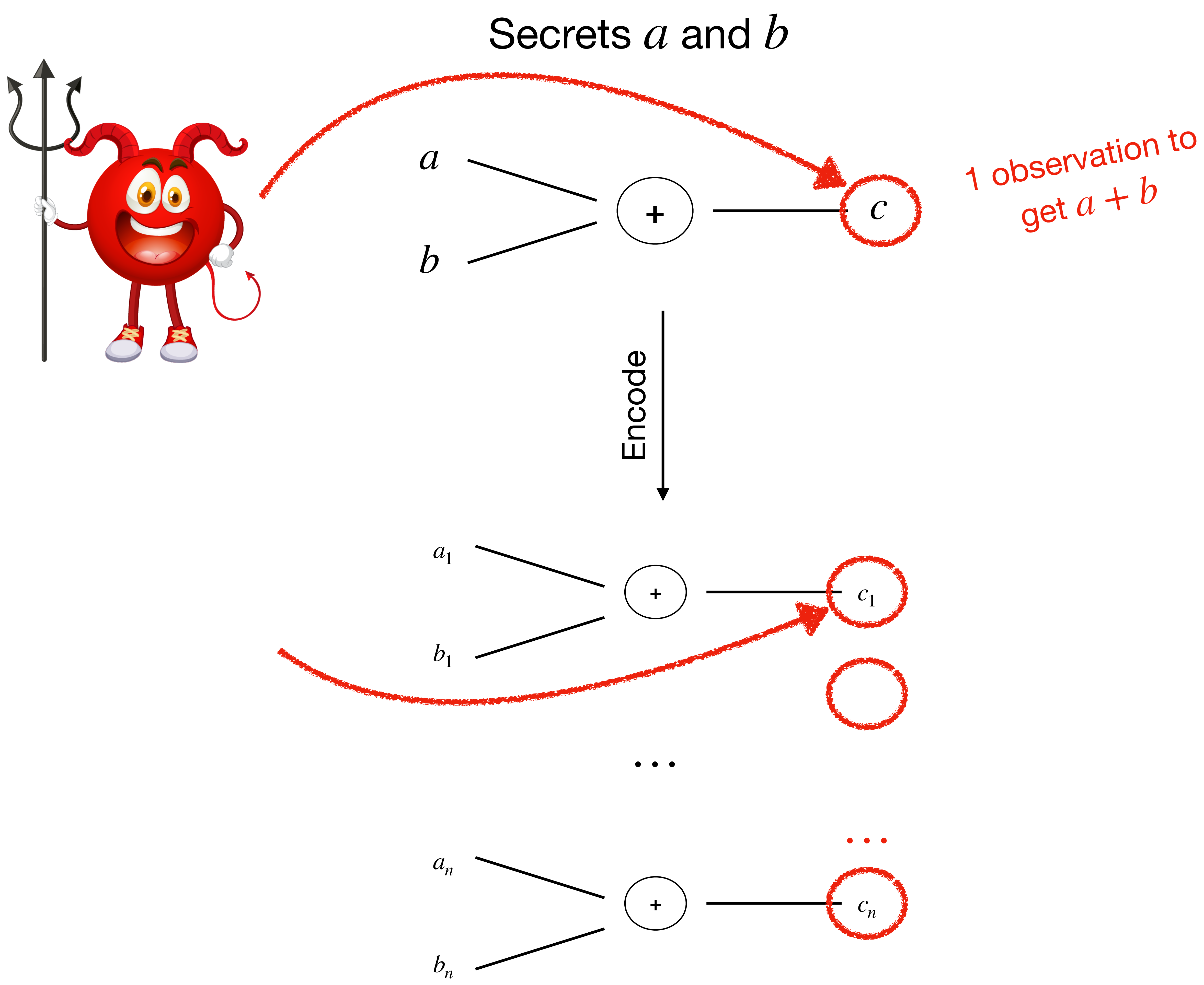
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

↓ Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

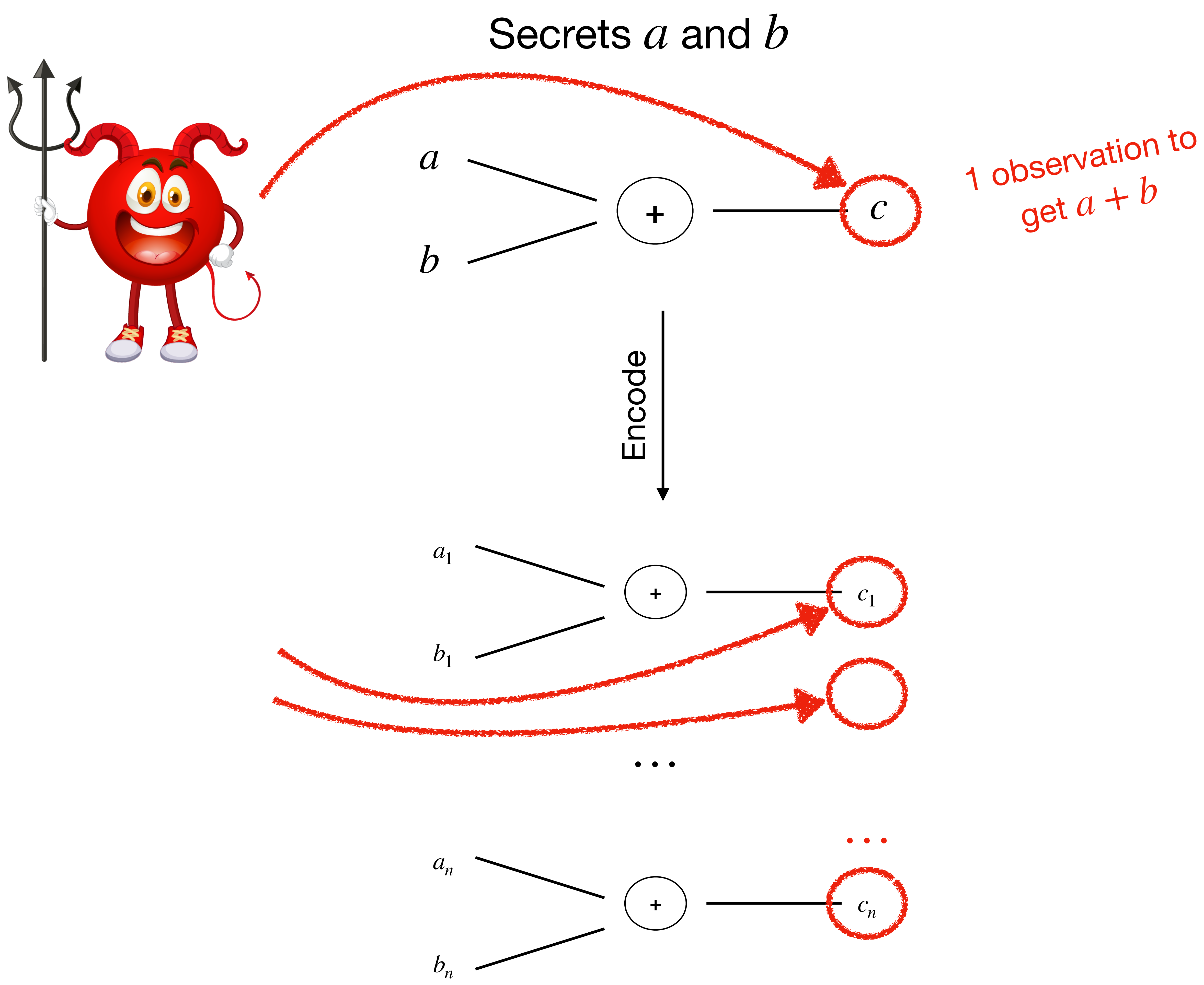
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

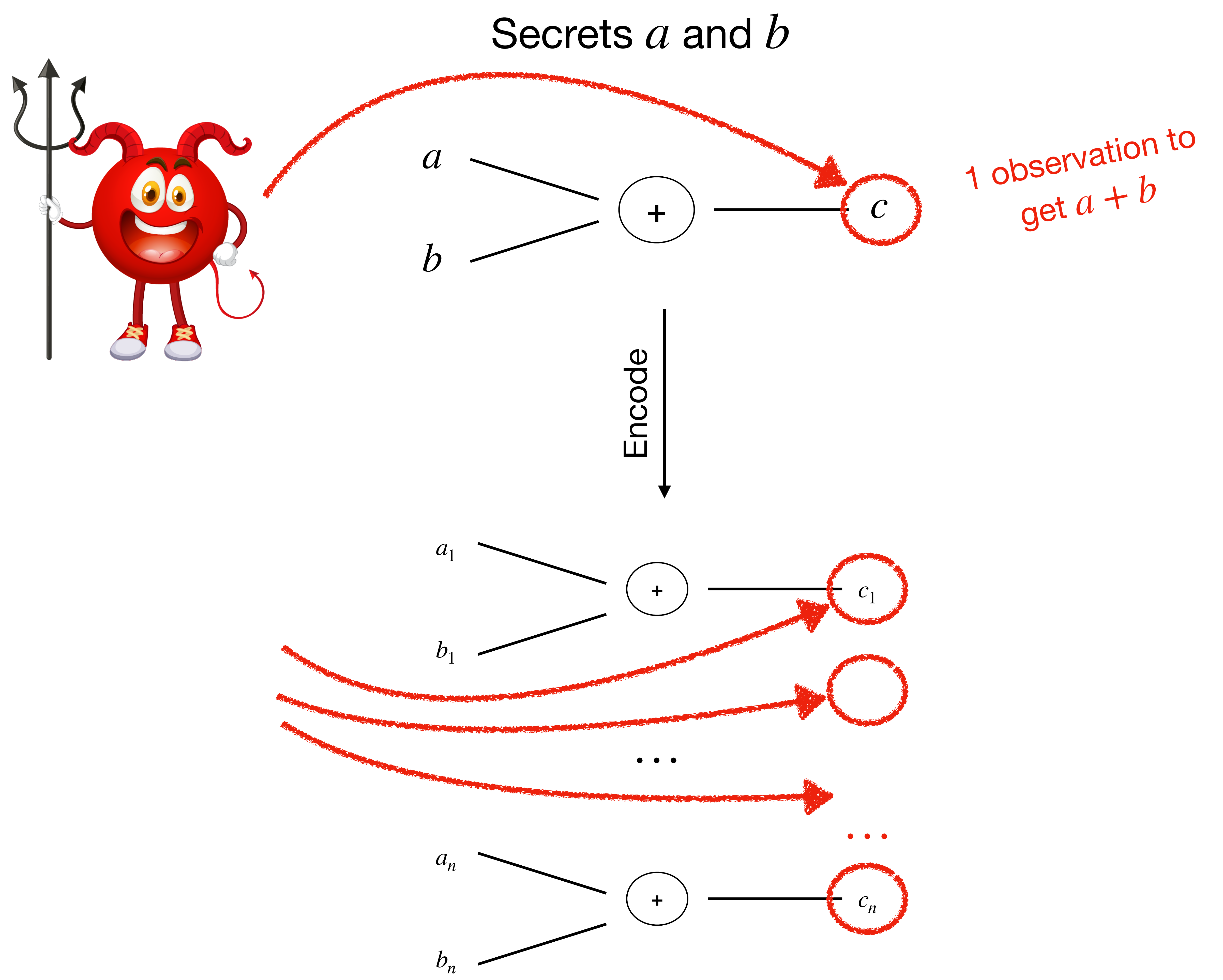
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

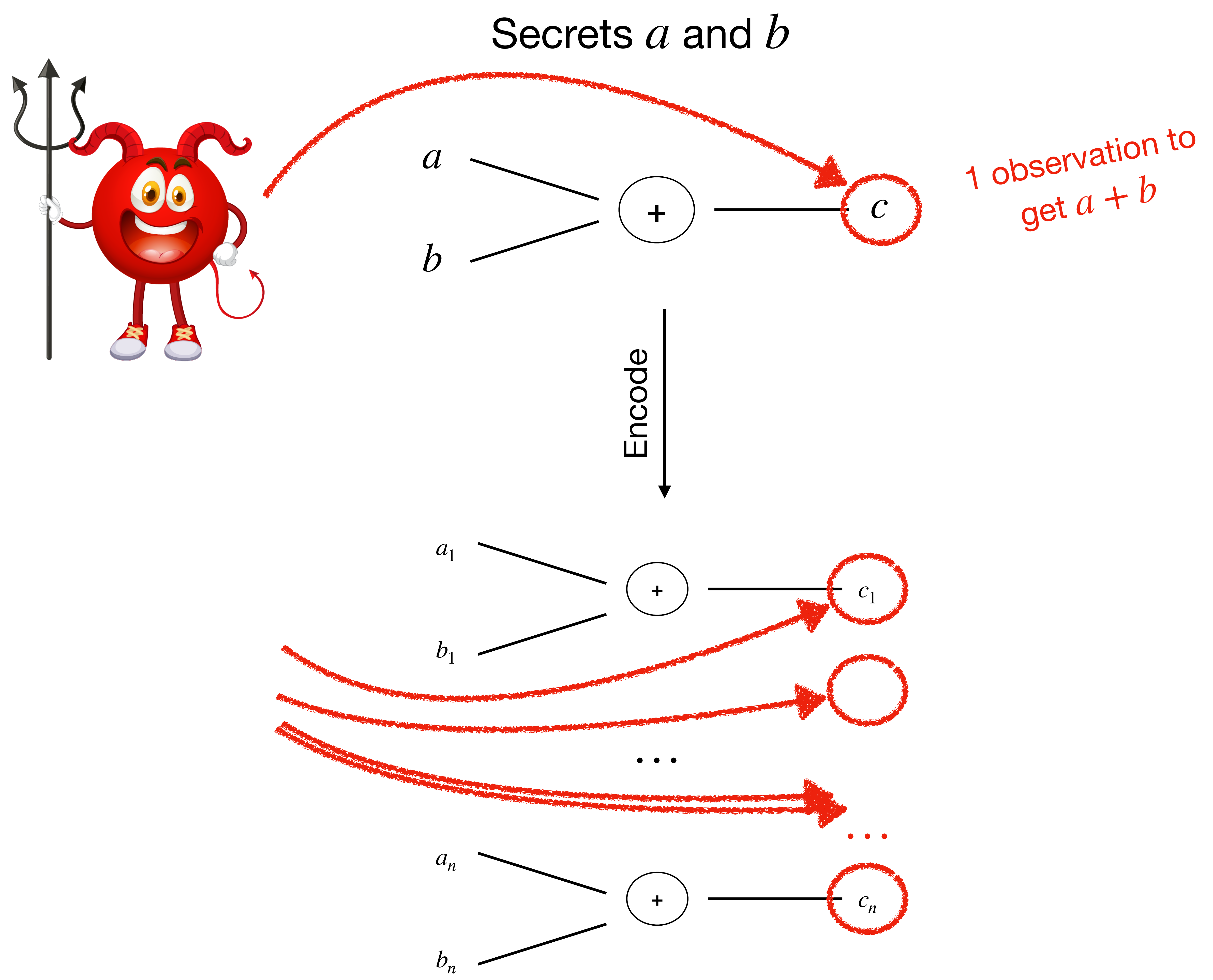
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$$x_1 \xleftarrow{\$} \mathbb{F}_2$$

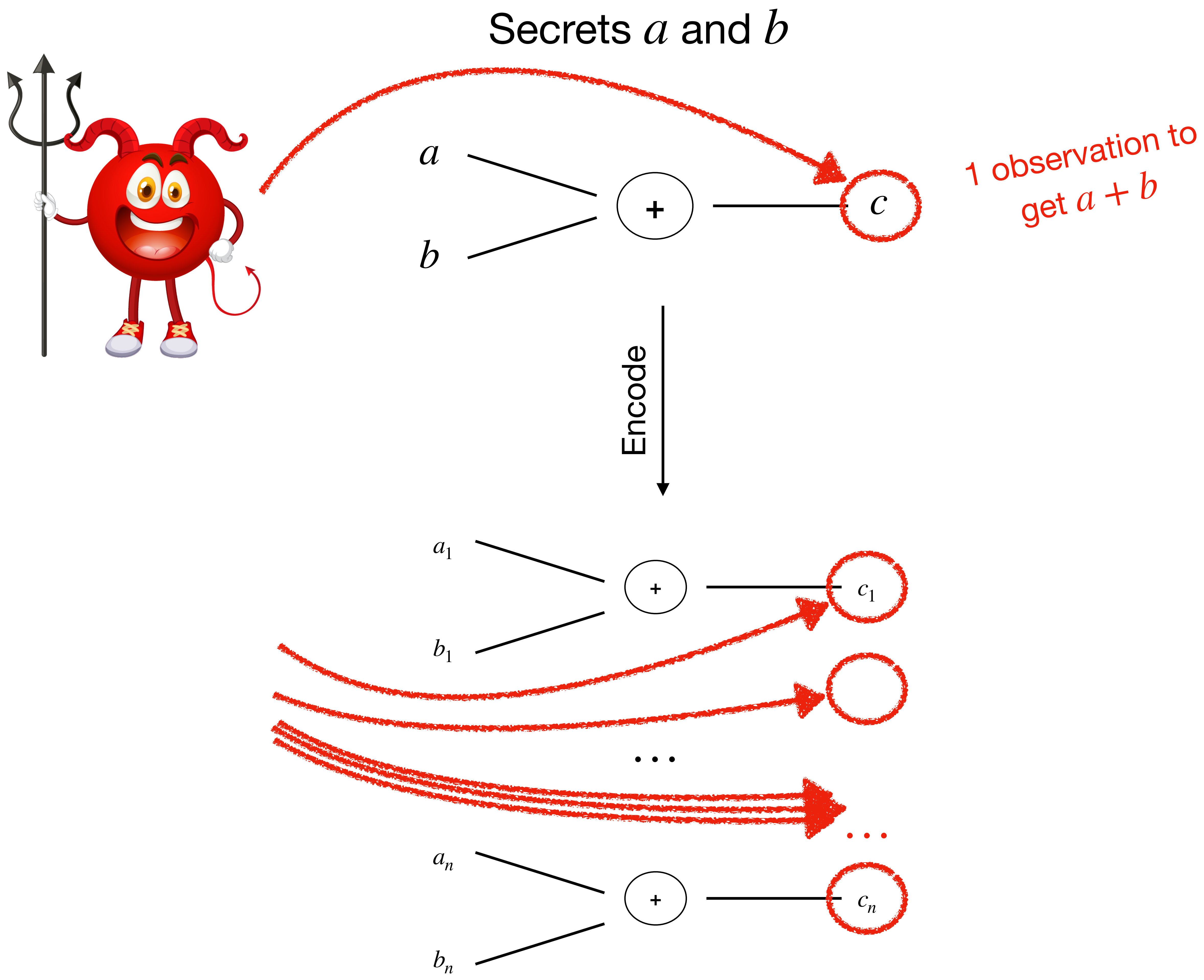
$$\ldots$$

$$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$$

$n-1$ random values

secret recombination

$$x_n \leftarrow x - x_1 \ldots - x_{n-1}$$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

$\ldots$

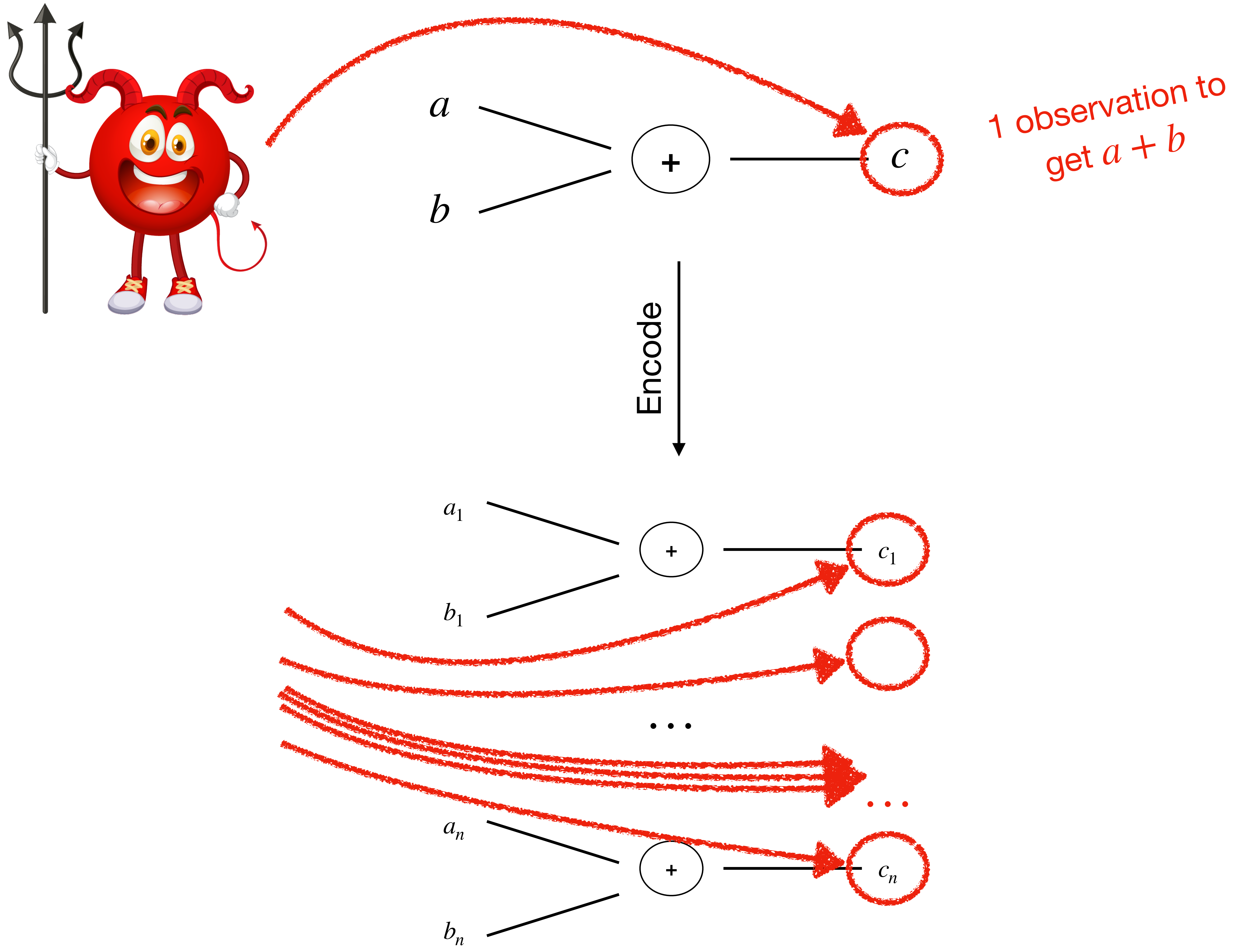
$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$

Secrets a and b



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

↓ Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

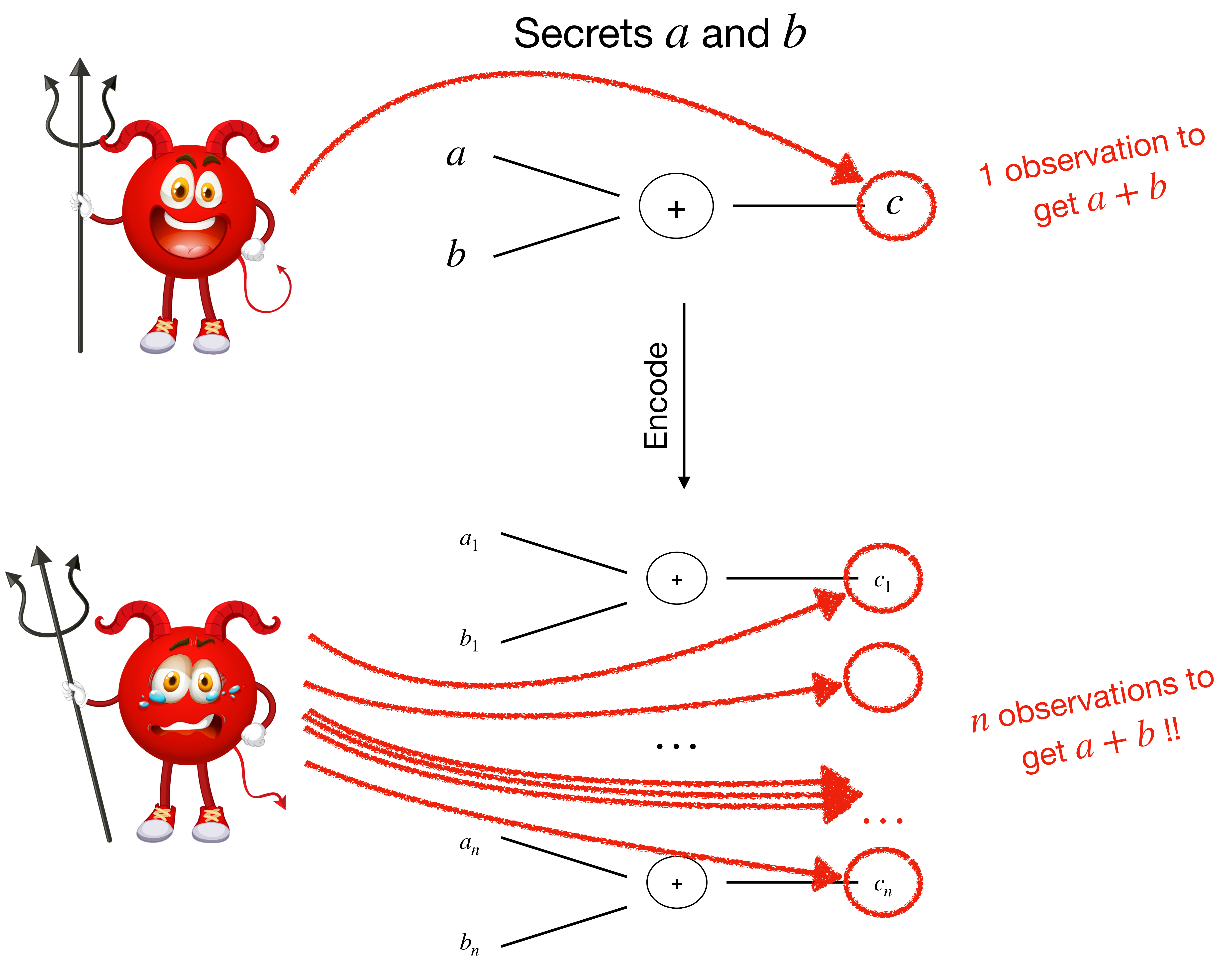
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

each observation comes with noise

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode ↓

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \overset{\$}{\leftarrow} \mathbb{F}_2$

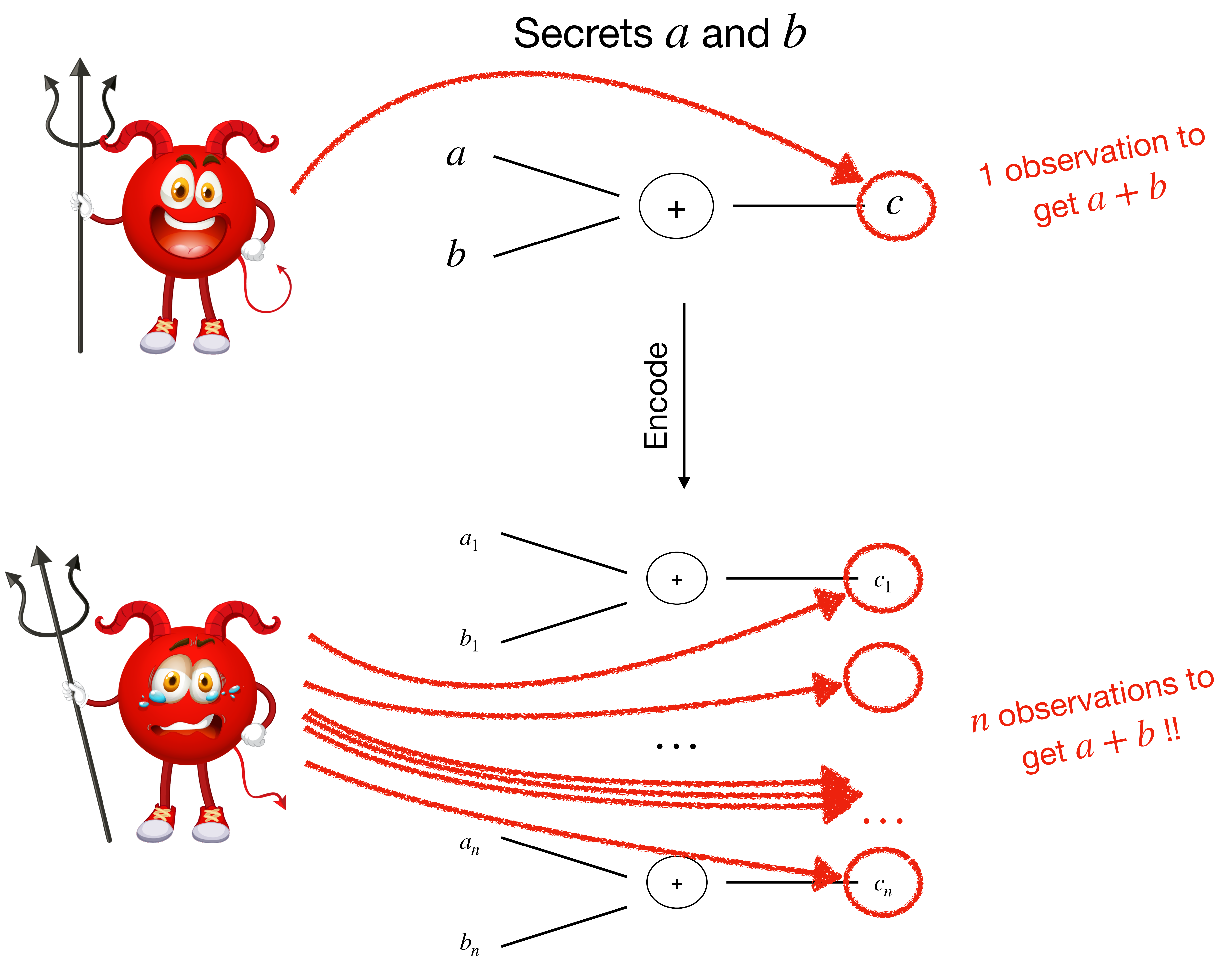
$\ldots$

$x_{n-1} \overset{\$}{\leftarrow} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

each observation comes with noise
Number of observation grows $\Longrightarrow$ exponential effort to retrieve the secret

Masking *Chari et Al [CRYPTO'99], Goubin and Patarin [CHES'99]*

Secret Variable $x \in \mathbb{F}_2$ (field)

Encode

shares

Secret Vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

s.t.

$x_1 \xleftarrow{\$} \mathbb{F}_2$

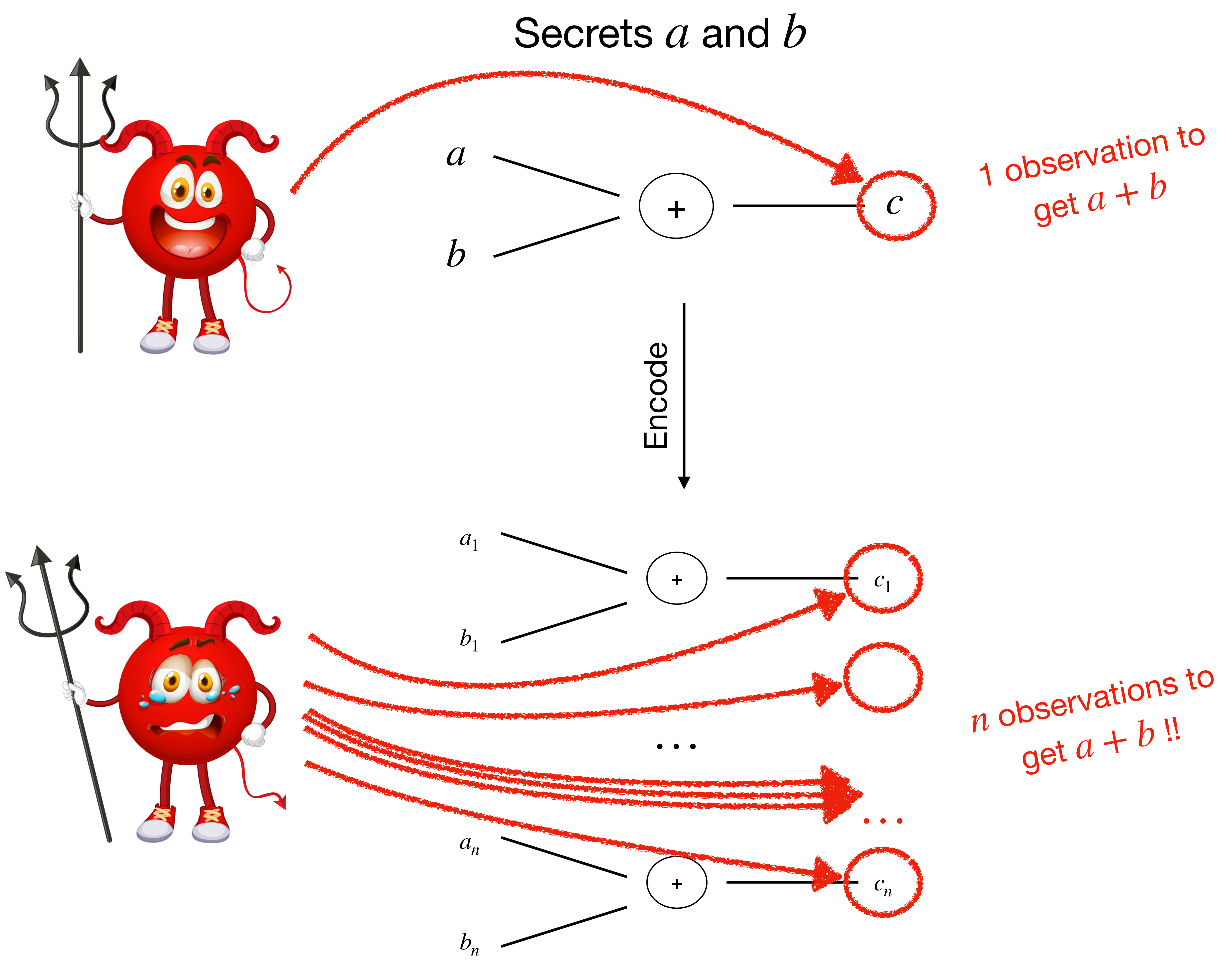
$\ldots$

$x_{n-1} \xleftarrow{\$} \mathbb{F}_2$

$n-1$ random values

secret recombination

$x_n \leftarrow x - x_1 \ldots - x_{n-1}$



Countermeasure

Gadgets

Countermeasure

Gadgets

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b $\quad \oplus \quad$ $a + b$

a, b $\quad \otimes \quad$ $a \times b$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\boxed{G_+}$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\boxed{G_\times}$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b ⊕ $a + b$

a, b ⊗ $a \times b$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\boxed{G_+}$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\boxed{G_\times}$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

a (||) a, a

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

(r) $r \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [G_+] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [$G_\times$] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

random

(r) $r \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [G_+] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [$G_\times$] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

random

(r) $r \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [G_+] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ [$G_\times$] $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

$(a_1, \ldots, a_n)$ [$G_{||}$]

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

random

(r) $r \stackrel{\$}{\leftarrow} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

new fresh shares

$(a_1, \ldots, a_n)$ $G_{||}$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

random

(r) $r \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

$(a_1, \ldots, a_n)$ $G_{refresh}$

new fresh shares

$(a_1, \ldots, a_n)$ $G_{||}$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b	$+$	$a + b$
a, b	$\times$	$a \times b$
a	copy $\|\|$	a, a
	random r	$r \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$	G_+	$(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$
$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$	$G_\times$	$(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$
$(a_1, \ldots, a_n)$	$G_{refresh}$	new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$
$(a_1, \ldots, a_n)$	$G_{\|\|}$	new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$; $(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b (+) $a + b$

a, b (×) $a \times b$

copy

a (||) a, a

random

(r) $r \xleftarrow{\$} \mathbb{F}_2$

G_r $r_1 \xleftarrow{\$} \mathbb{F}_2, \ldots, r_n \leftarrow \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

$(a_1, \ldots, a_n)$ $G_{refresh}$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(a_1, \ldots, a_n)$ $G_{||}$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

Gadgets

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b $\quad (+) \quad$ $a + b$

a, b $\quad (\times) \quad$ $a \times b$

copy

a $\quad (||) \quad$ a, a

random

$(r) \quad r \xleftarrow{\$} \mathbb{F}_2$

G_r $\quad r_1 \xleftarrow{\$} \mathbb{F}_2, \ldots, r_n \leftarrow \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\quad [G_+] \quad$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $\quad [G_\times] \quad$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

$(a_1, \ldots, a_n)$ $\quad [G_{refresh}] \quad$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(a_1, \ldots, a_n)$ $\quad [G_{||}] \quad$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$, $(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

Gadgets

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

Operations over variables $\mathbb{F}_2$

Atomic gates

a, b $(+)$ $a + b$

a, b $(\times)$ $a \times b$

copy

a $(||)$ a, a

random

(r) $r \xleftarrow{\$} \mathbb{F}_2$

G_r $r_1 \xleftarrow{\$} \mathbb{F}_2, \ldots, r_n \leftarrow \xleftarrow{\$} \mathbb{F}_2$

Operations over masked variables in $\mathbb{F}_2^n$

n-share Gadgets formed of atomic gates

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ G_+ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a + b$

$(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ $G_\times$ $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a \times b$

n^2 terms $a_1 \times b_1, \ldots, a_n \times b_n$ to recombine

$(a_1, \ldots, a_n)$ $G_{refresh}$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

$(a_1, \ldots, a_n)$ $G_{||}$ new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$ $(d_1, \ldots, d_n)$ s.t. $d_1 + \ldots + d_n = a$

Countermeasure

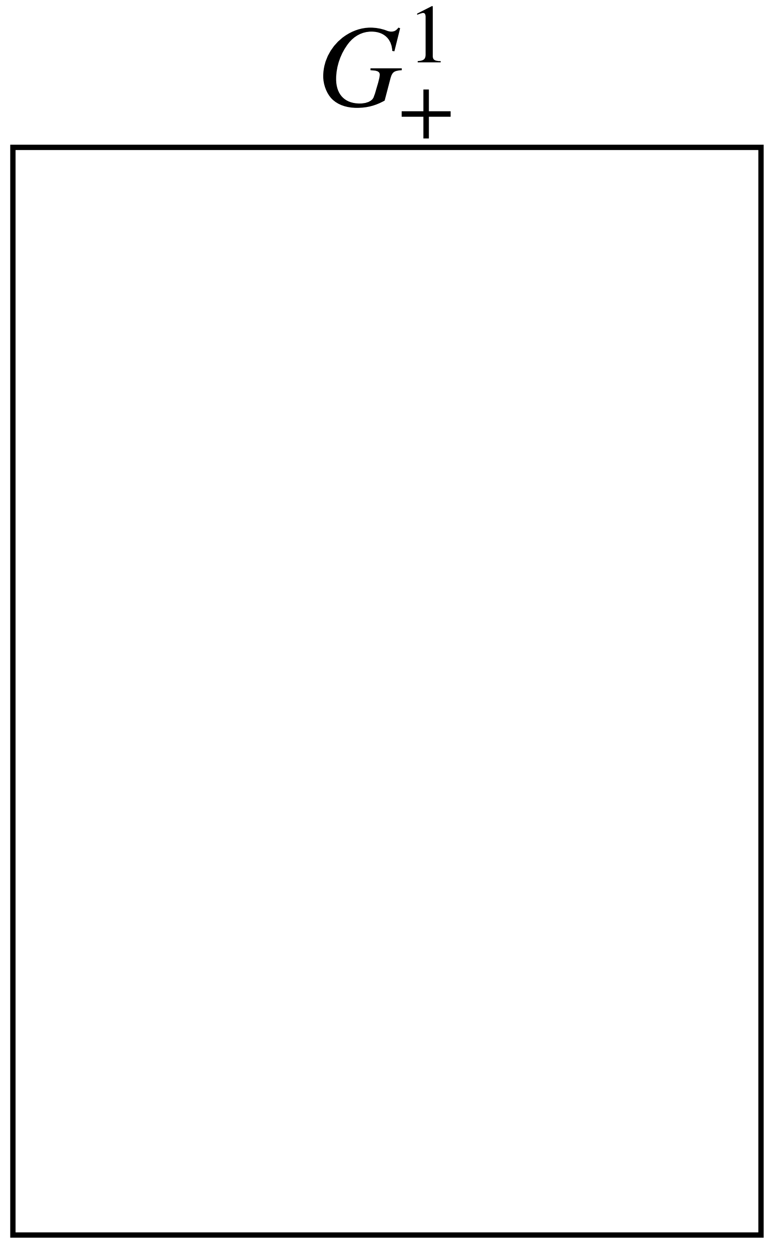
Gadgets with $n = 2$

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

Countermeasure

Gadgets with $n = 2$

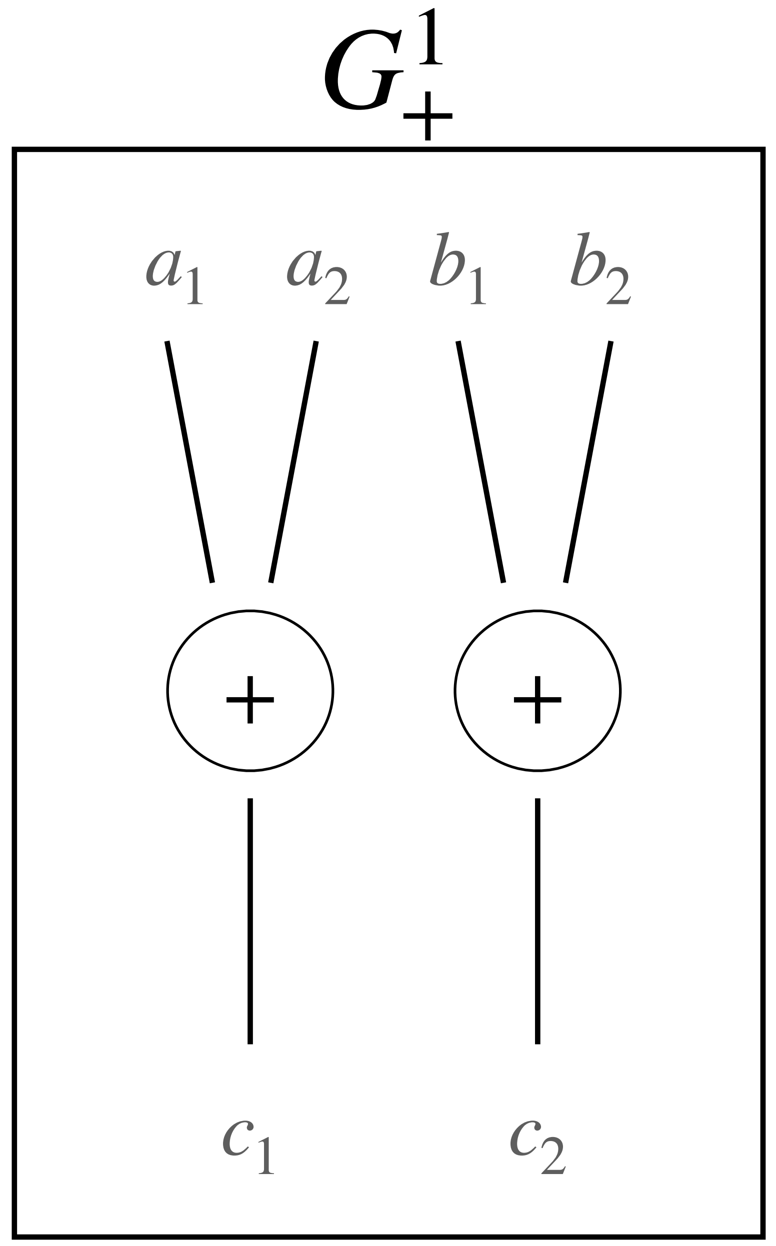
Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets



Countermeasure

Gadgets with $n = 2$

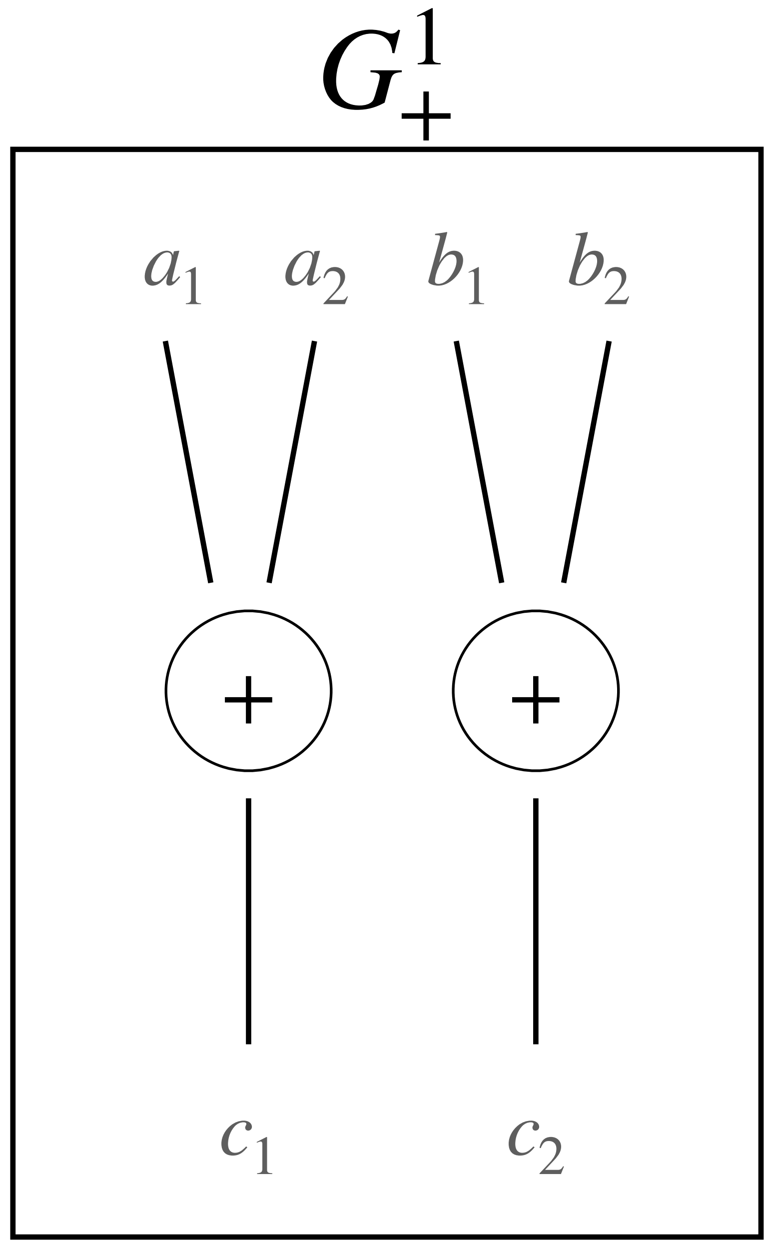
Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

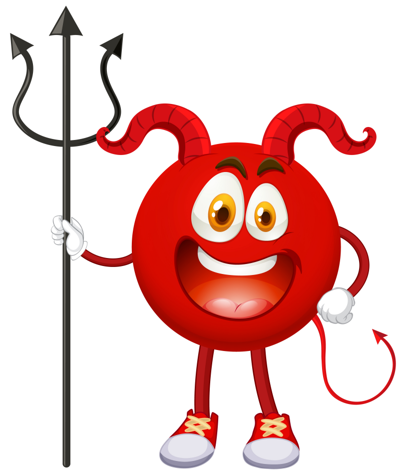


Countermeasure

Gadgets with $n = 2$

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

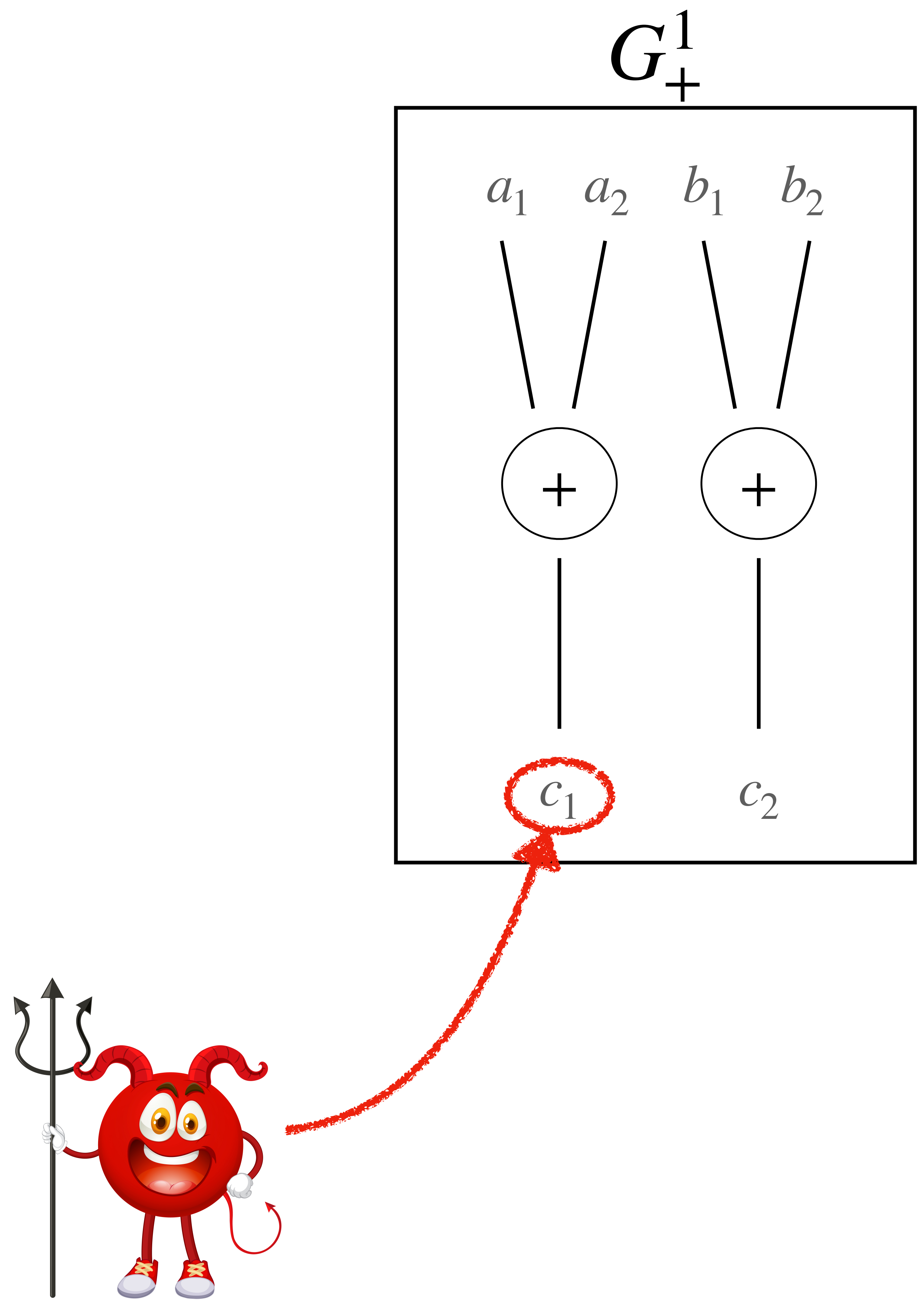




Countermeasure

Gadgets with $n = 2$

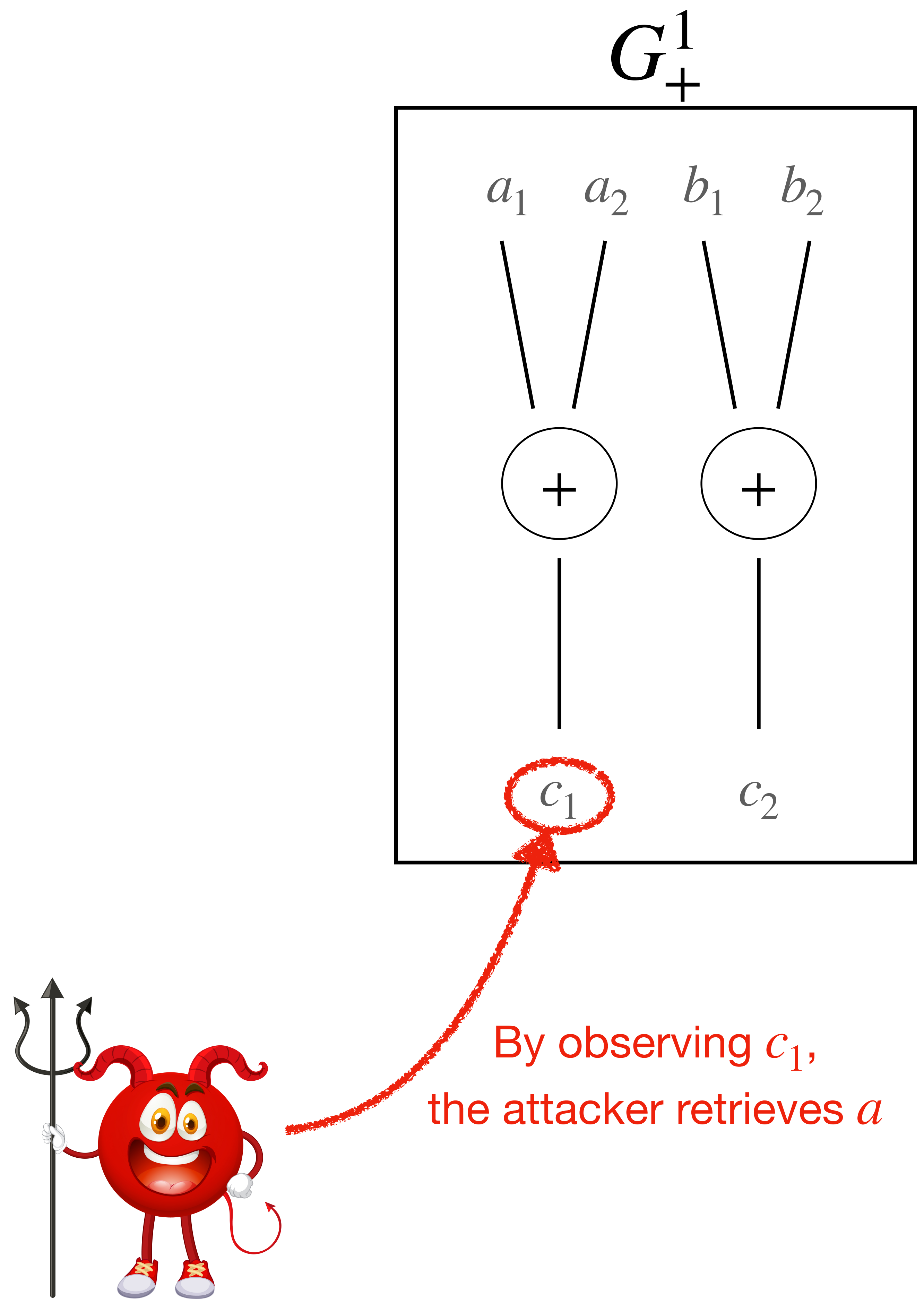
Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets



Countermeasure

Gadgets with $n = 2$

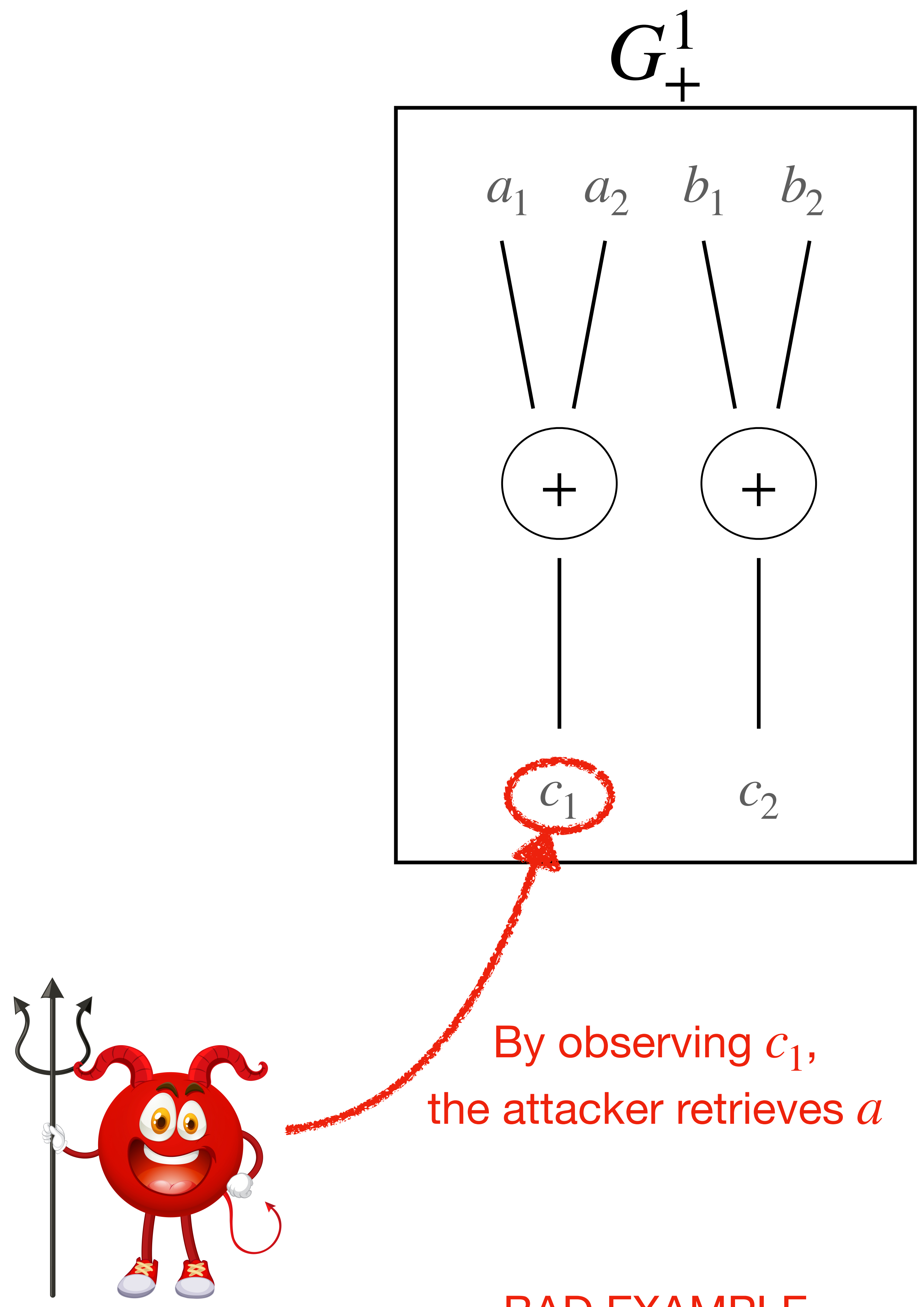
Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets



Countermeasure

Gadgets with $n = 2$

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

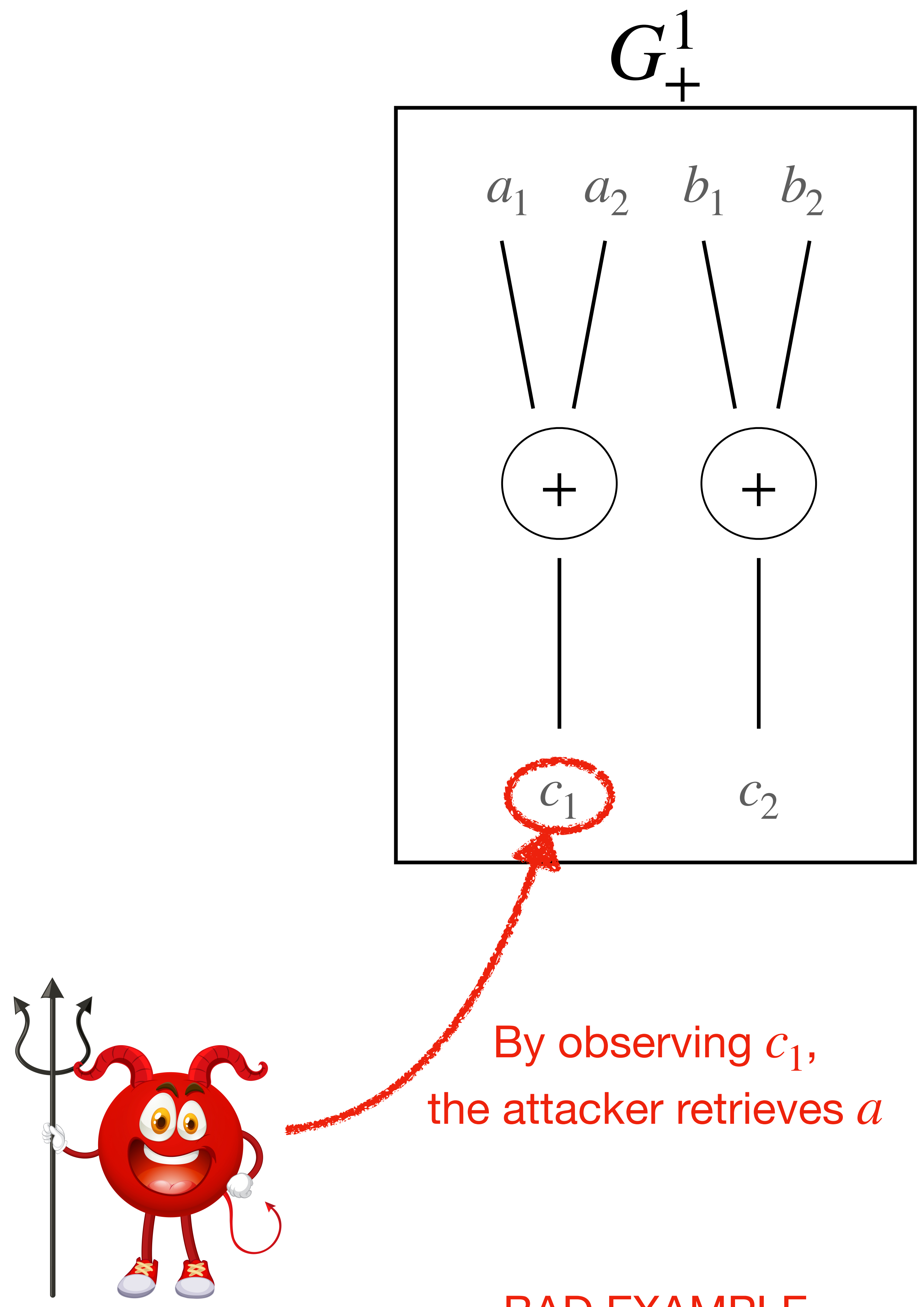


BAD EXAMPLE

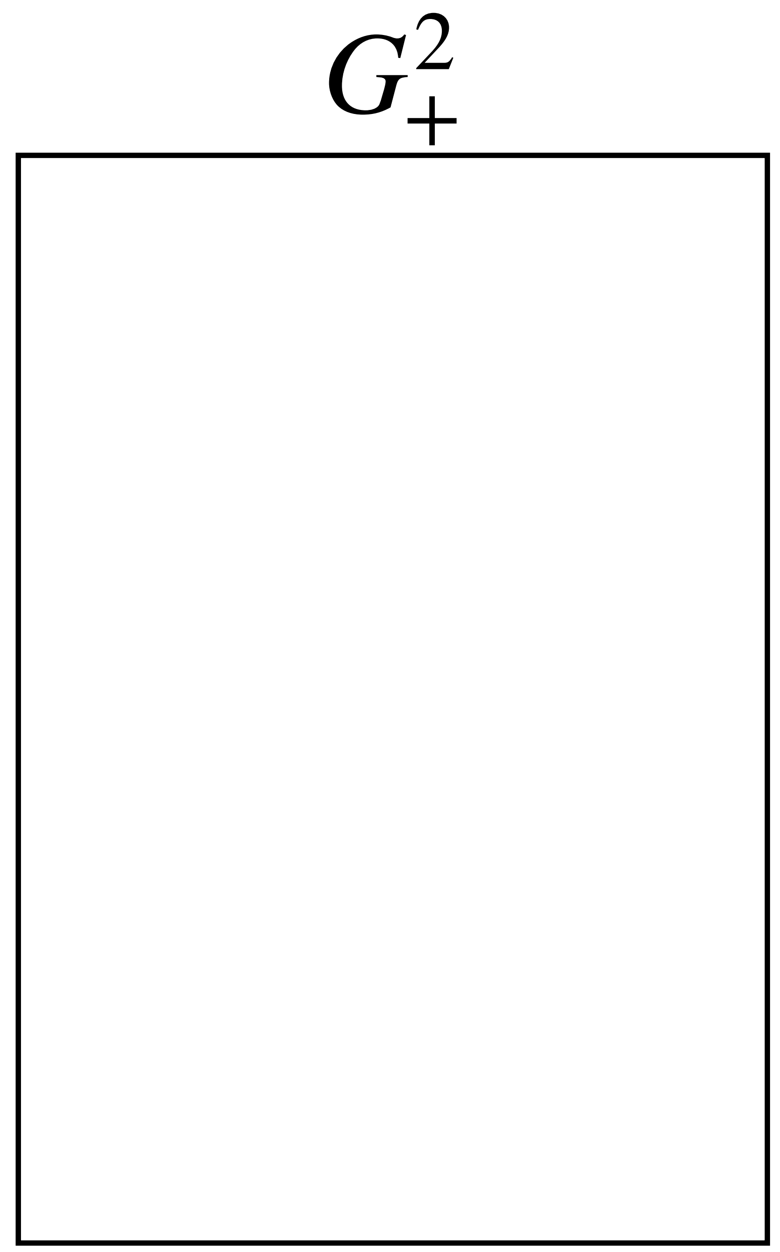
Countermeasure

Gadgets with $n = 2$

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets



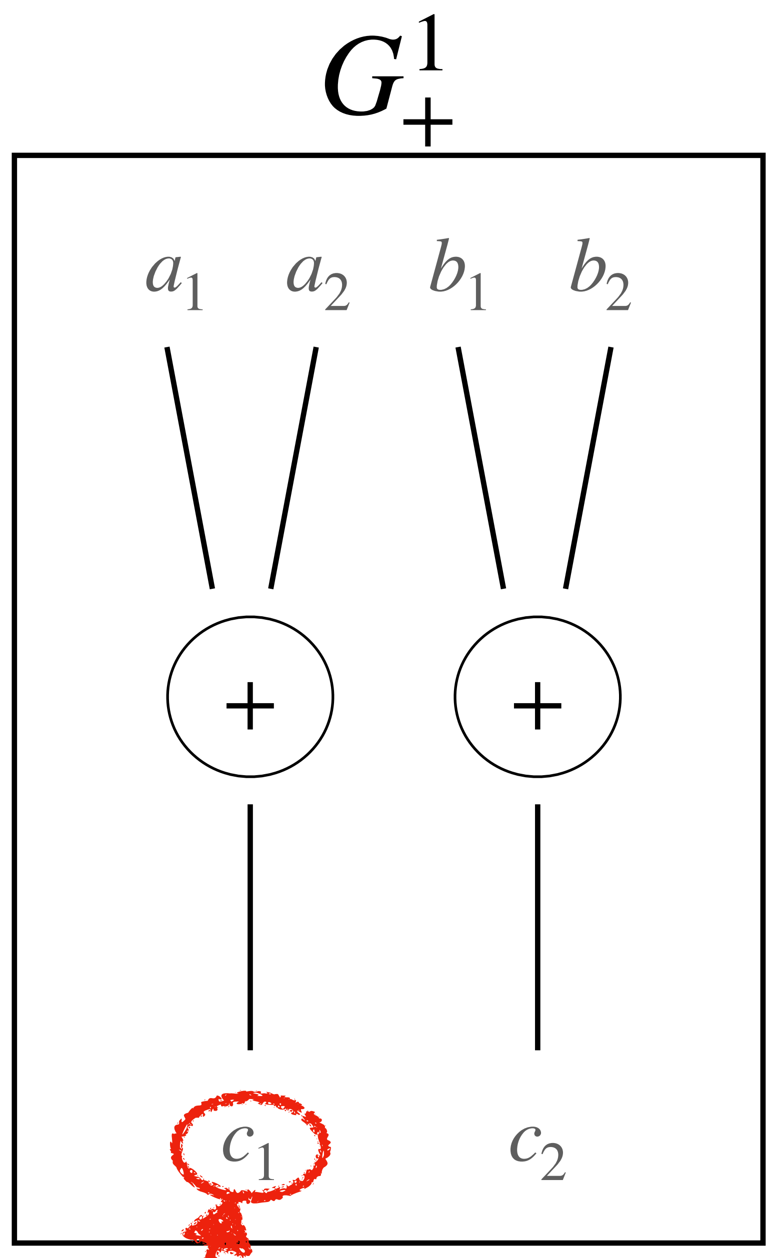
BAD EXAMPLE

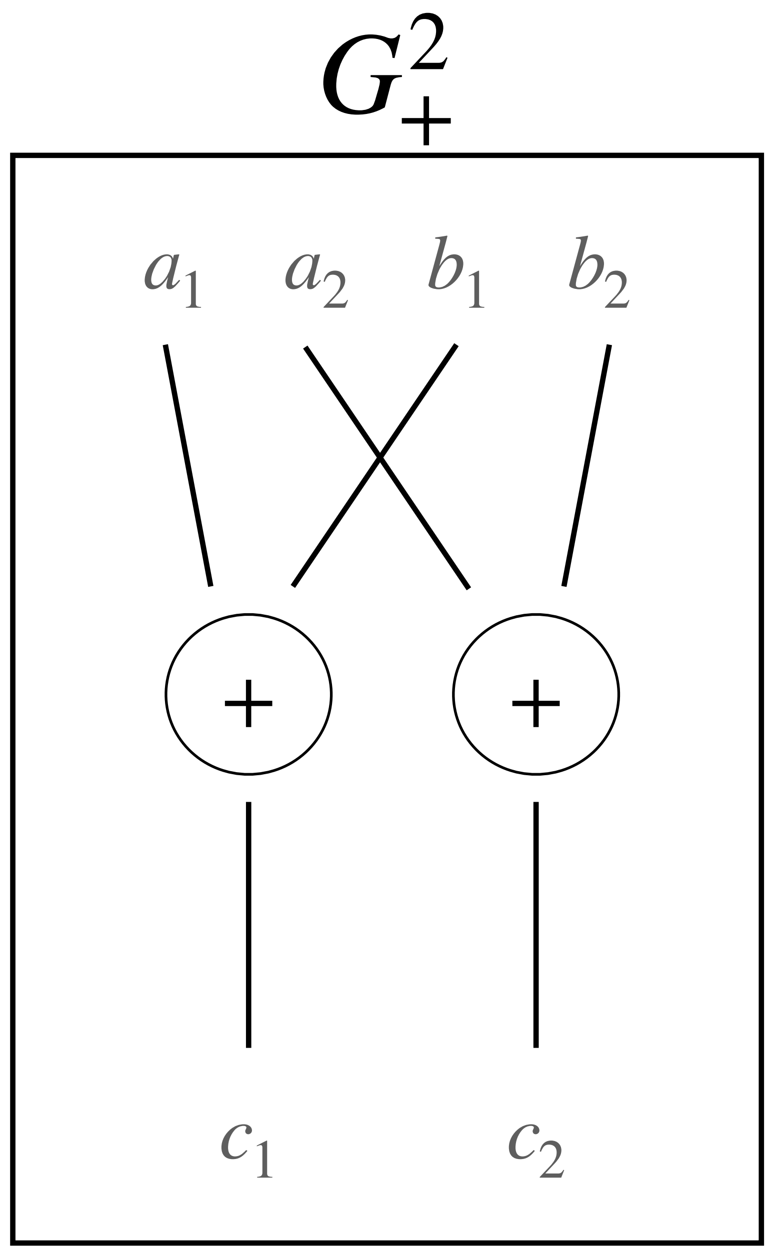


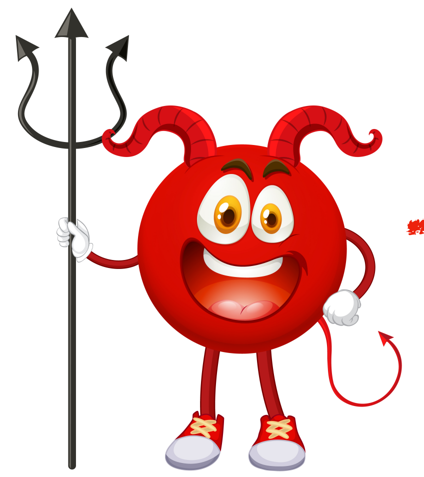
Countermeasure

Gadgets with $n = 2$

Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets







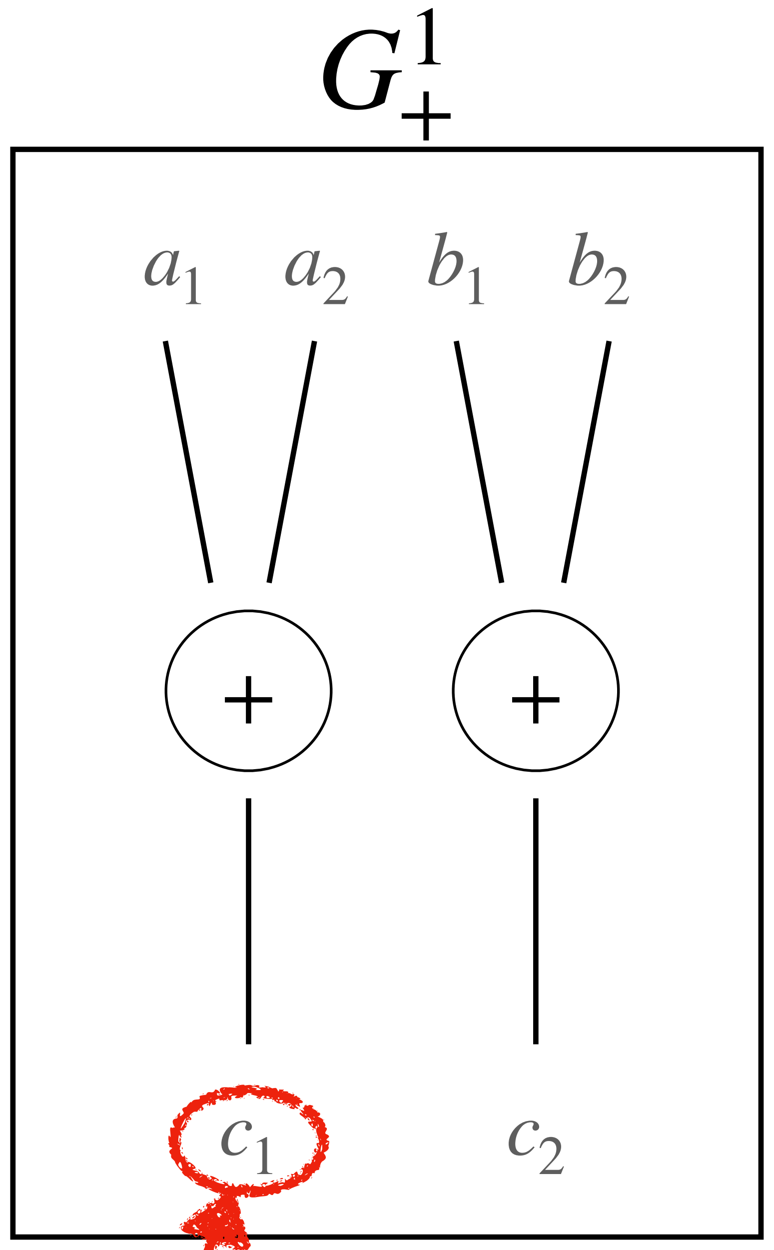
By observing c_1,
the attacker retrieves a

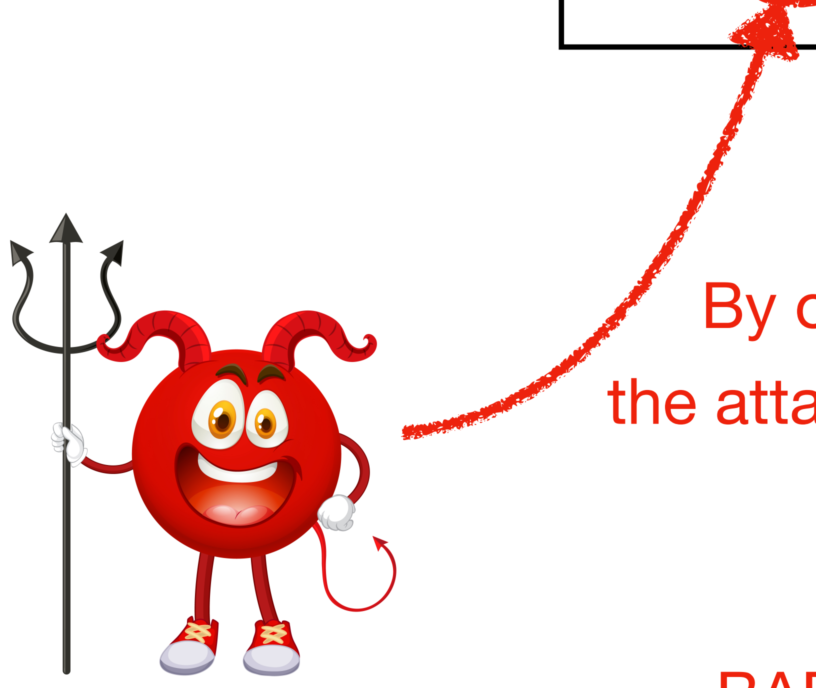
BAD EXAMPLE

Countermeasure

Gadgets with $n = 2$

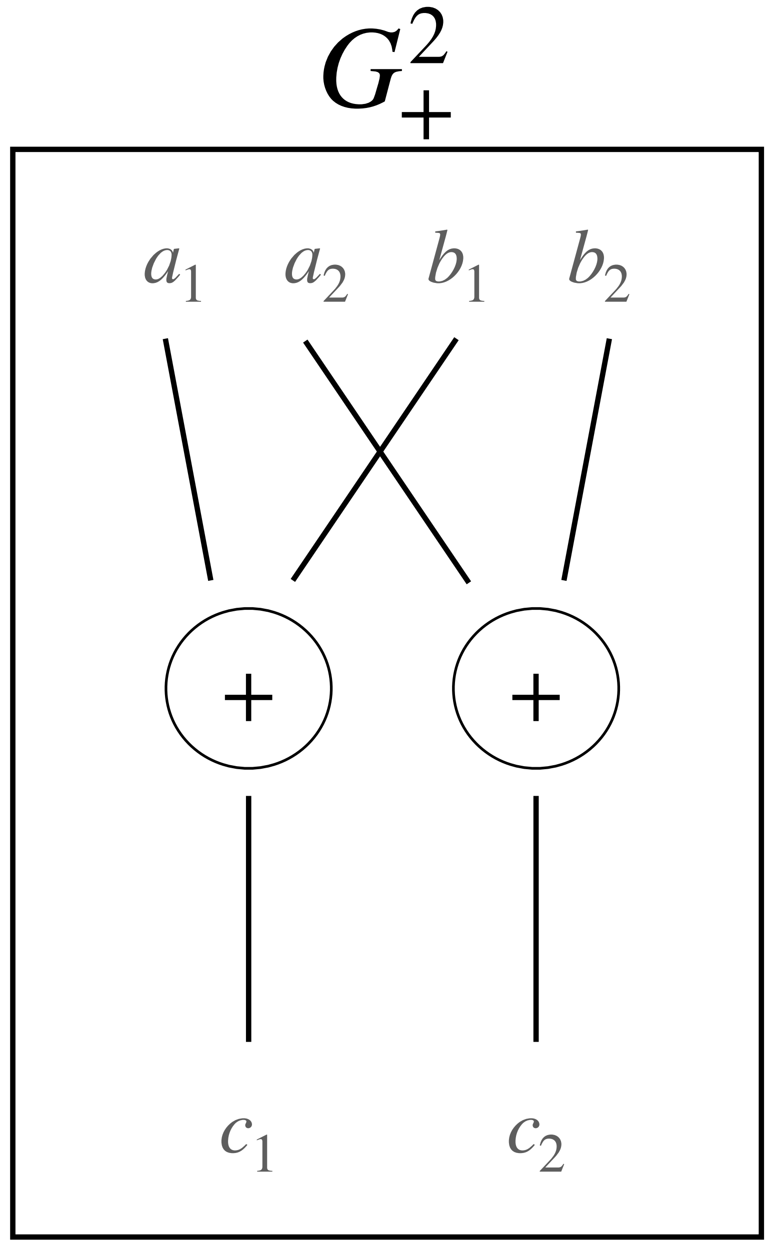
Intuitively, a gadget is considered « secure » if an attacker needs at least n observations to retrieve the secrets

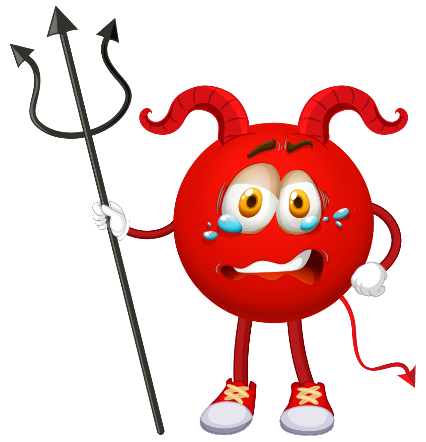




By observing c_1,
the attacker retrieves a

BAD EXAMPLE





No single observation can
retrieve a or b

GOOD EXAMPLE

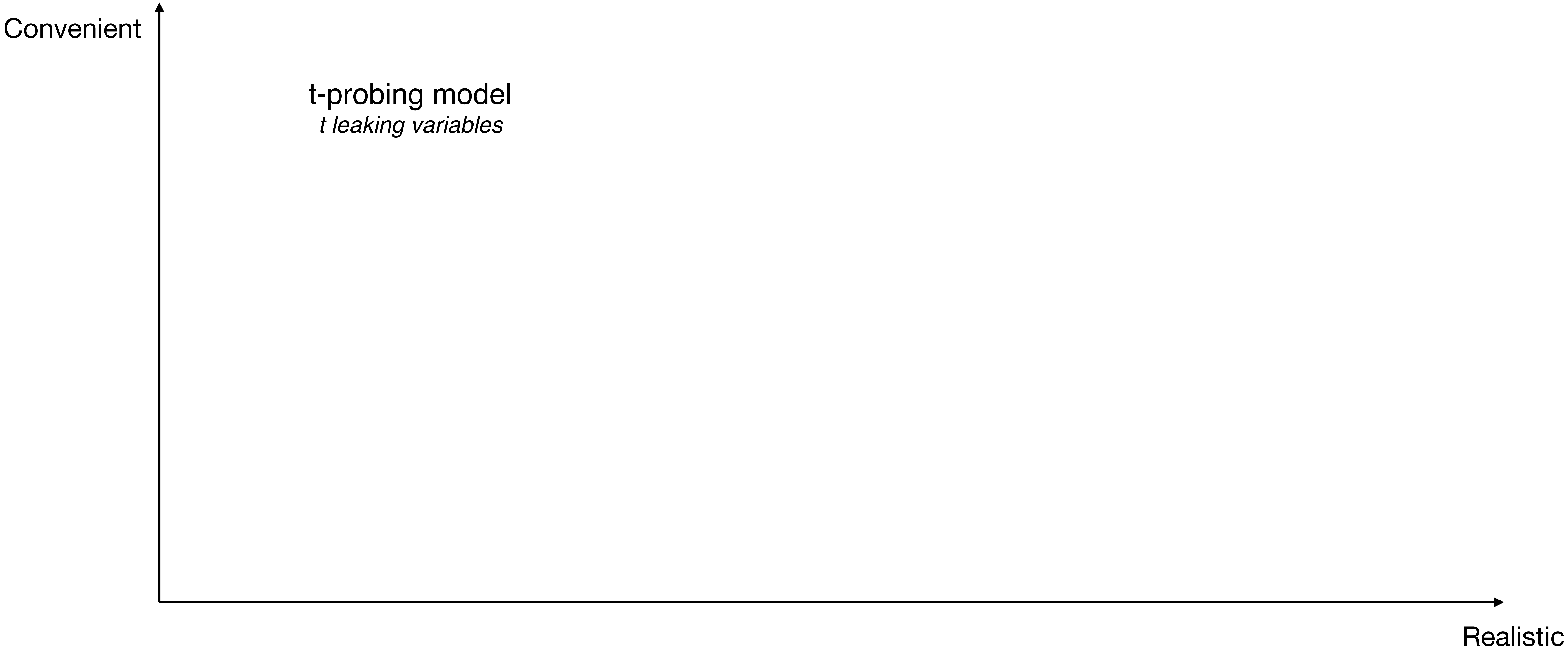
Theoretical Security

Leakage models

Theoretical Security
Leakage models

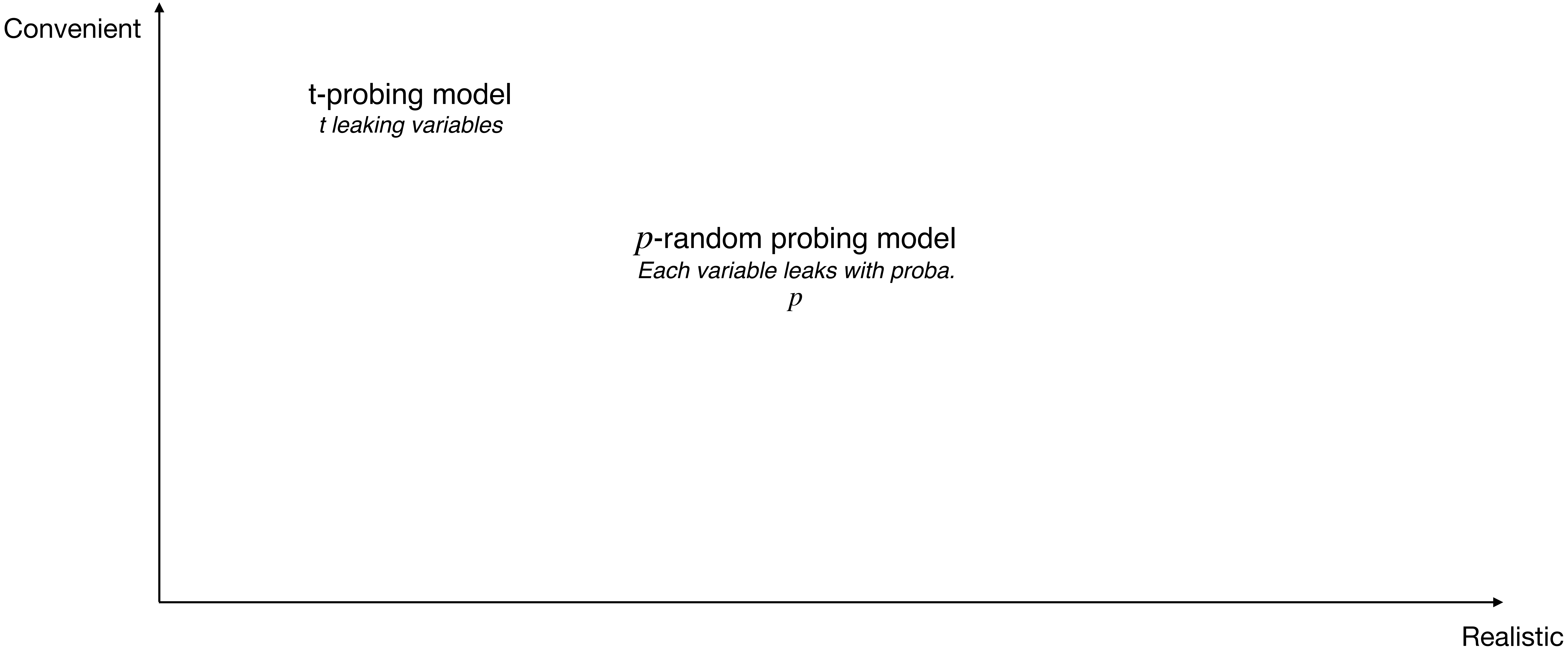
Theoretical Security

Leakage models



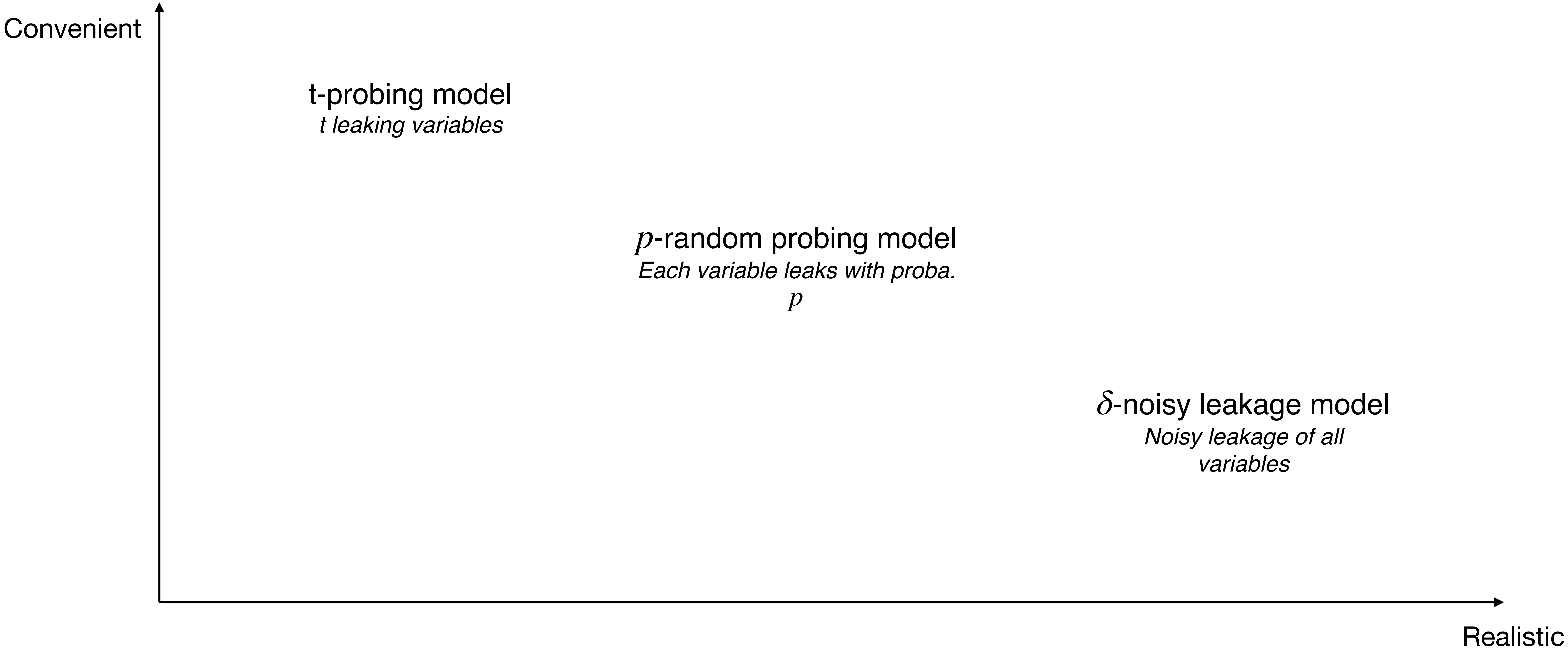
Theoretical Security

Leakage models



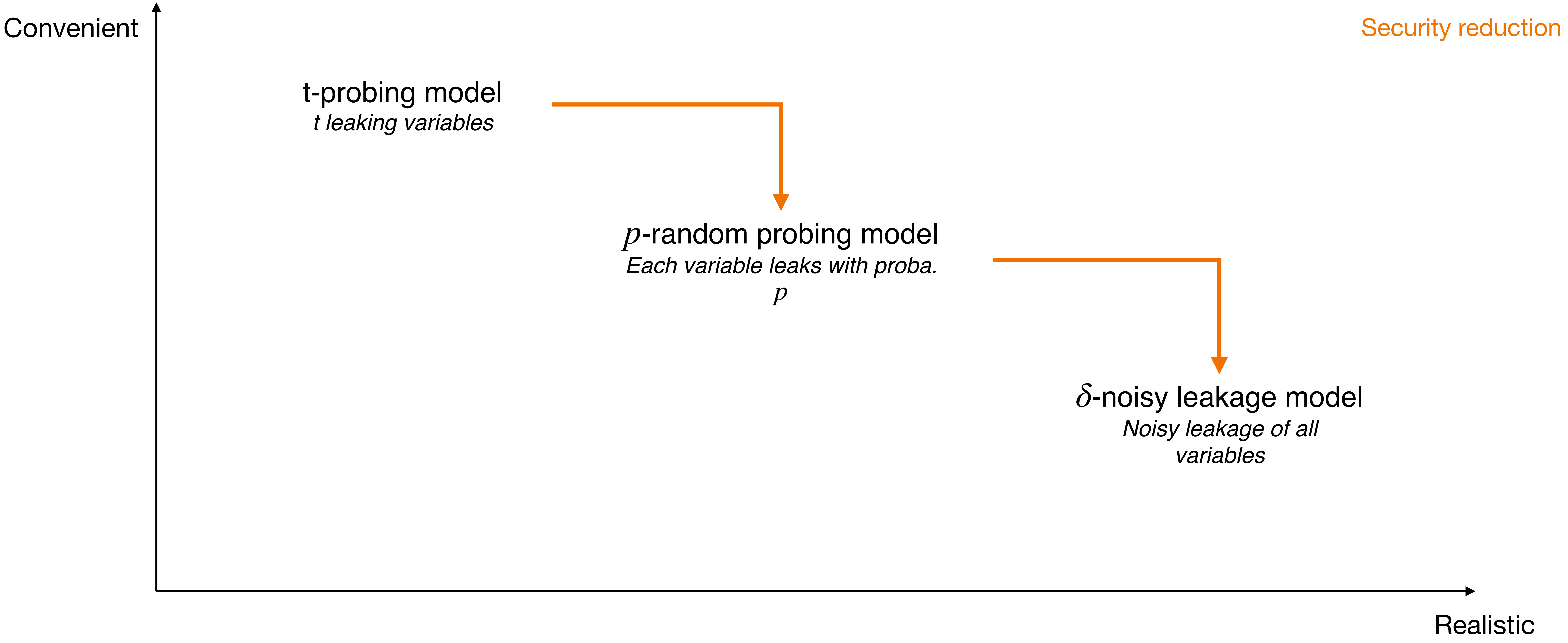
Theoretical Security

Leakage models



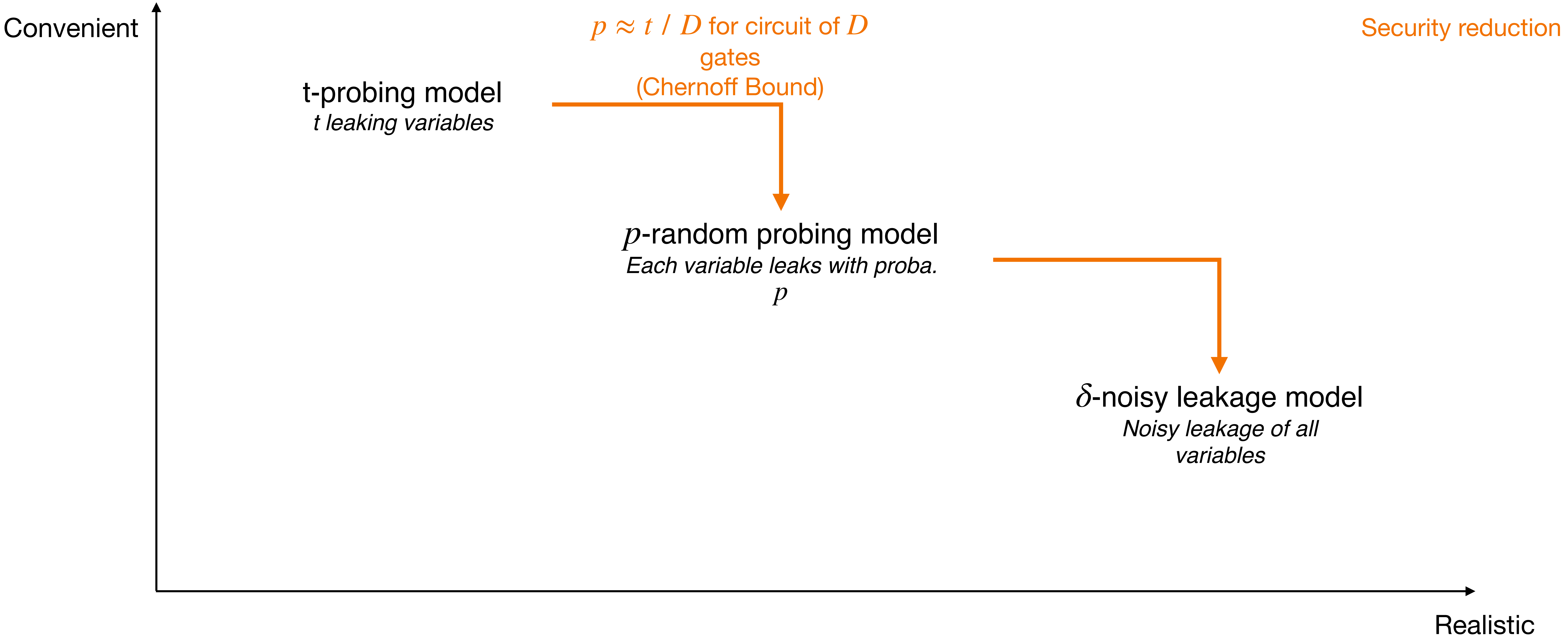
Theoretical Security

Leakage models



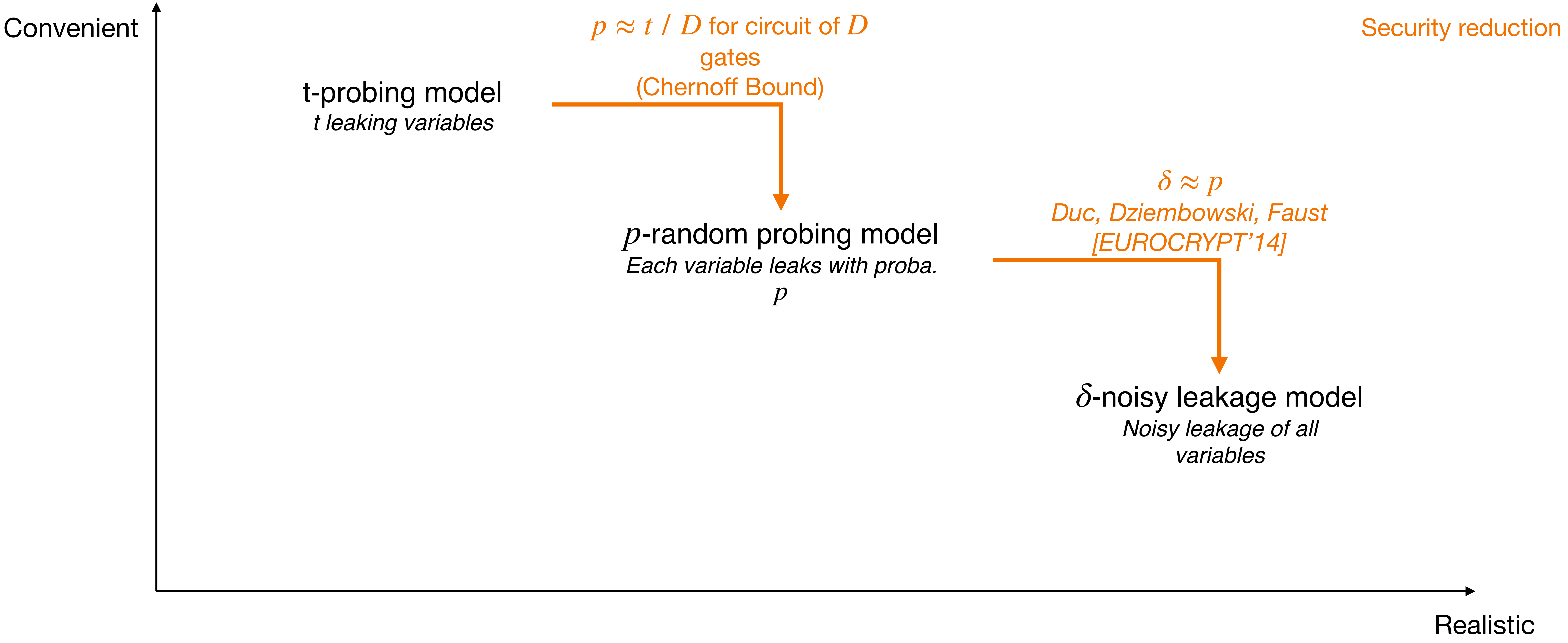
Theoretical Security

Leakage models



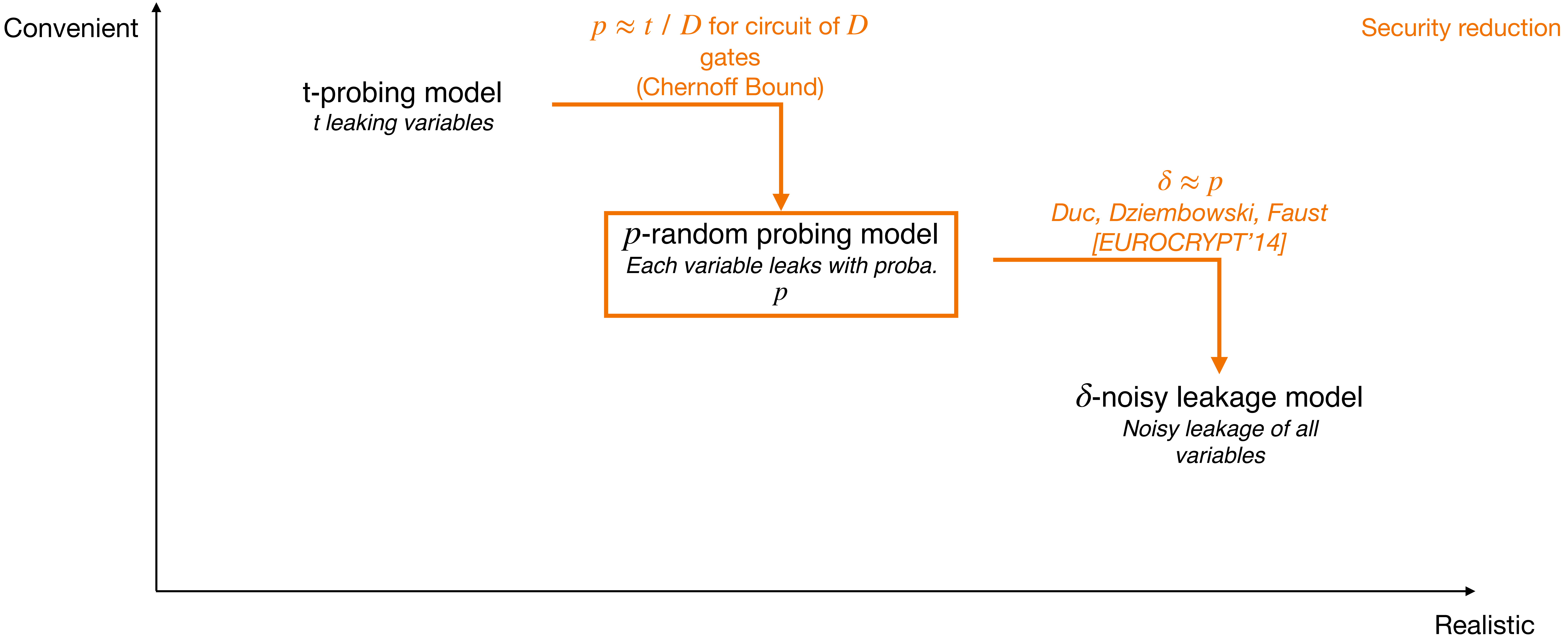
Theoretical Security

Leakage models



Theoretical Security

Leakage models



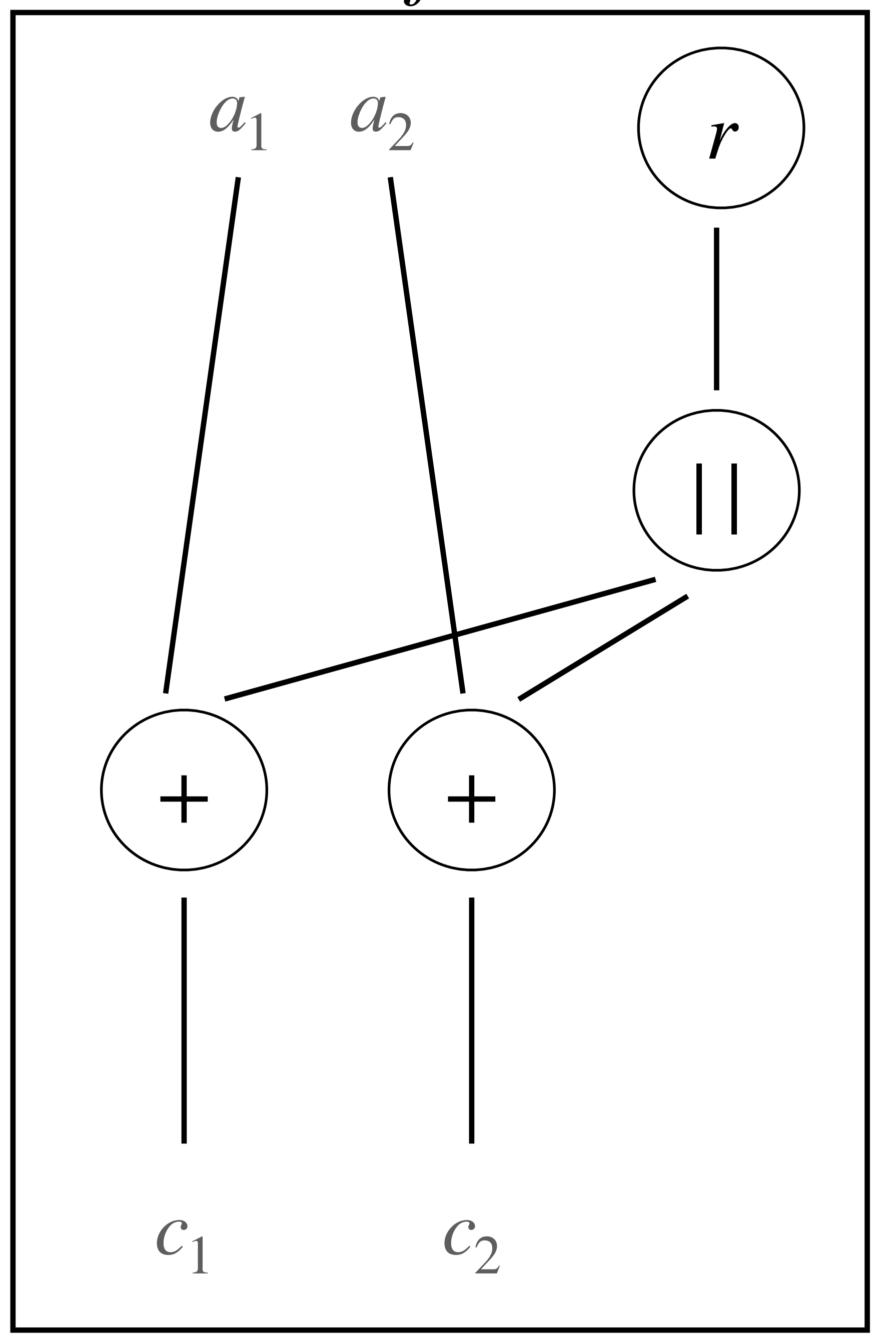
Random Probing Security

Definition

Random Probing Security

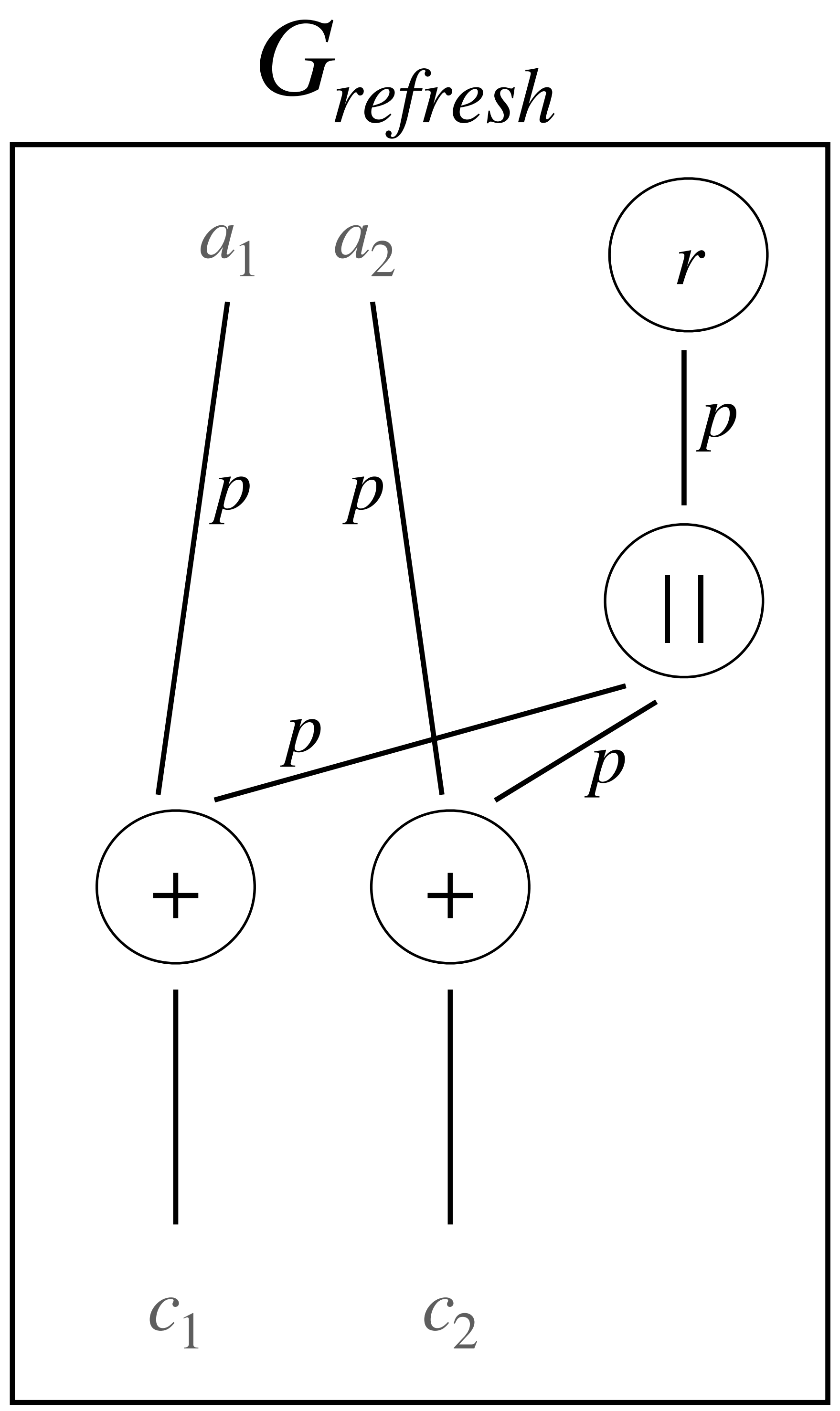
Definition

$G_{refresh}$



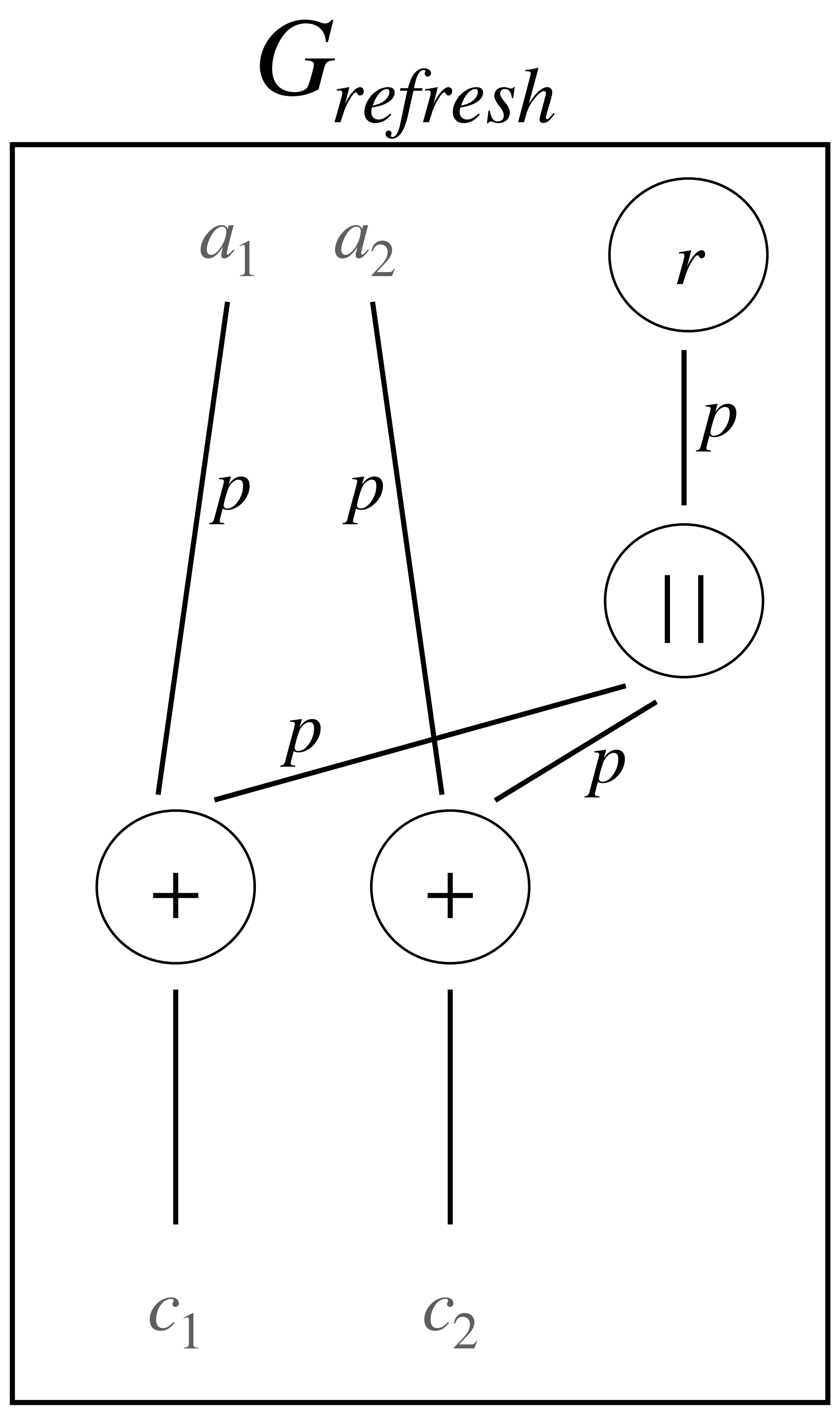
Random Probing Security

Definition



Random Probing Security

Definition

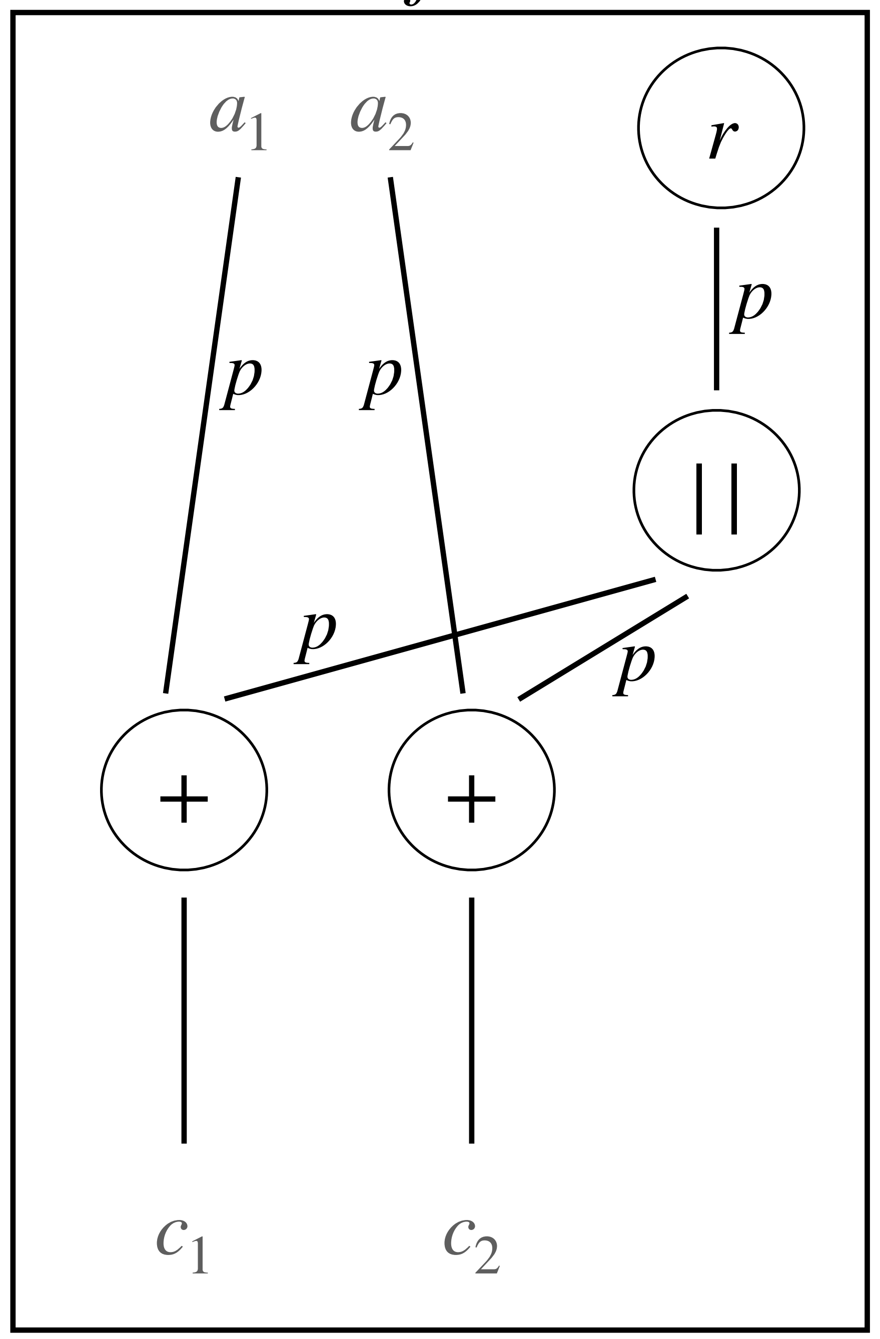


Choice: no leak on output shares, inputs of the next circuit

Random Probing Security

Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

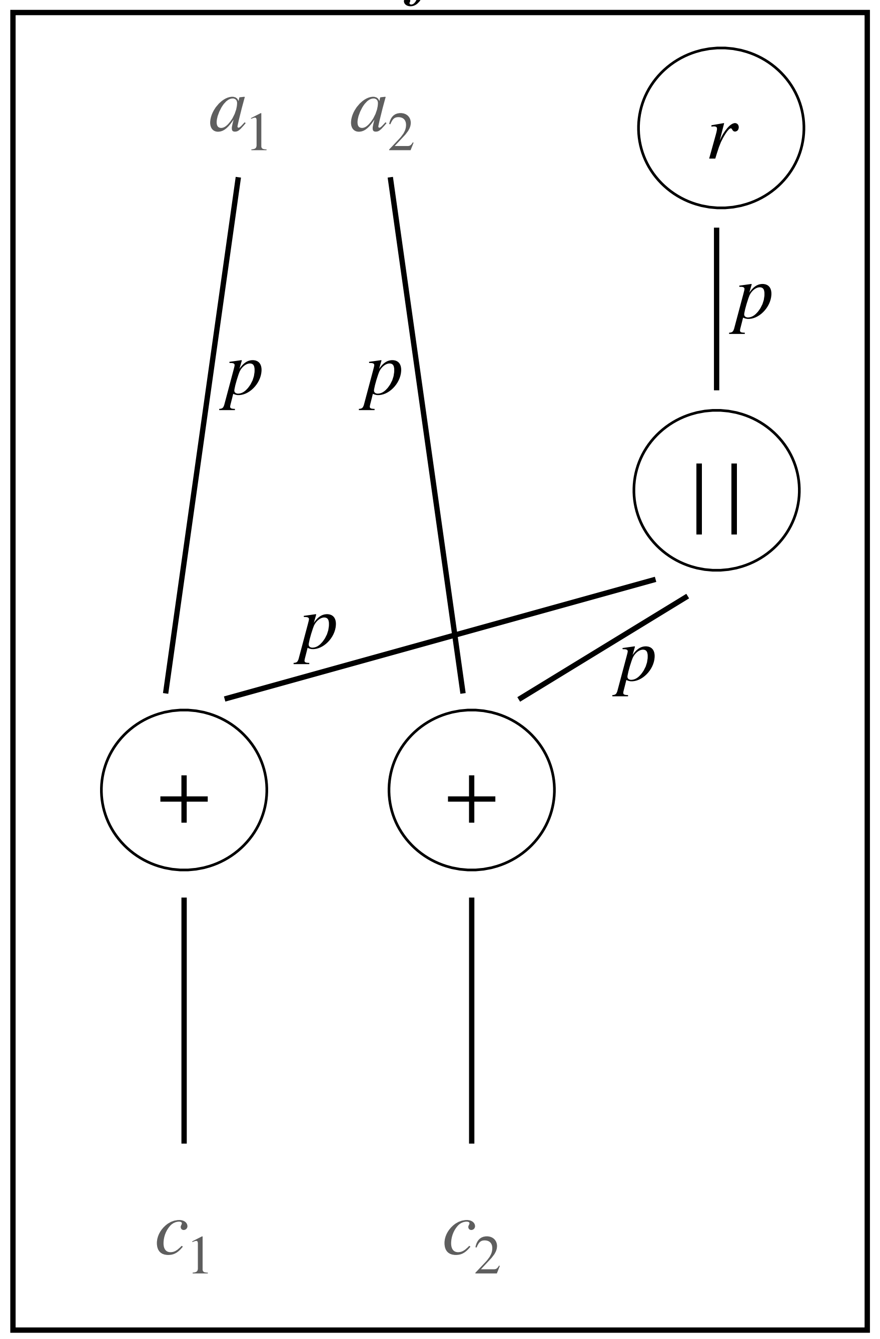
(p, ε) – random probing security

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

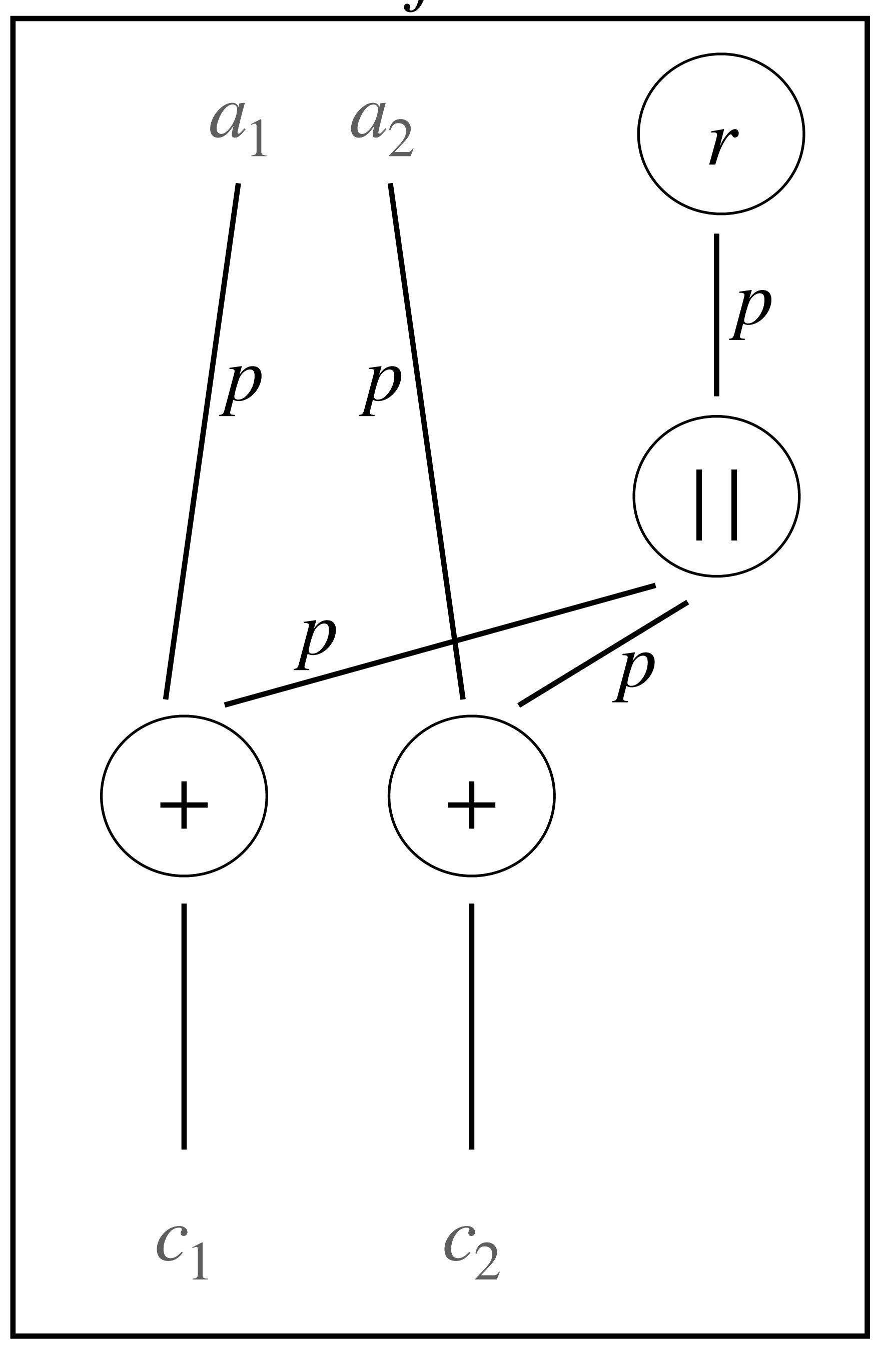
W set of wires

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

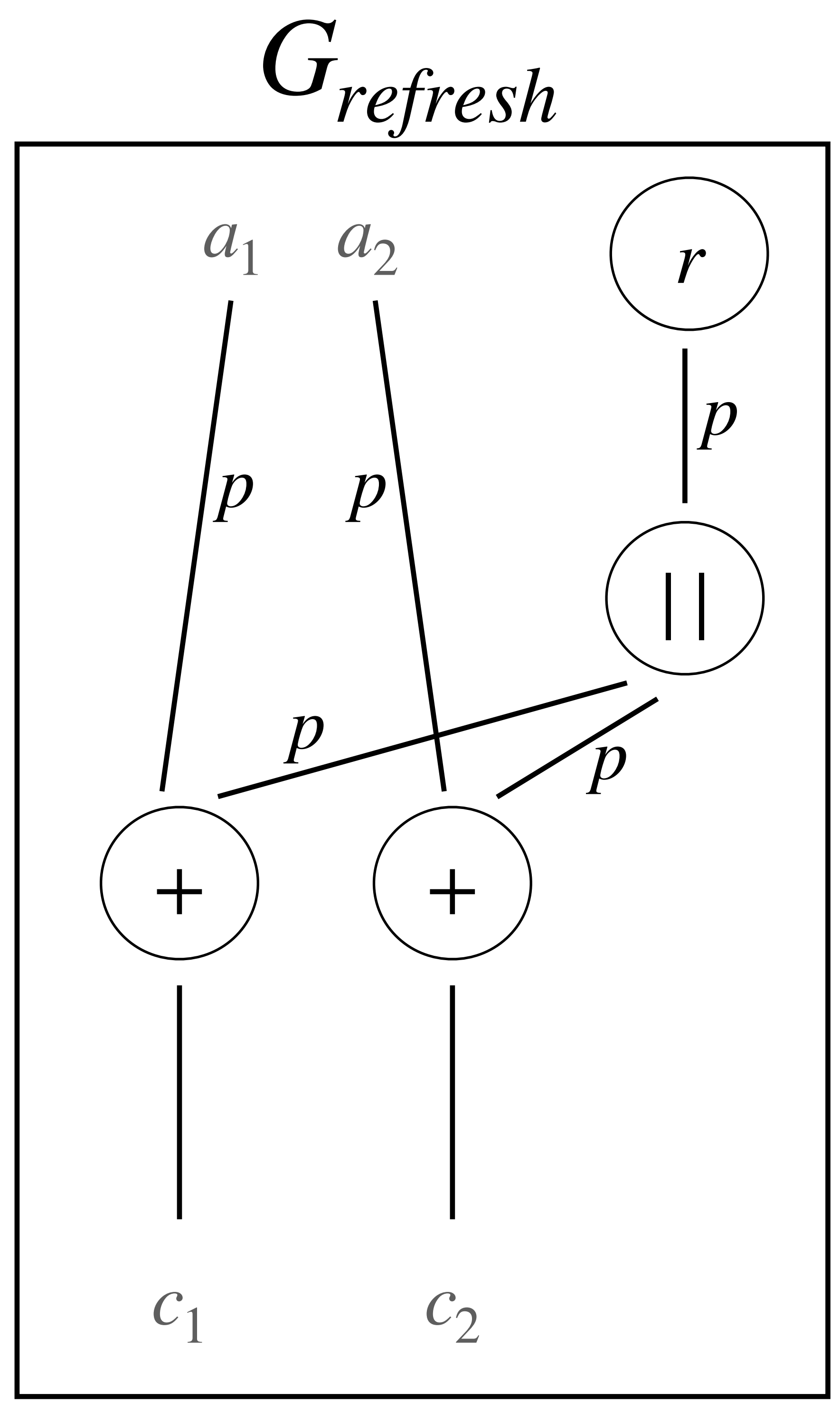
(p, ε) – random probing security

W set of wires

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

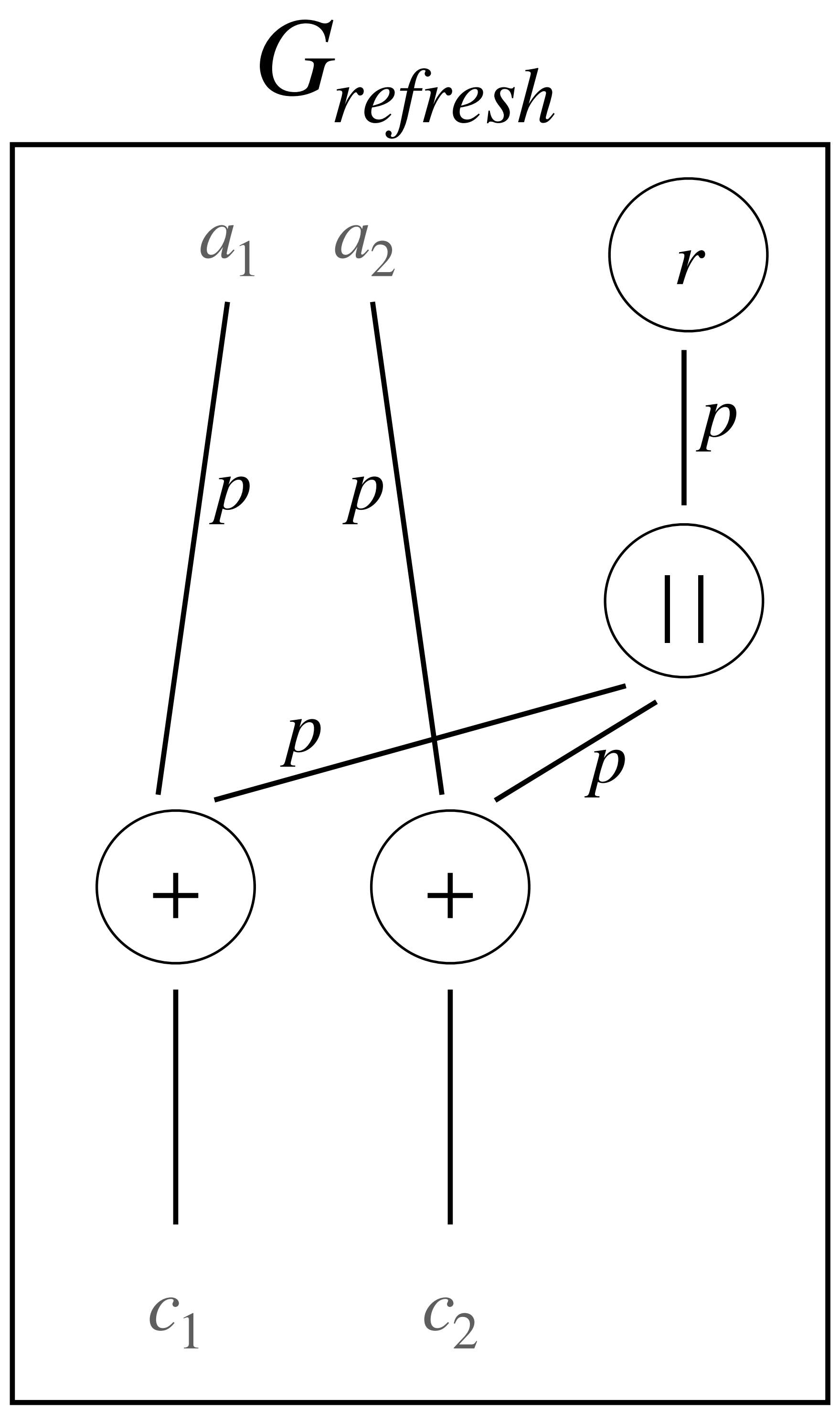
W set of wires

Independent from secrets ?

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

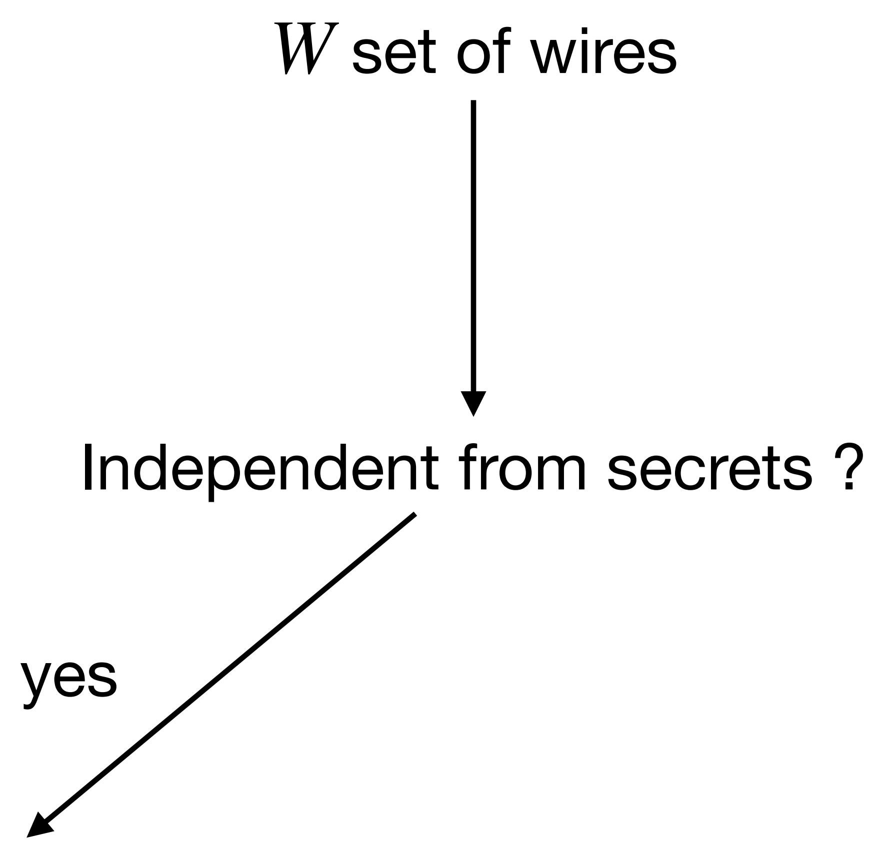
Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

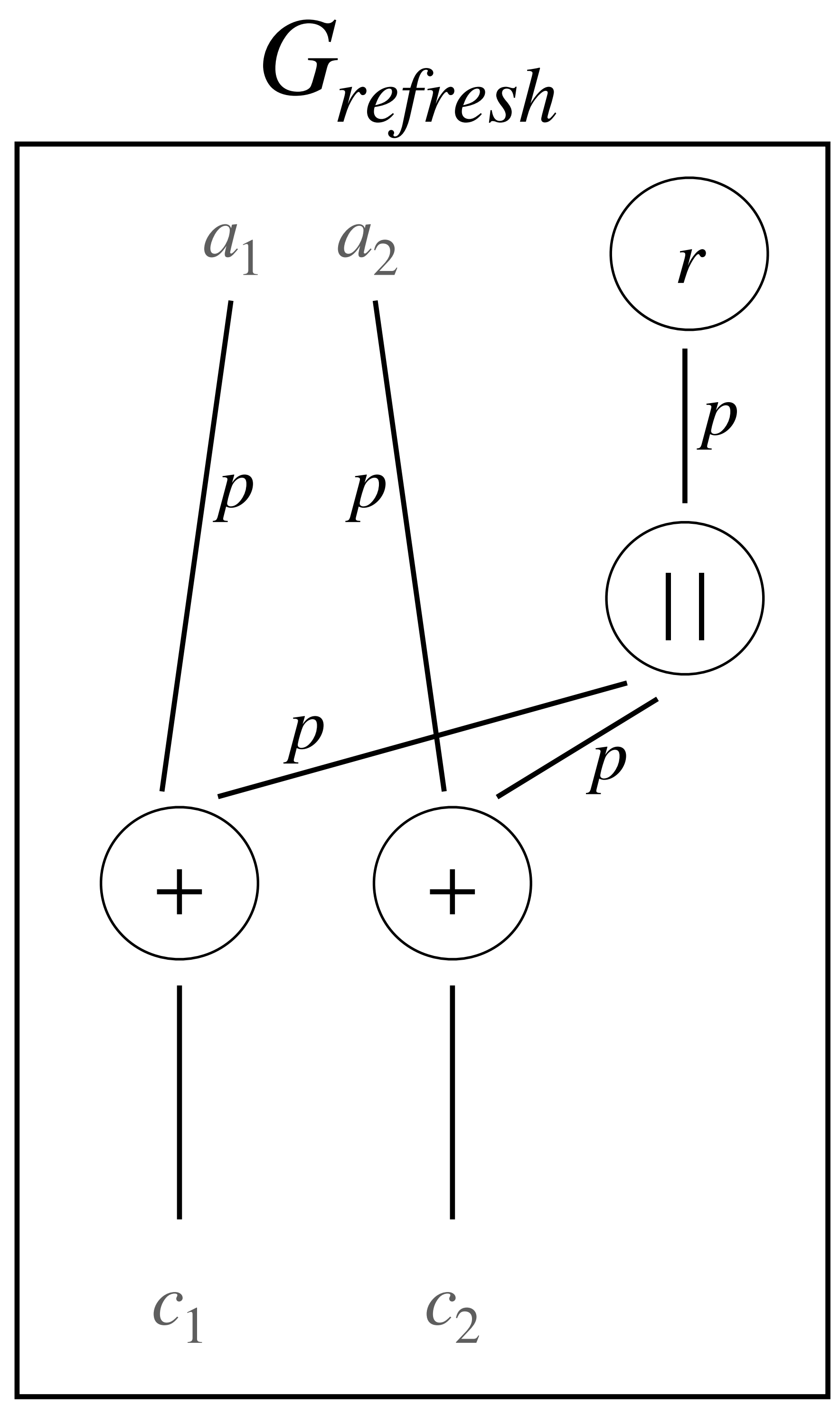
(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

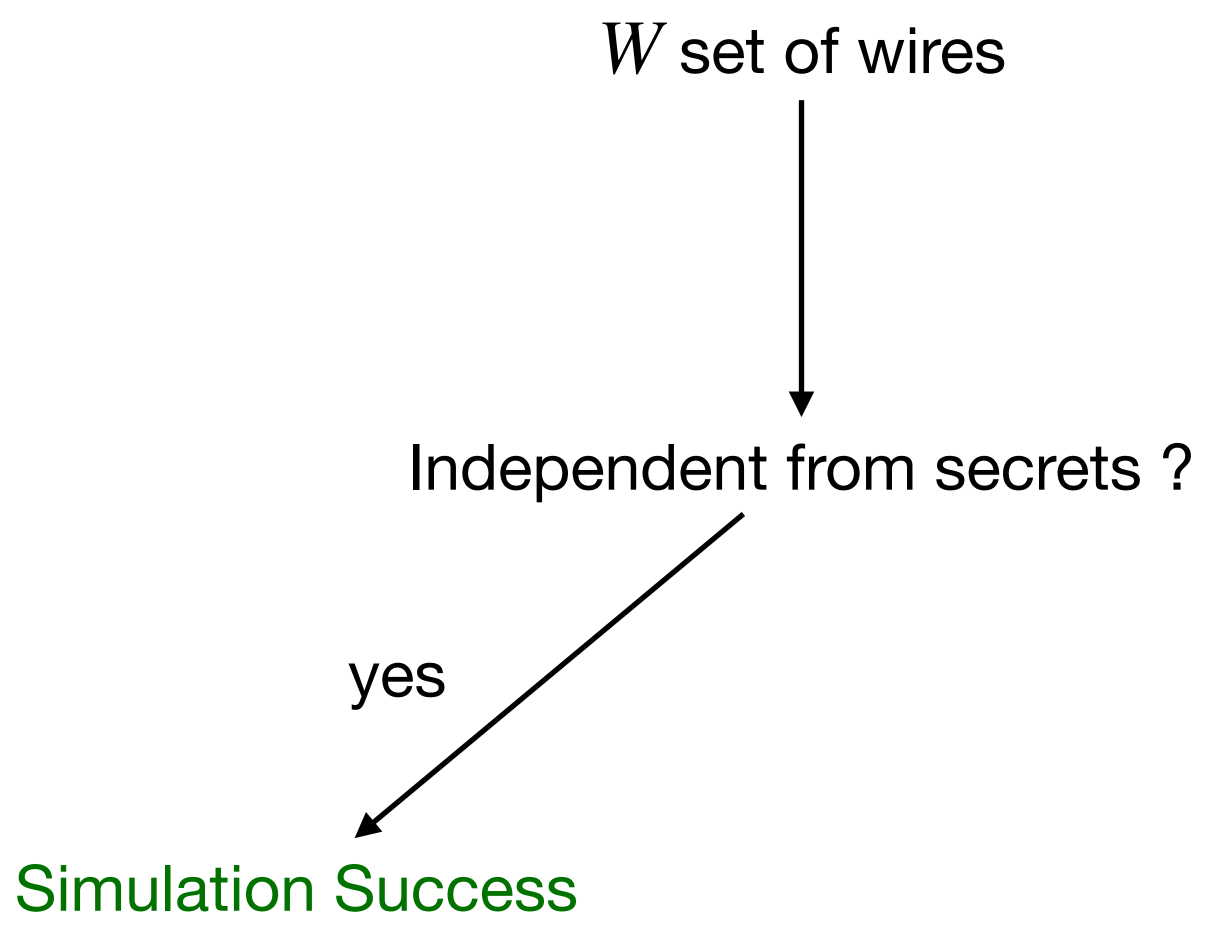
Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

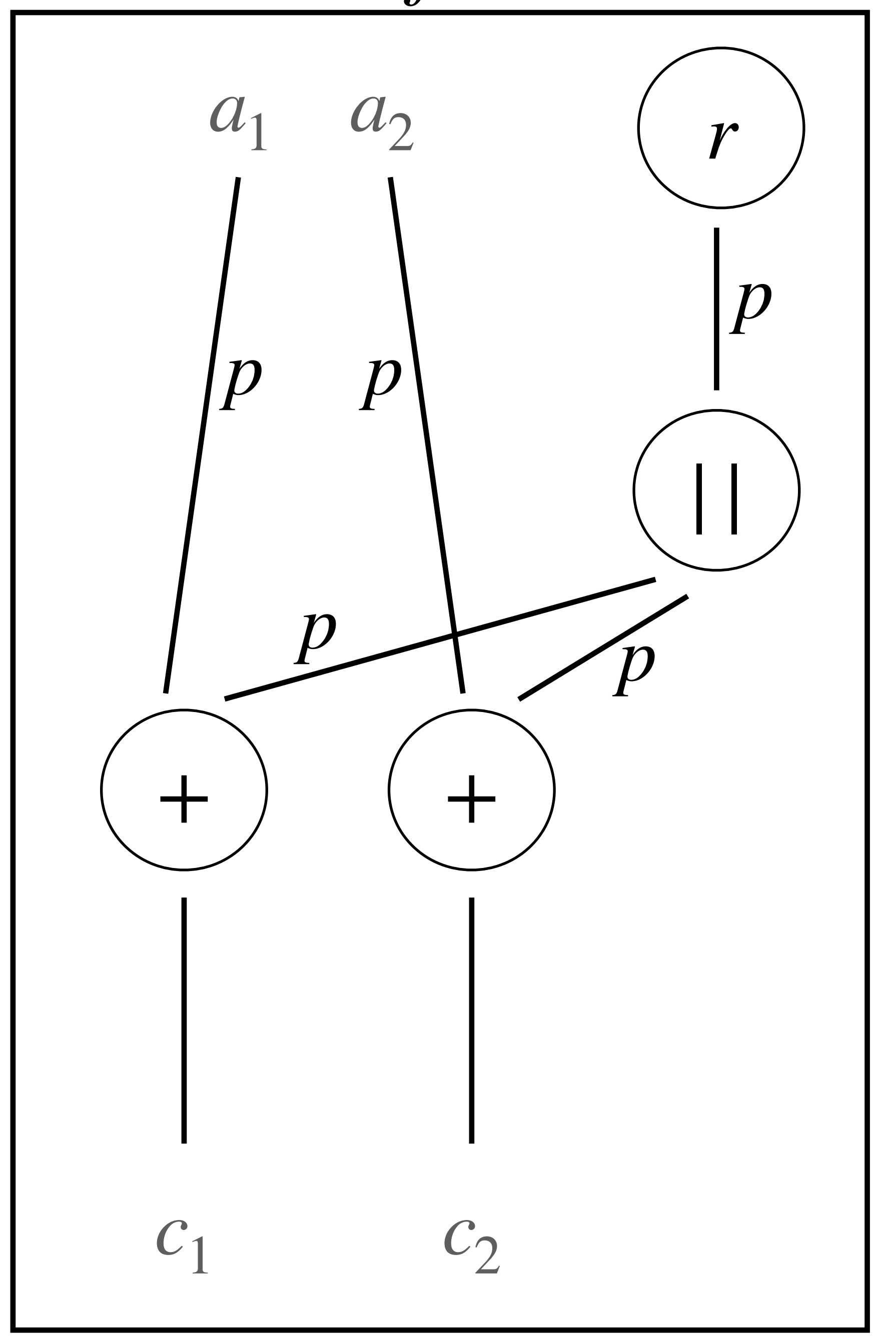


Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

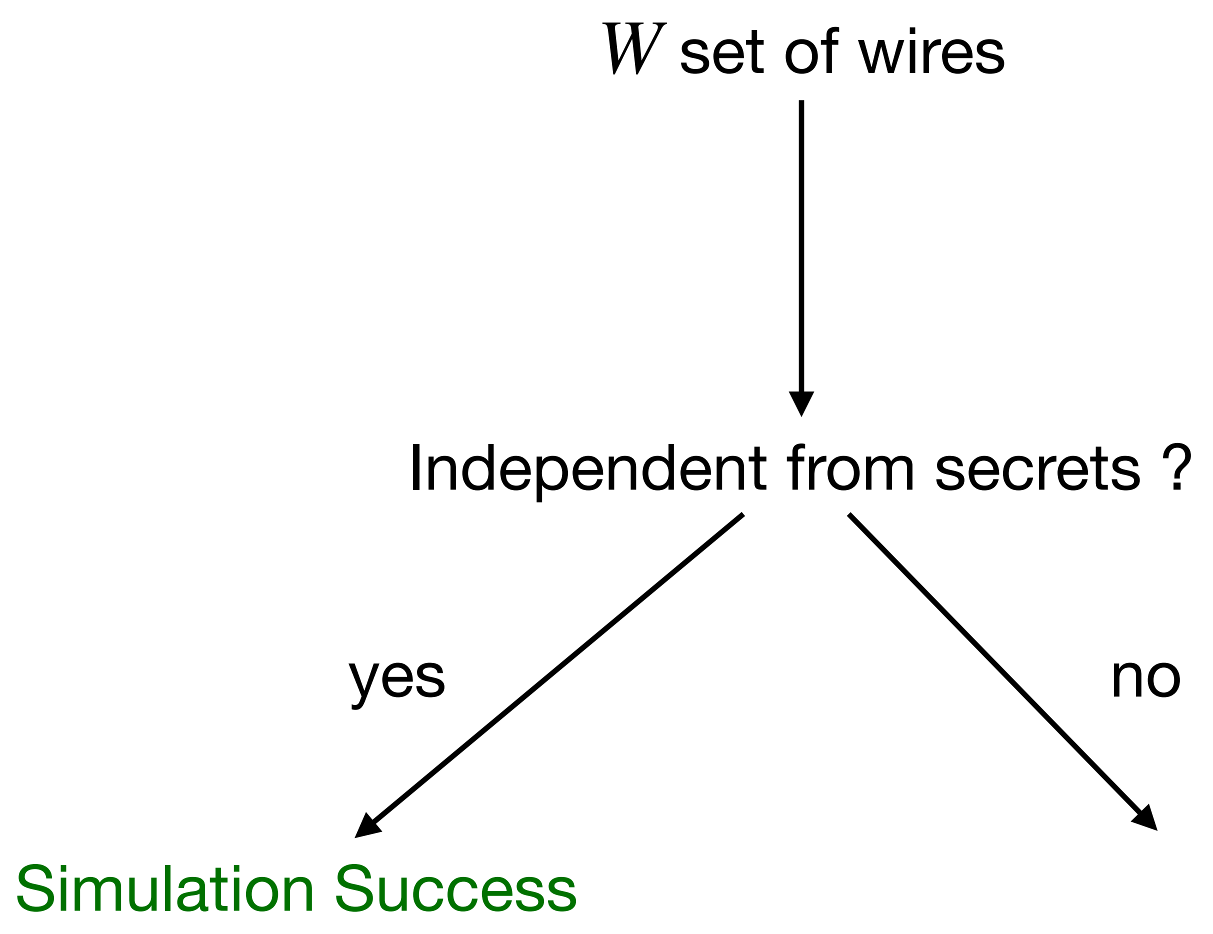
Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

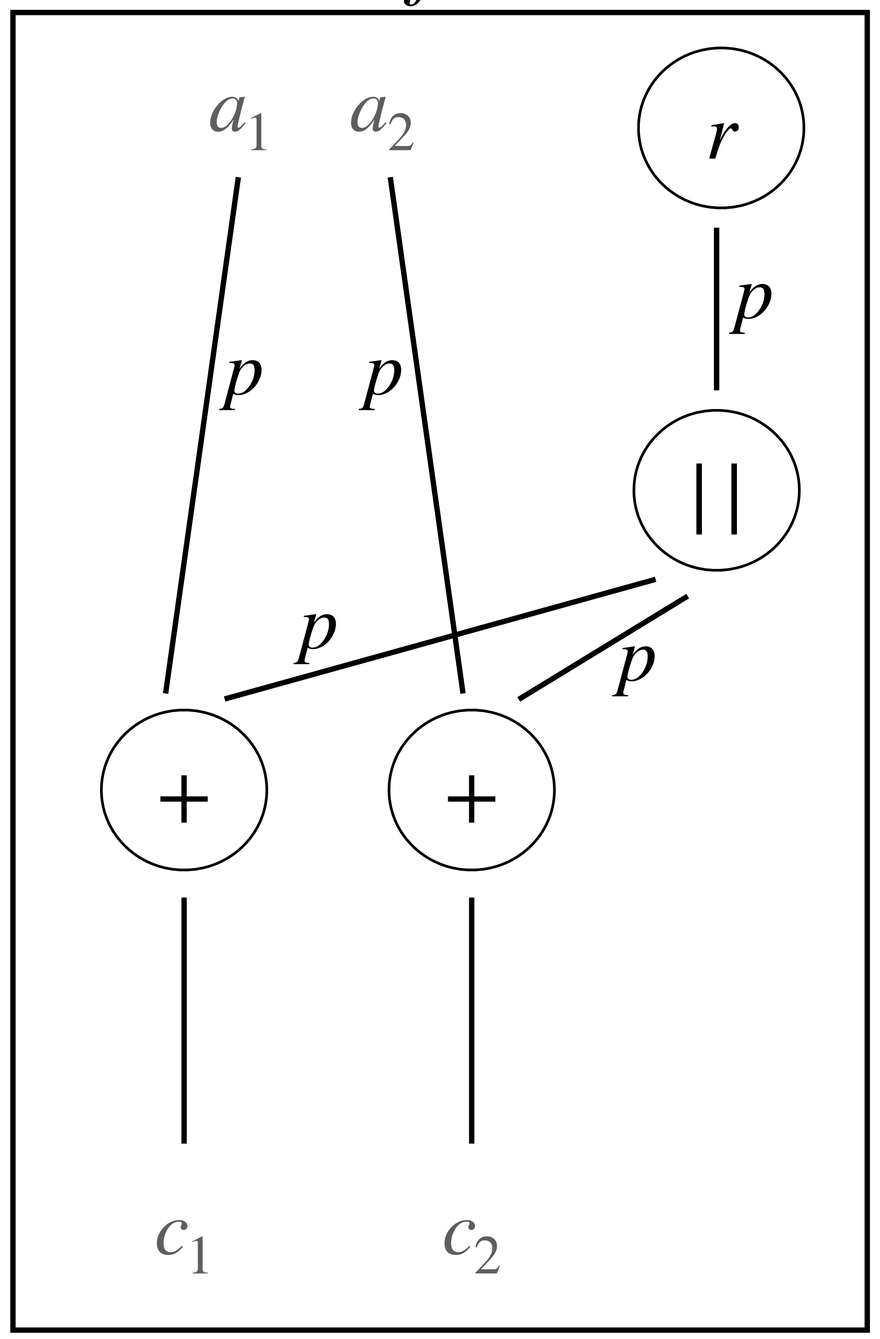


Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

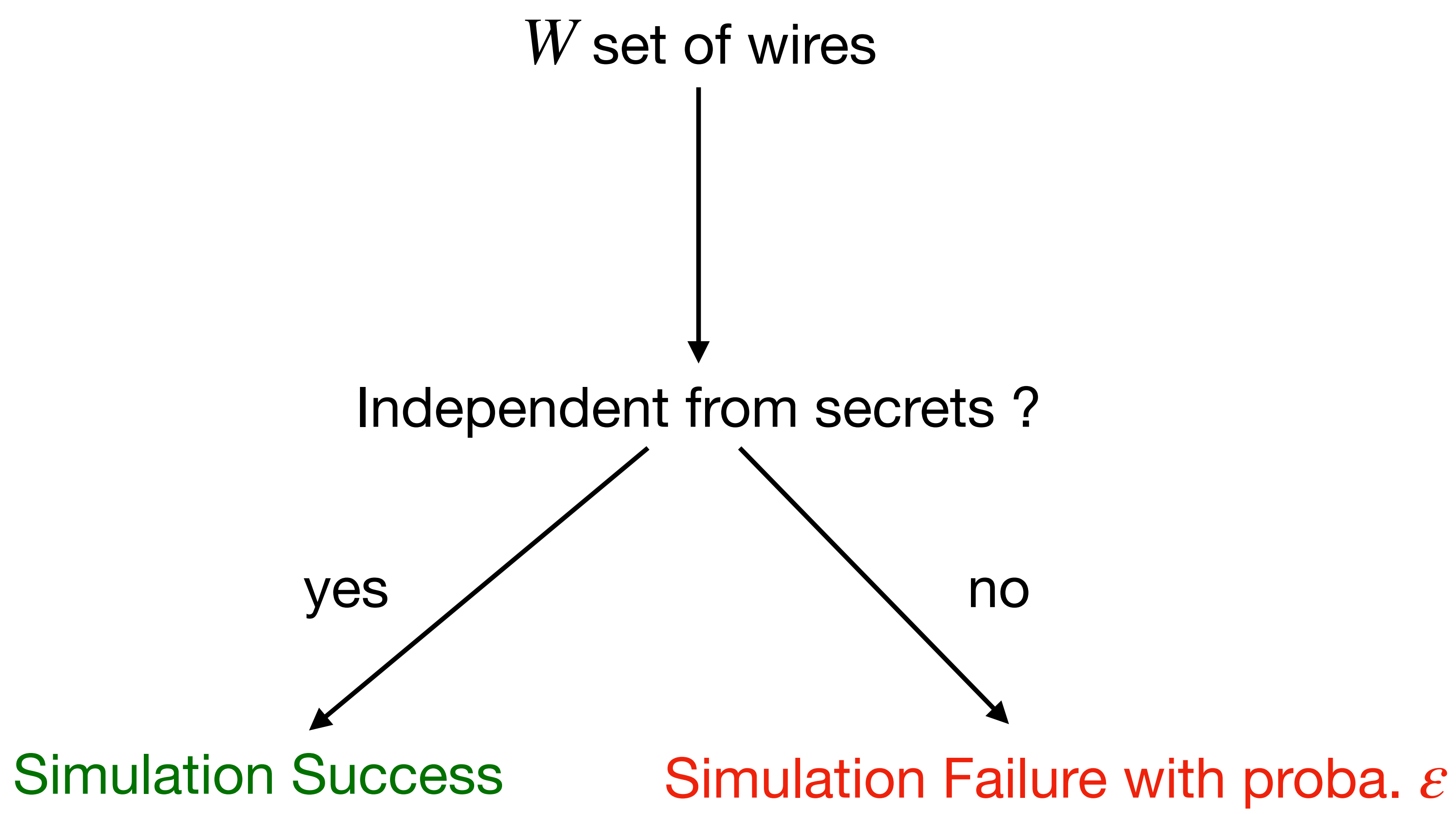
Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

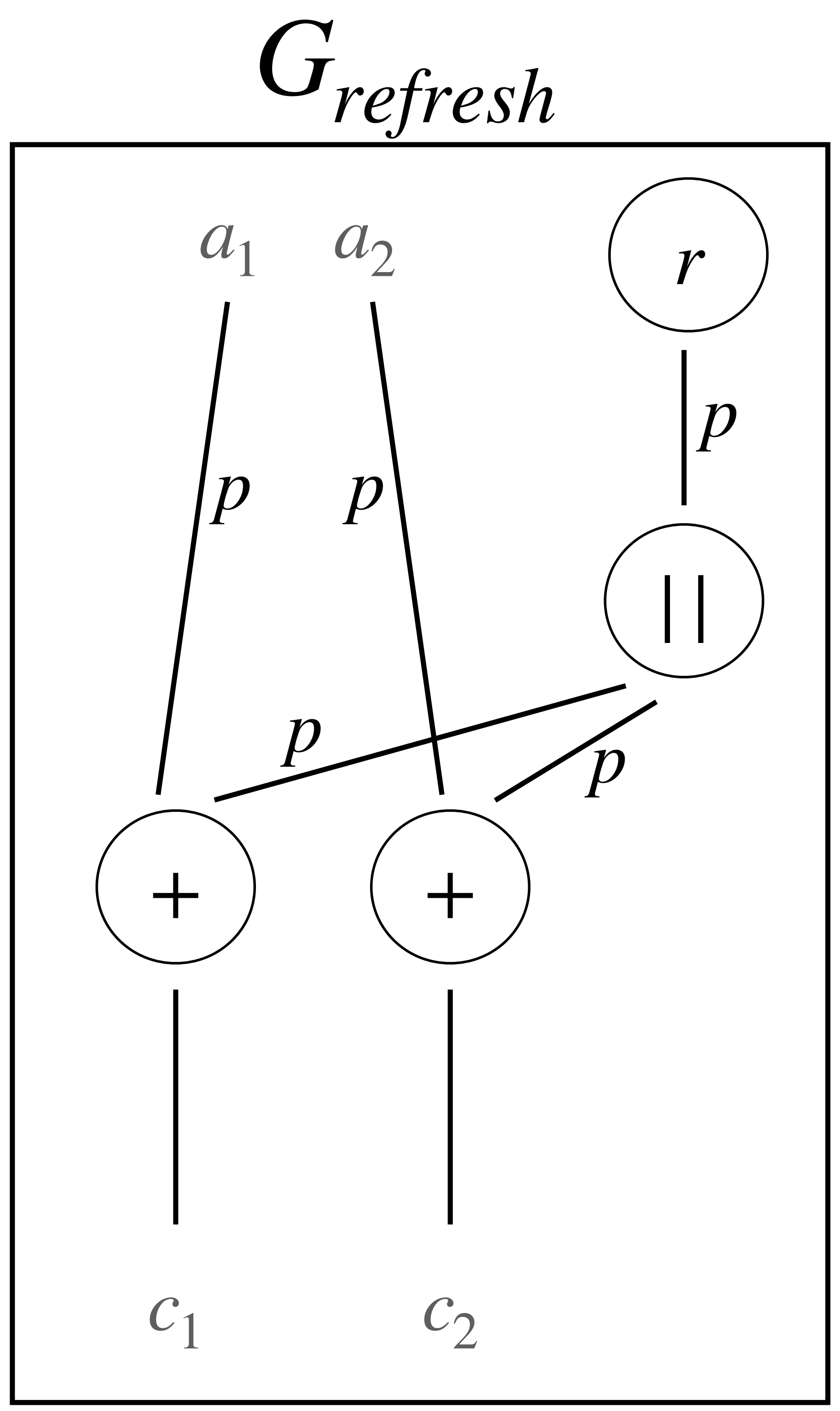
(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

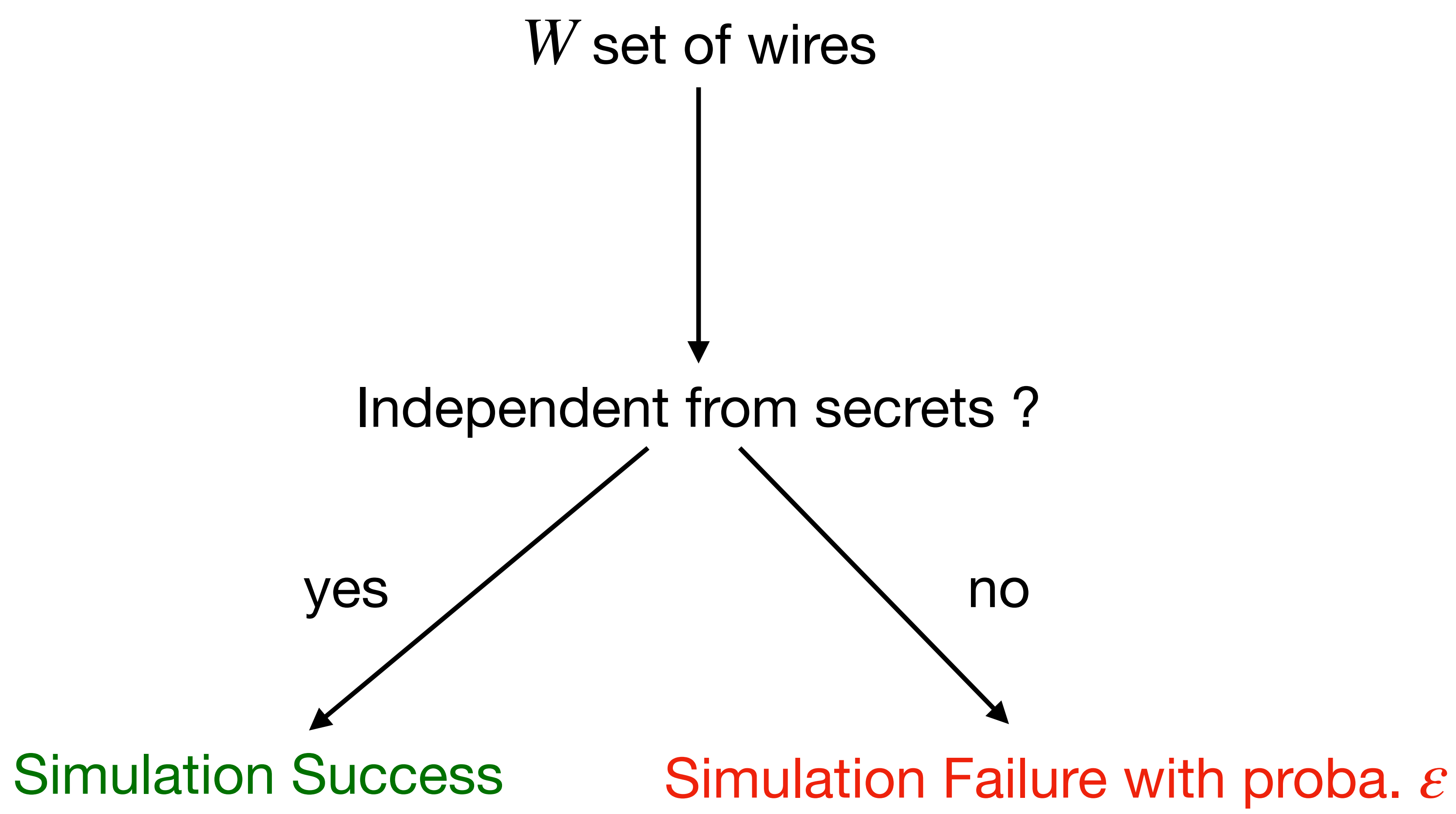
Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

Examples

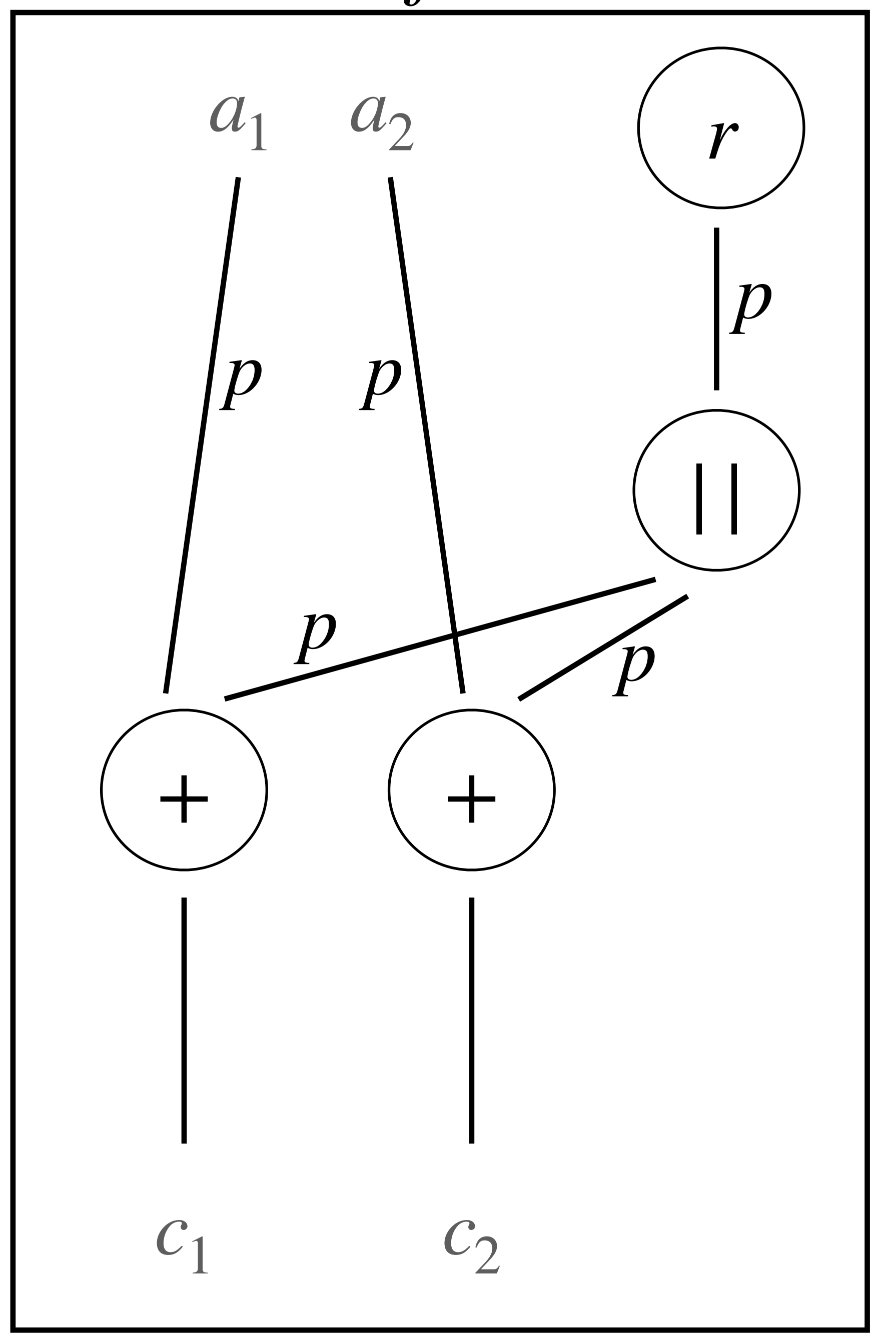


Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

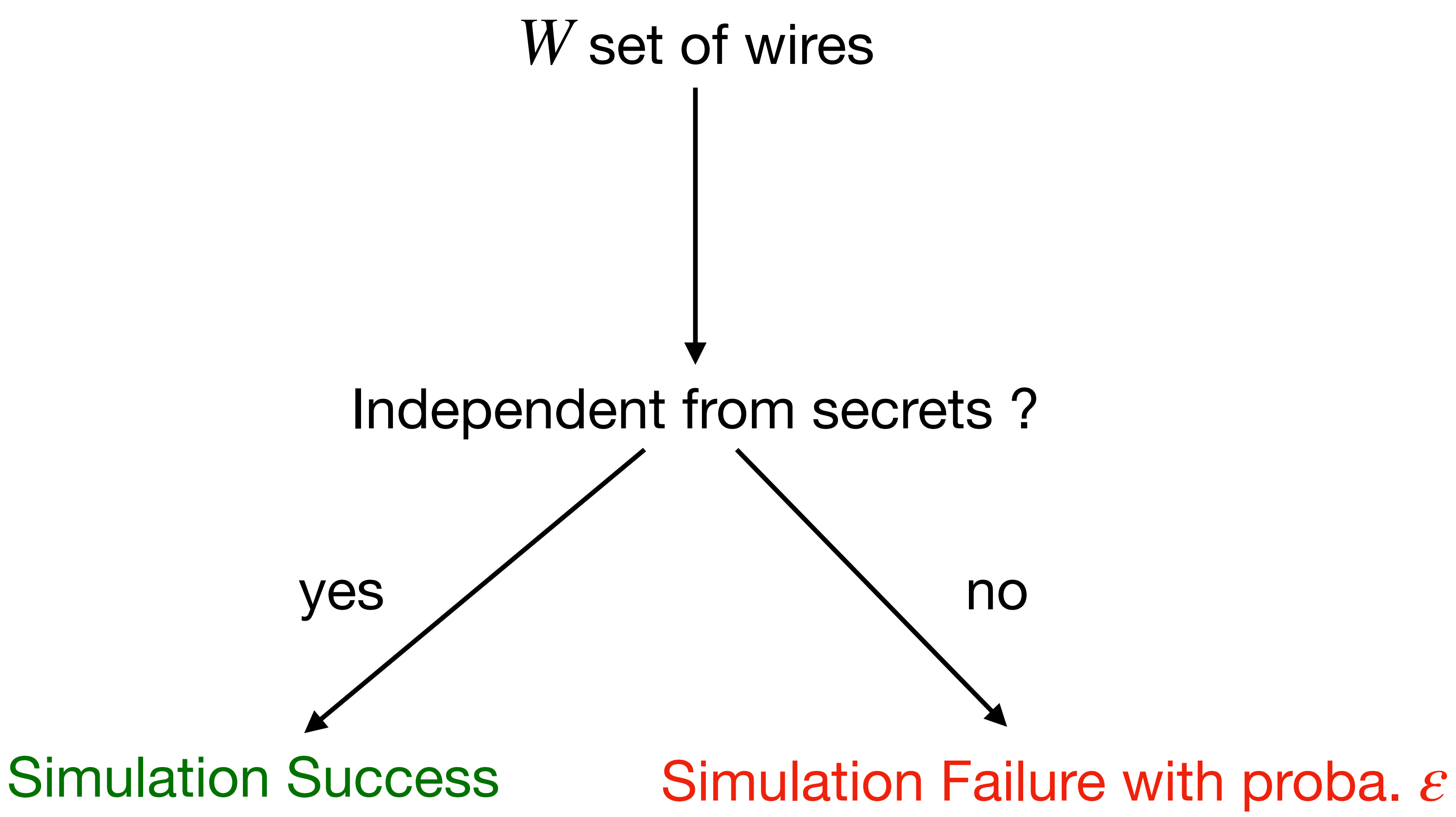
Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

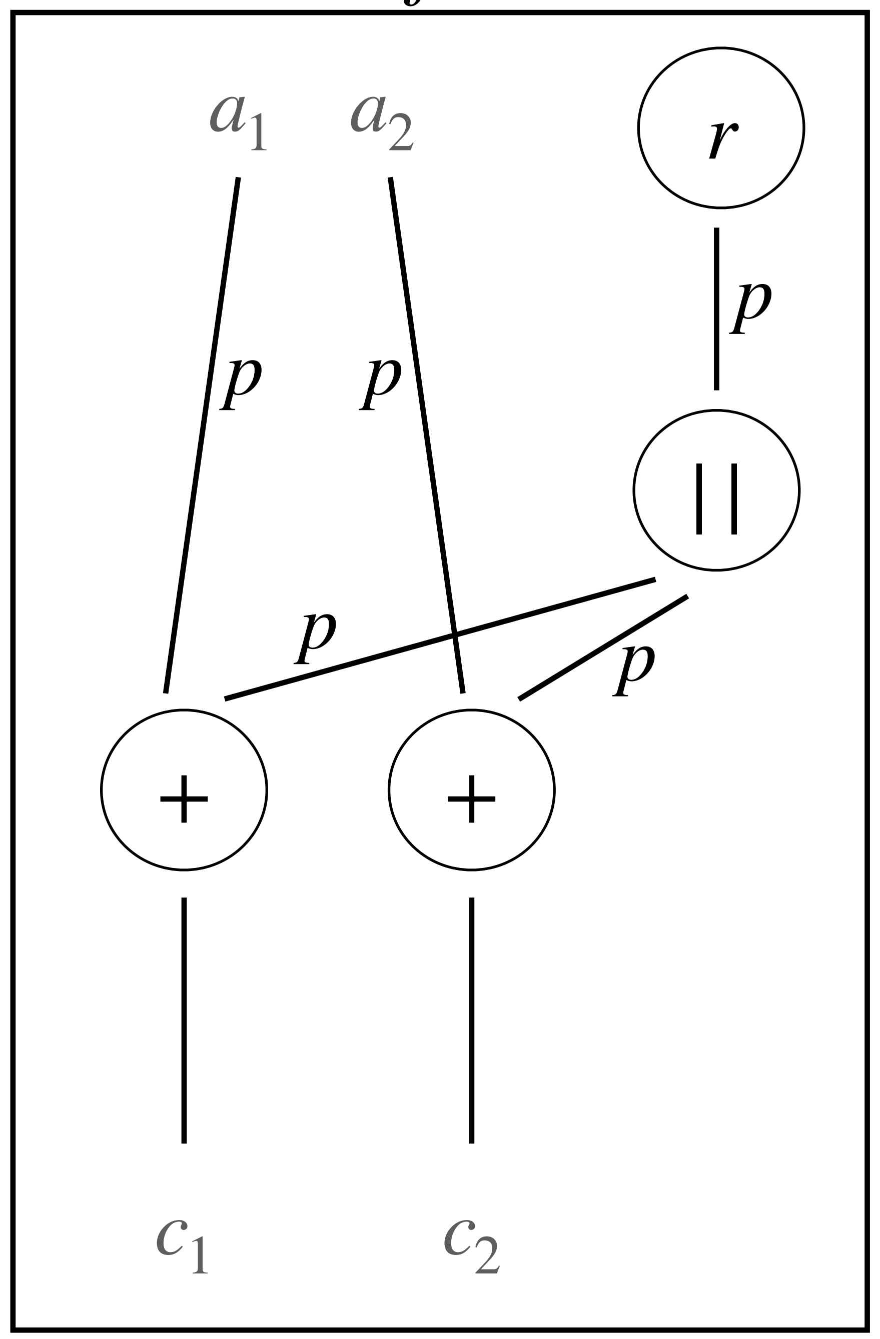
Examples

$\{a_1\}$

Random Probing Security

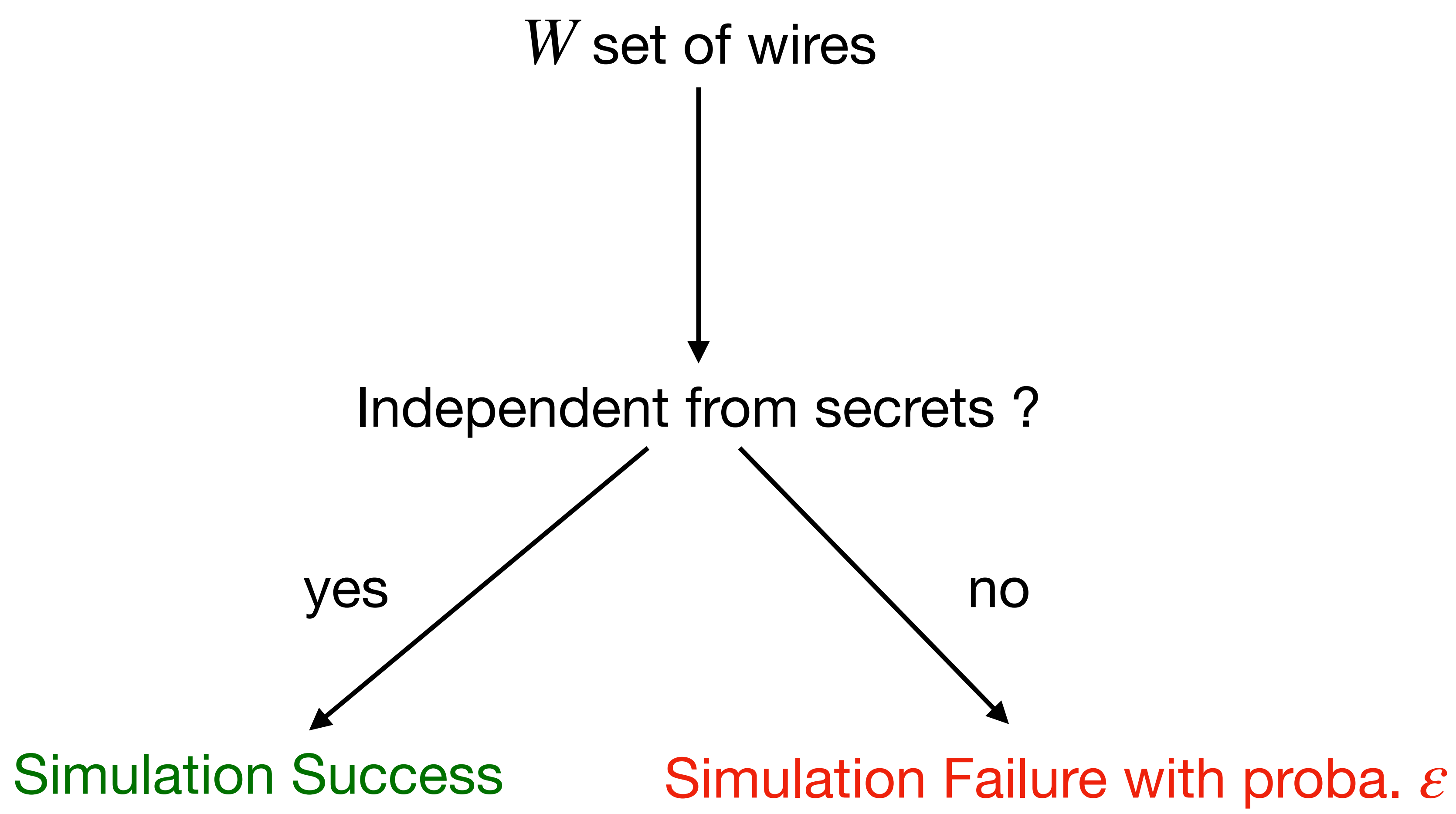
Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security



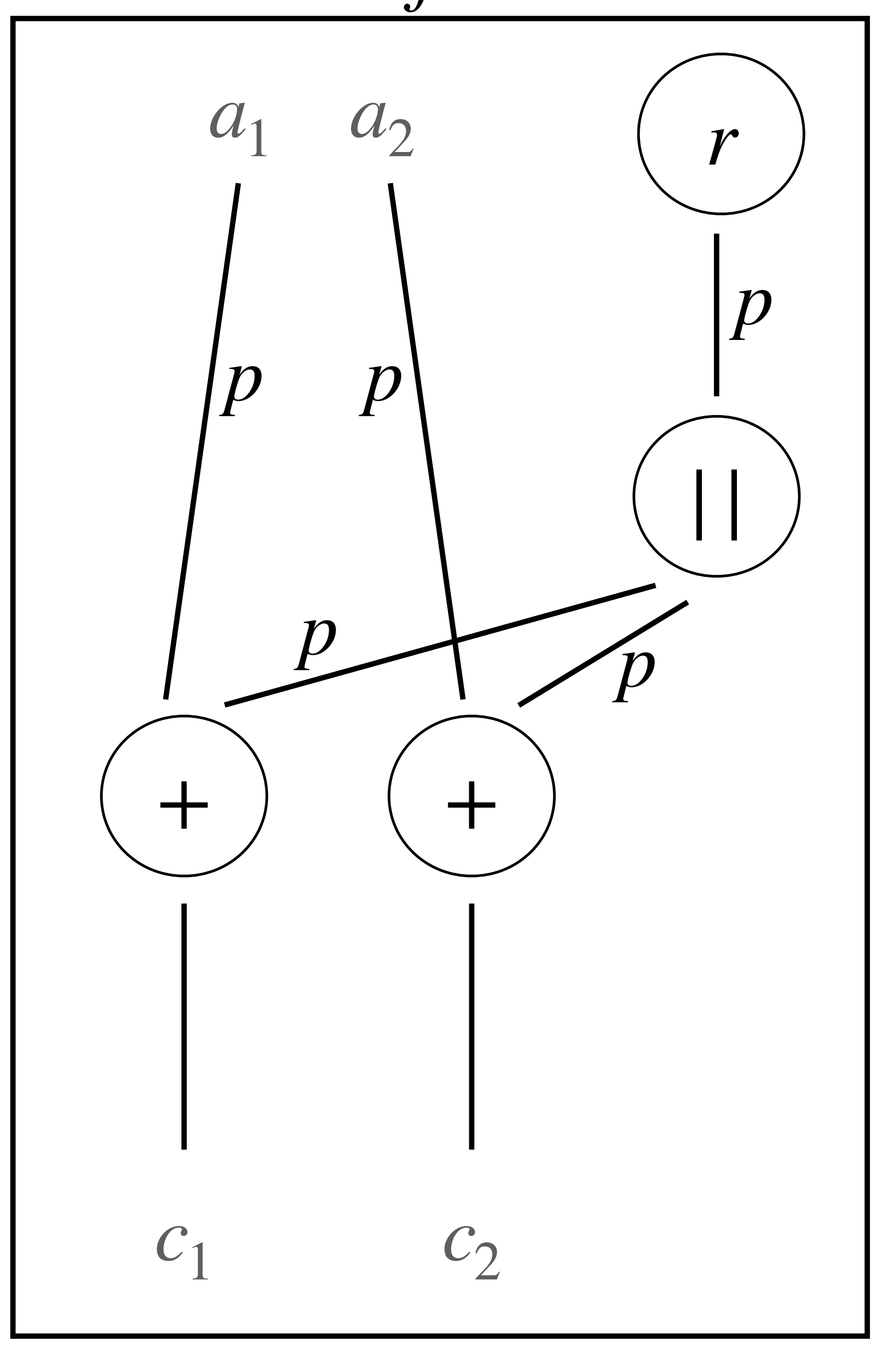
Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

Success $\{a_1\}$

Random Probing Security
Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

W set of wires

Independent from secrets ?

yes

no

Simulation Success

Simulation Failure with proba. ε

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

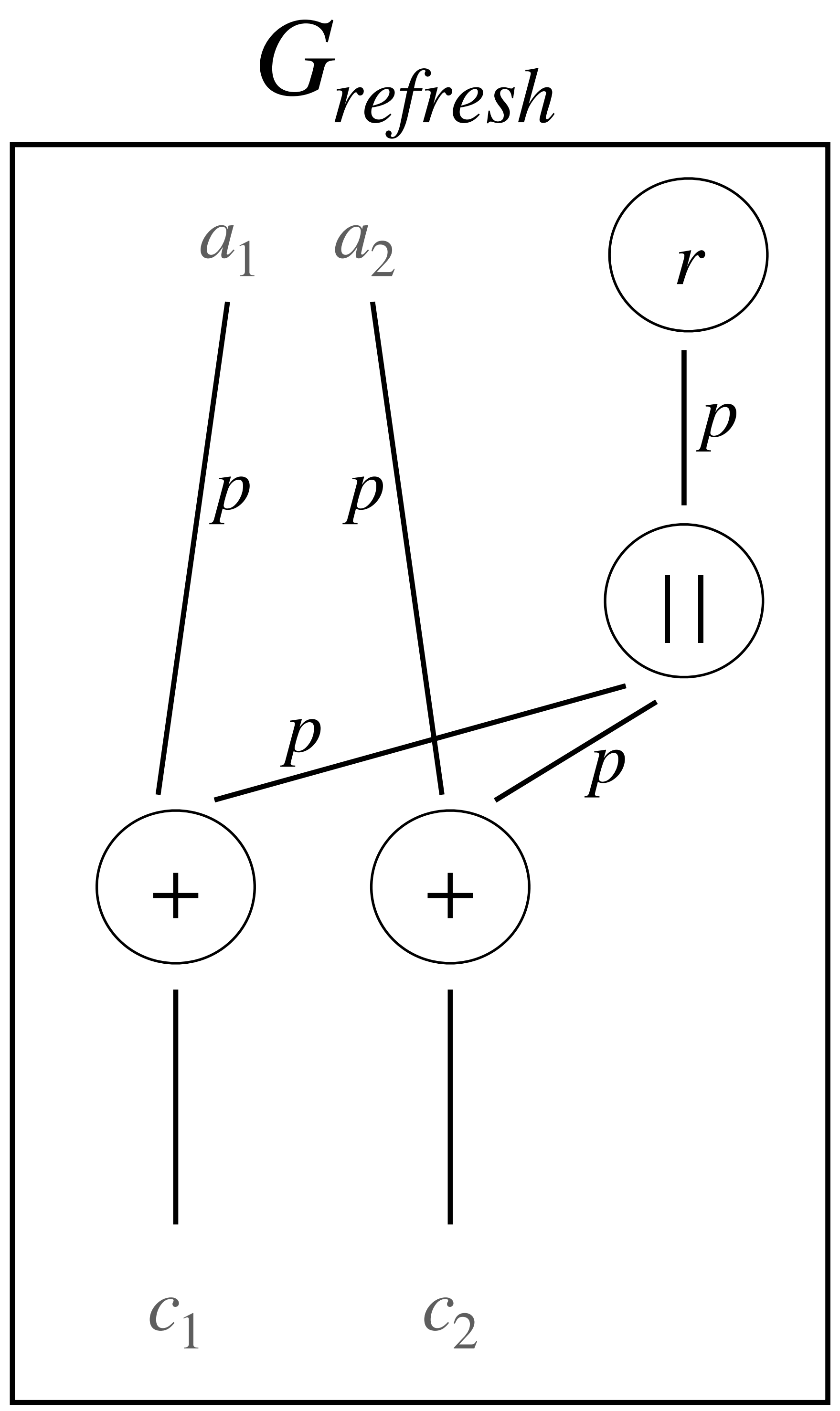
Success

$\{a_1\}$

$\{a_2, r\}$

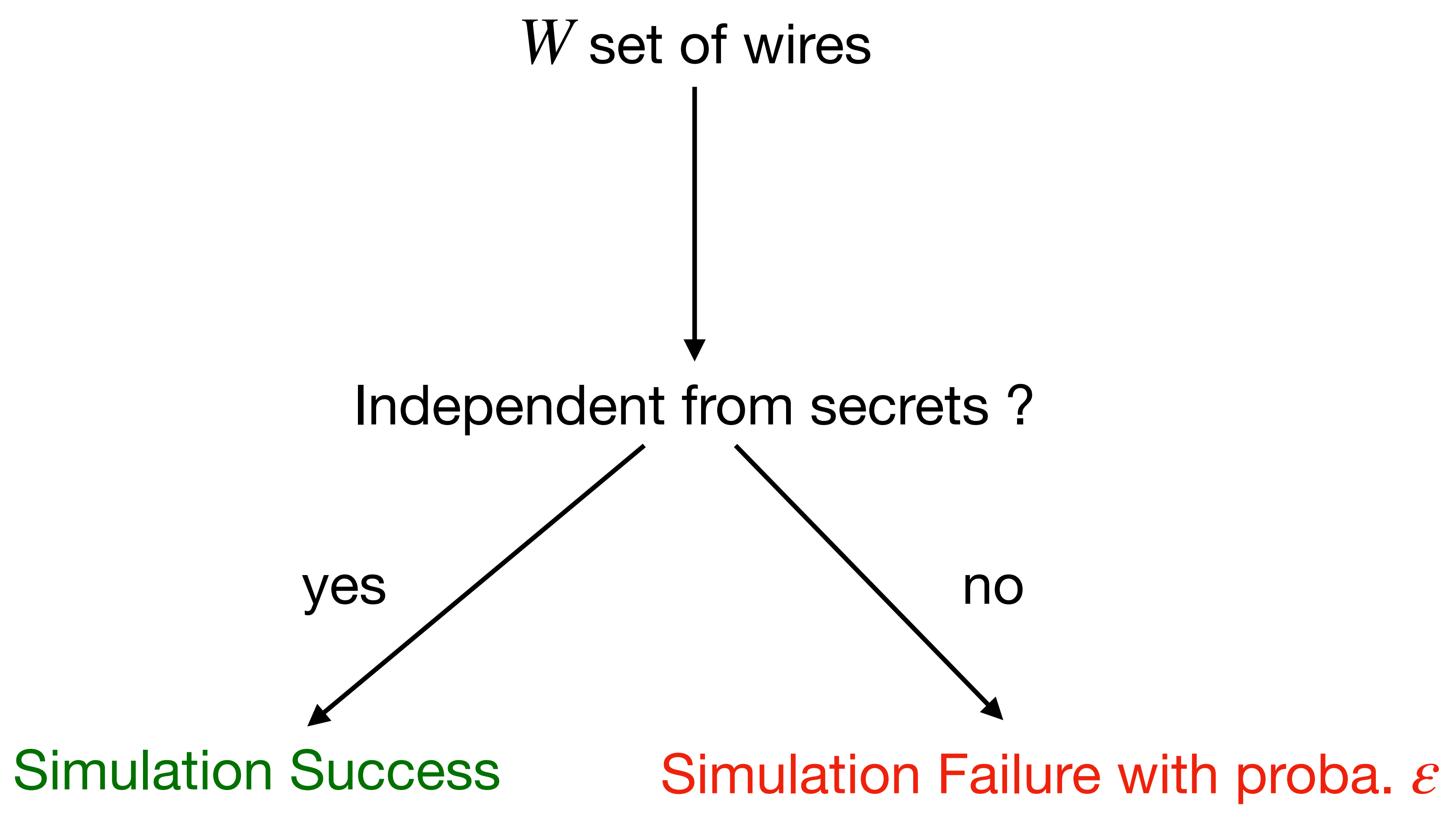
Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

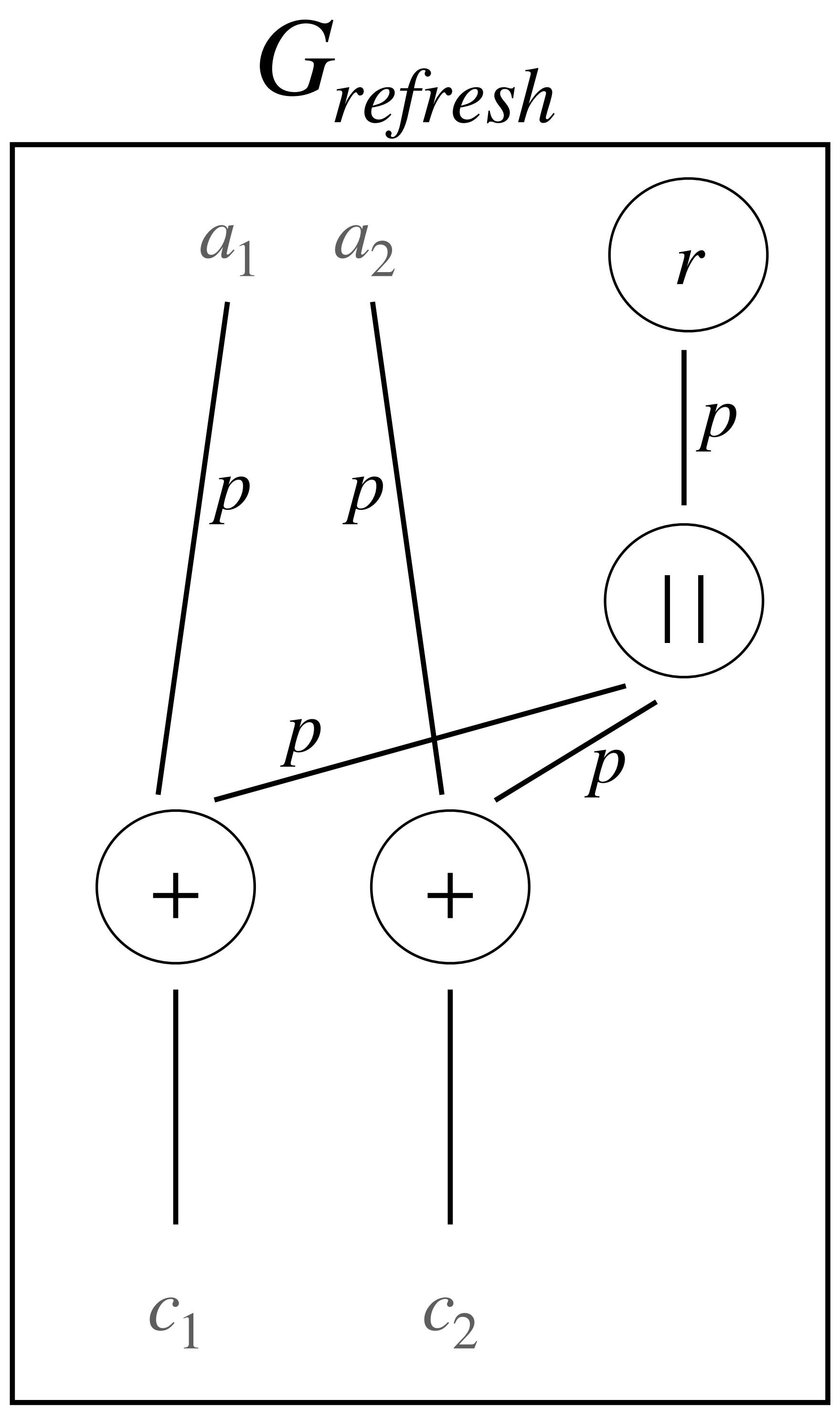
Examples

Success $\{a_1\}$

Success $\{a_2, r\}$

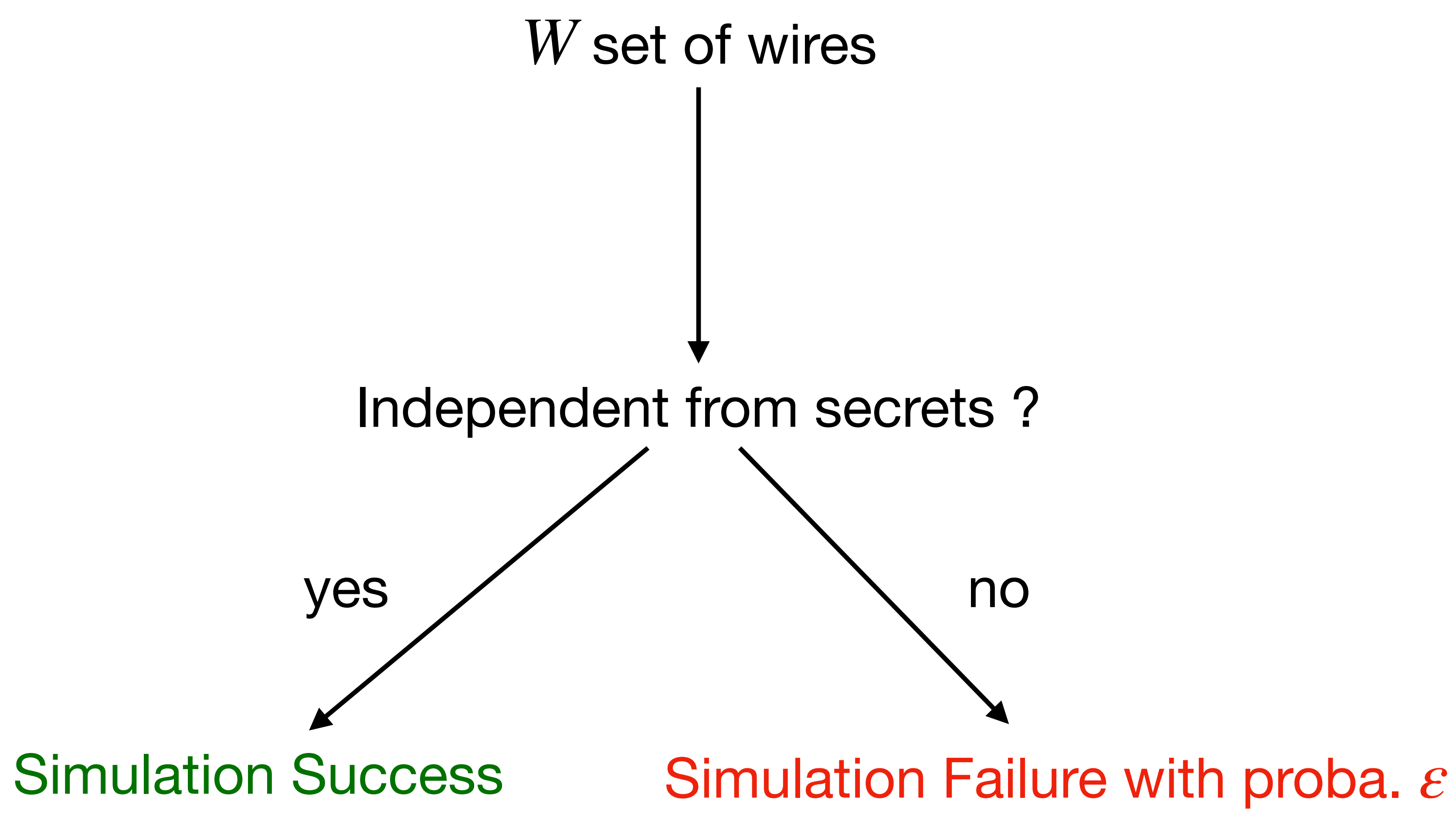
Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

Success $\{a_1\}$

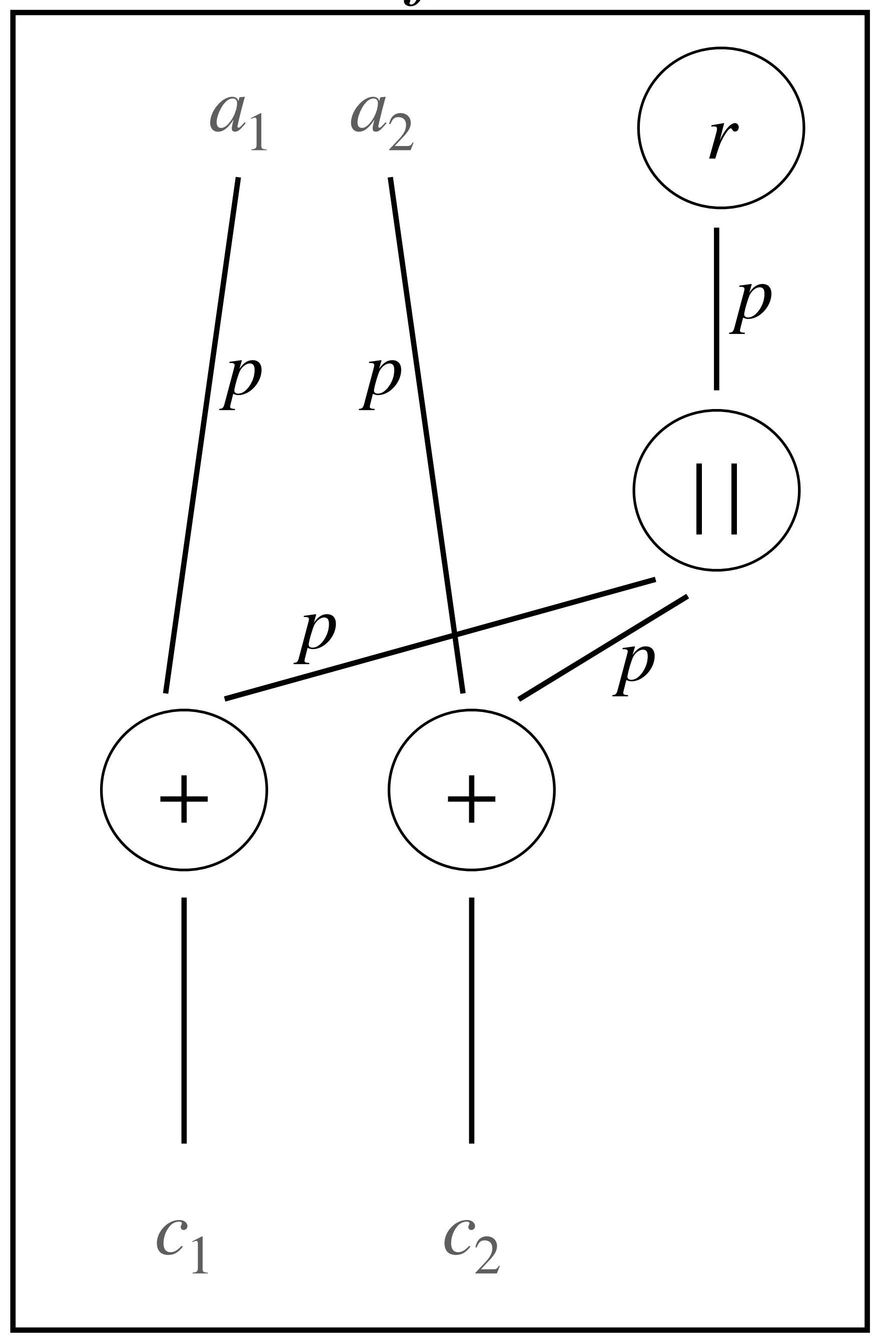
Success $\{a_2, r\}$

$\{a_1, a_2\}$

Random Probing Security

Definition

$G_{refresh}$



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

W set of wires

↓

Independent from secrets ?

yes → Simulation Success

no → Simulation Failure with proba. ε

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

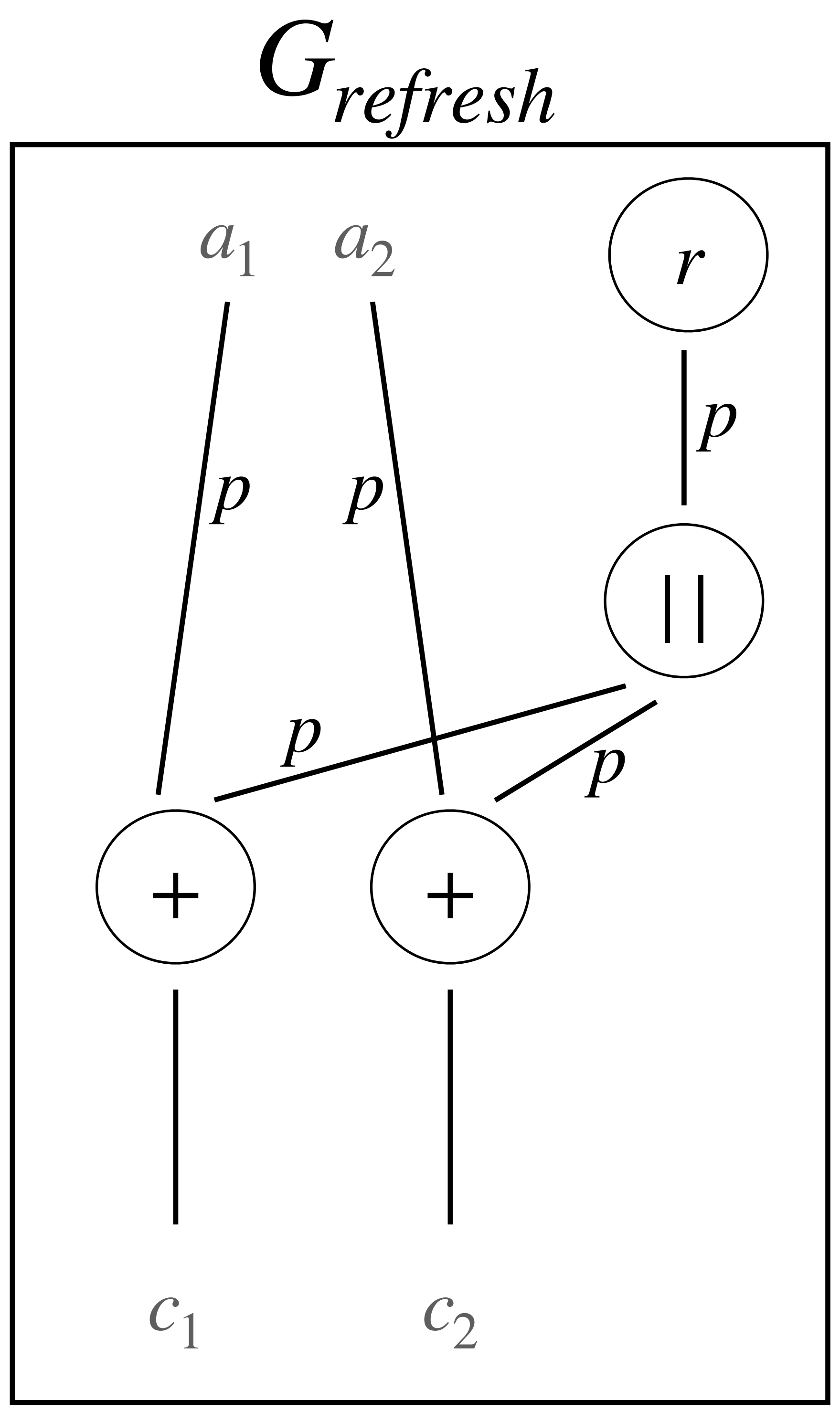
Success $\{a_1\}$

Success $\{a_2, r\}$

Failure $\{a_1, a_2\}$

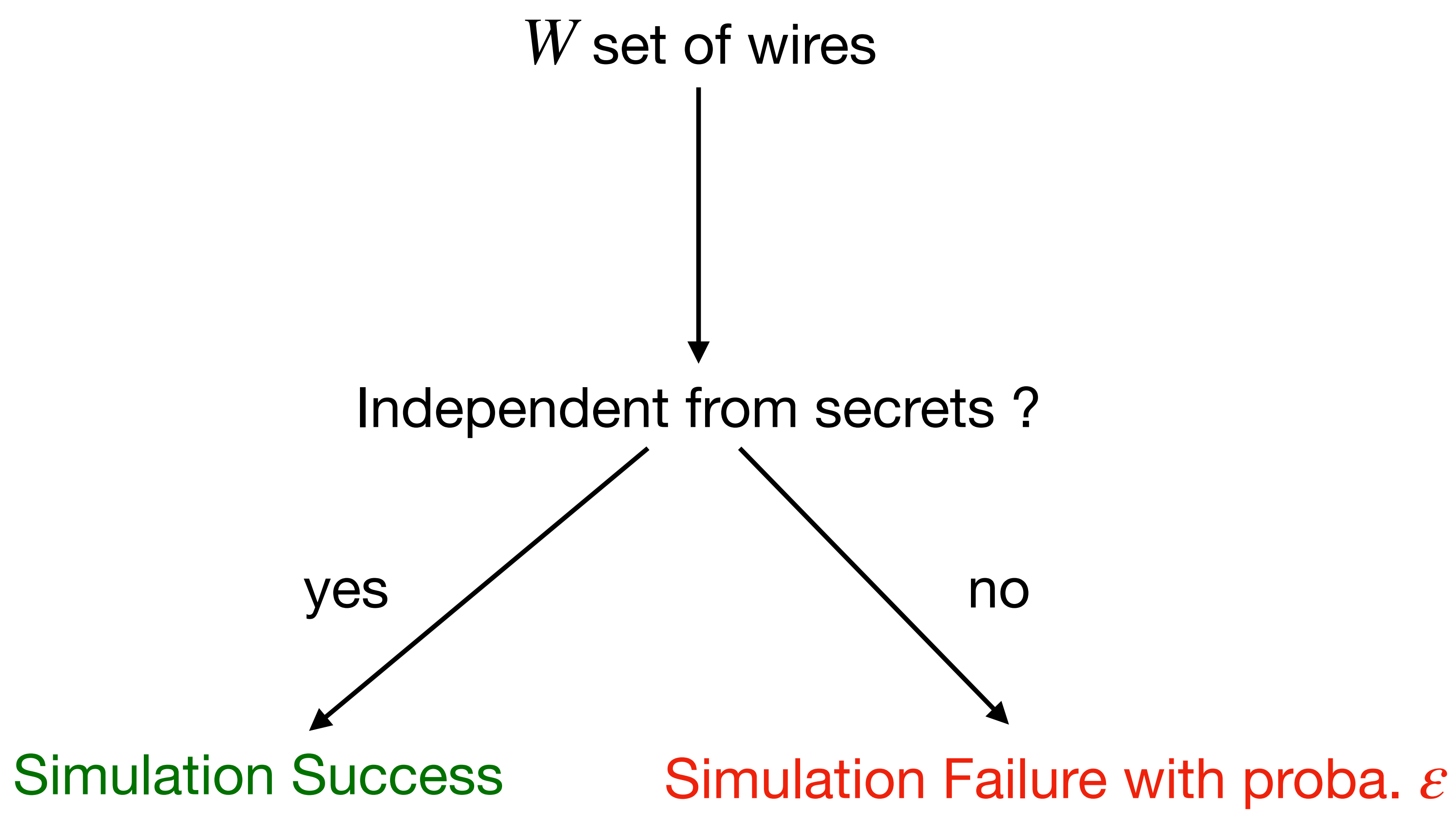
Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security



Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

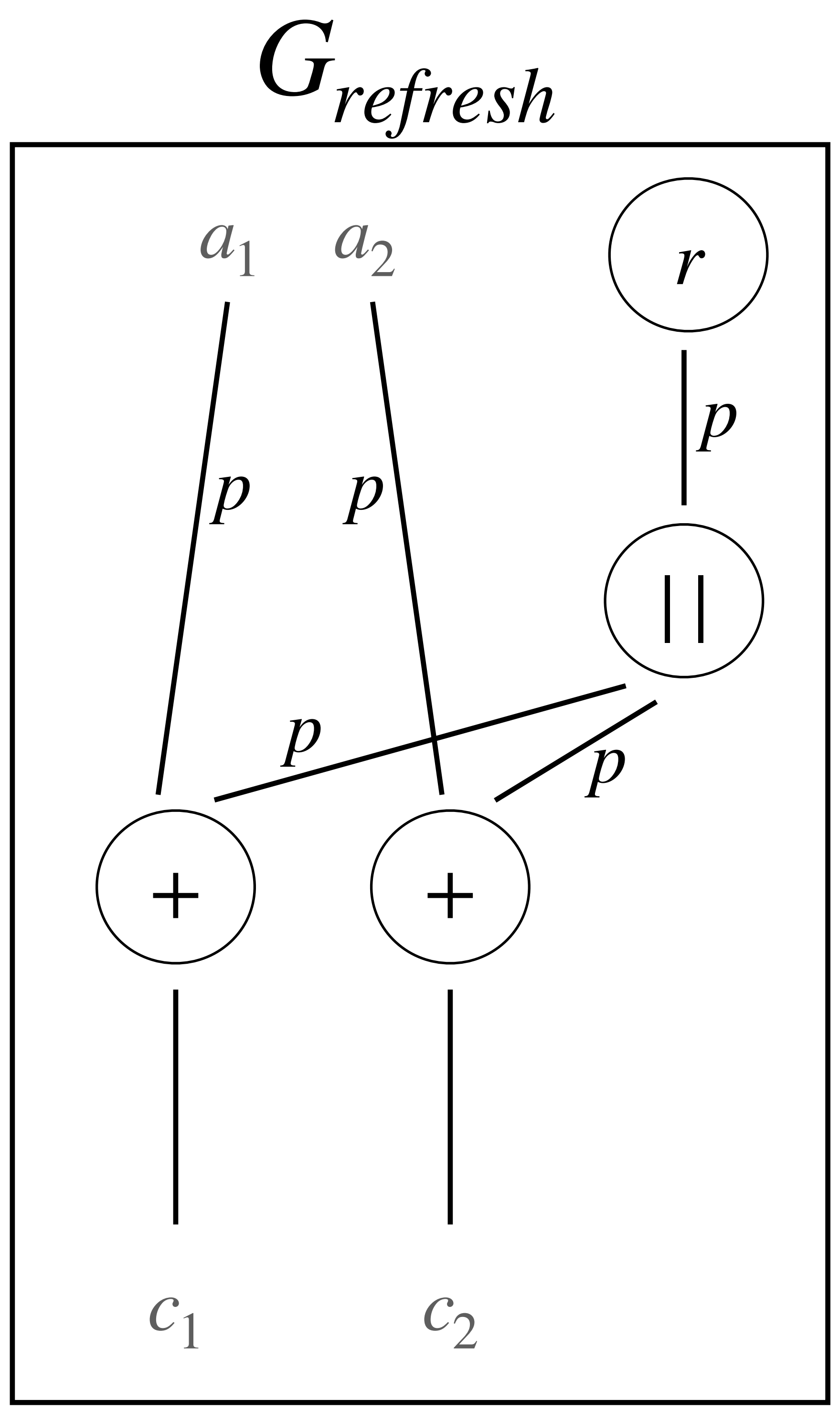
Success $\{a_1\}$ $Pr(\{a_1\}) = p(1-p)^4$

Success $\{a_2, r\}$ $Pr(\{a_2, r\}) = p^2(1-p)^3$

Failure $\{a_1, a_2\}$ $Pr(\{a_1, a_2\}) = p^2(1-p)^3$

Random Probing Security

Definition



Choice: no leak on output shares, inputs of the next circuit

(p, ε) – random probing security

$f(p)$

W set of wires

Independent from secrets ?

yes → Simulation Success

no → Simulation Failure with proba. ε

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Examples

Success $\{a_1\}$ $Pr(\{a_1\}) = p(1-p)^4$

Success $\{a_2, r\}$ $Pr(\{a_2, r\}) = p^2(1-p)^3$

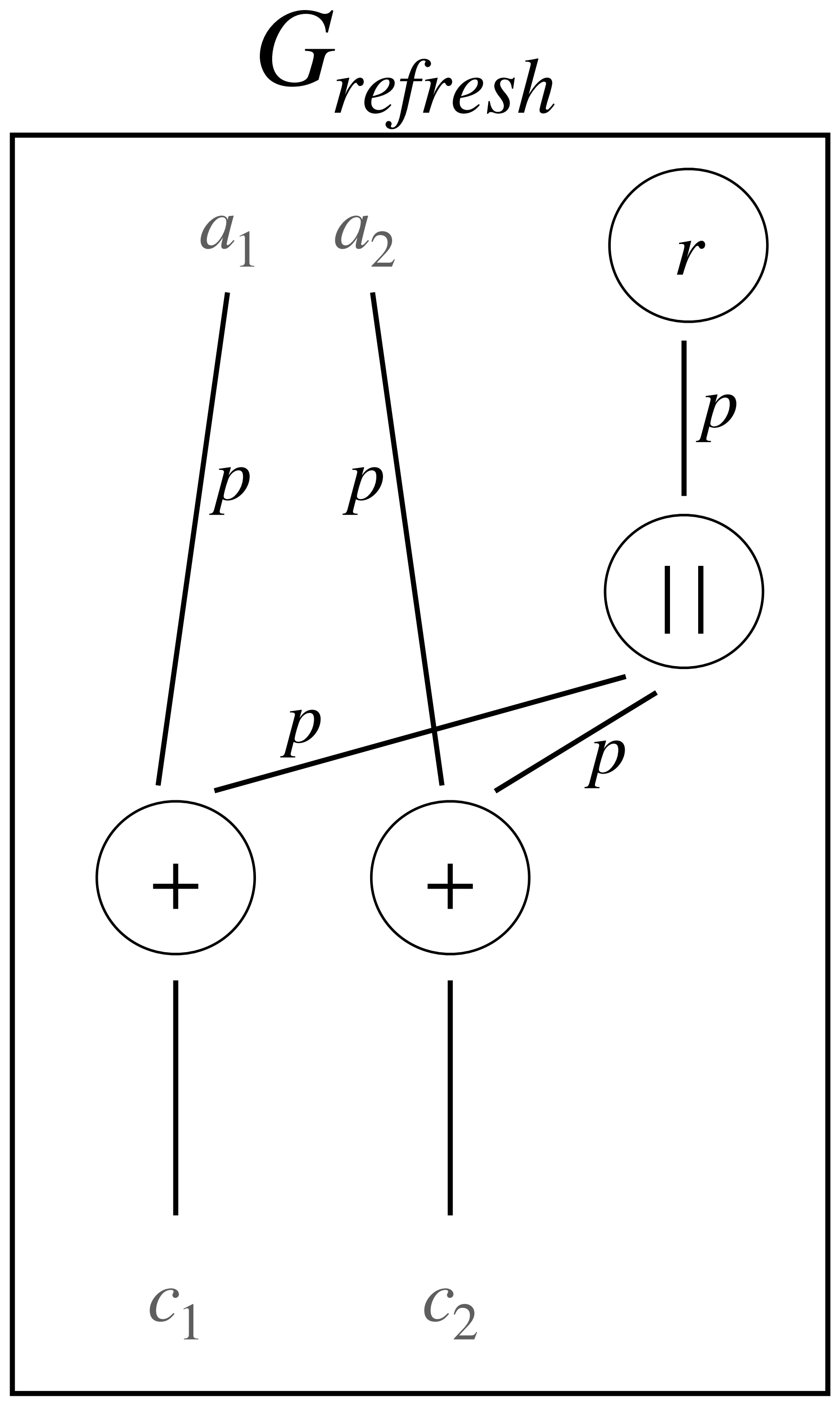
Failure $\{a_1, a_2\}$ $Pr(\{a_1, a_2\}) = p^2(1-p)^3$

Random Probing Security

Definition: how to compute ε ?

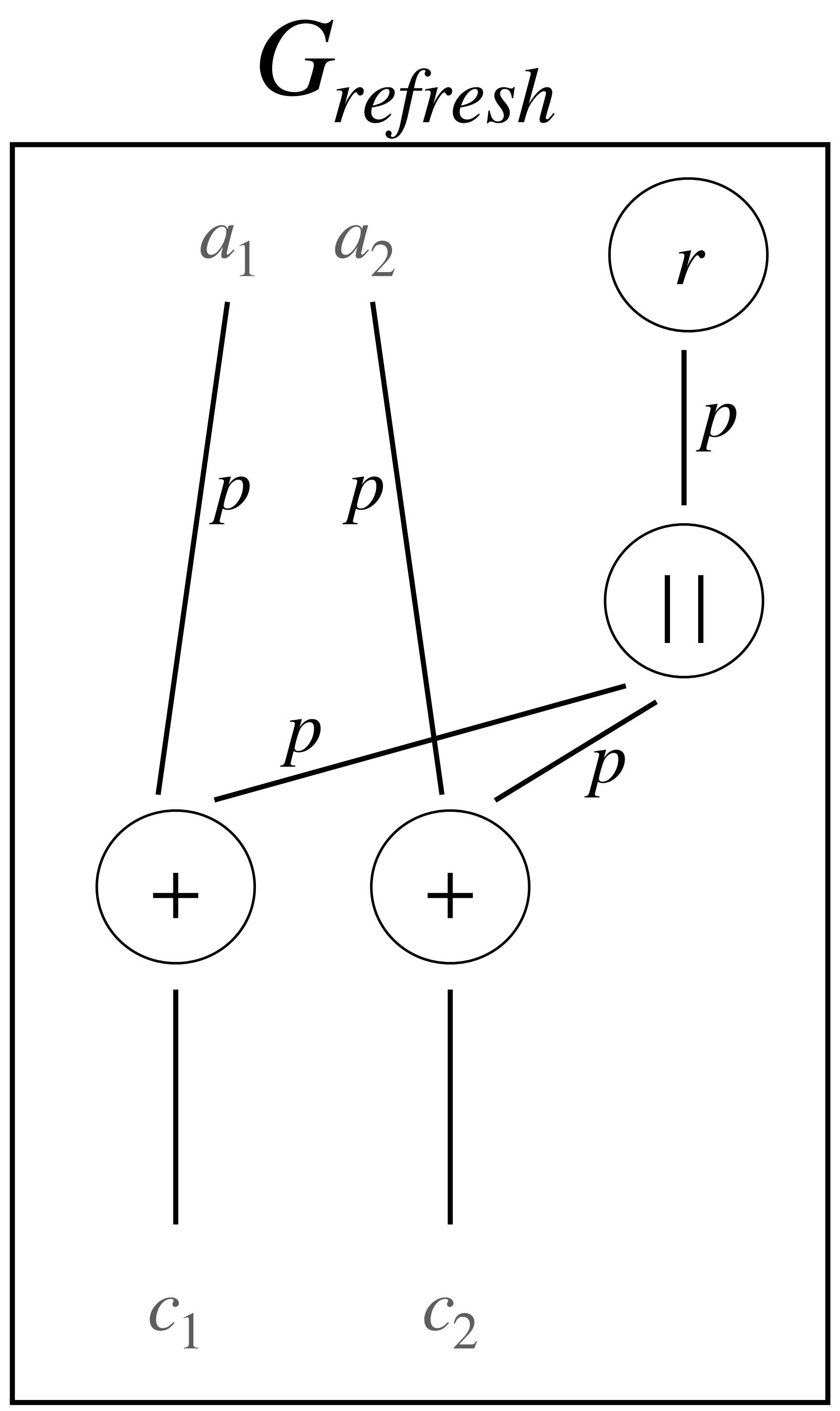
Random Probing Security

Definition: how to compute ε ?



Random Probing Security

Definition: how to compute ε ?



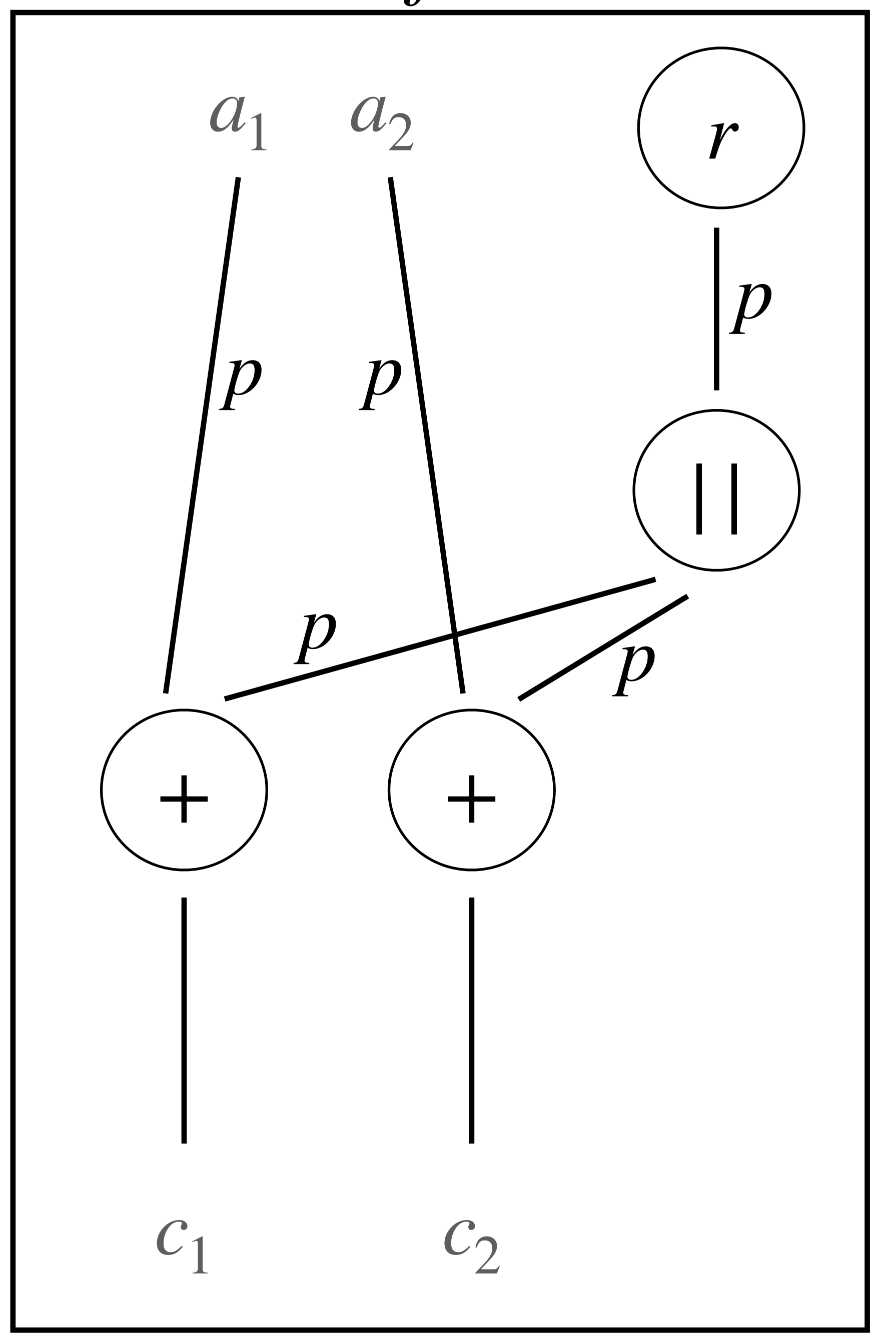
(p, ε) – random probing security

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?

$G_{refresh}$



(p, ε) – random probing security

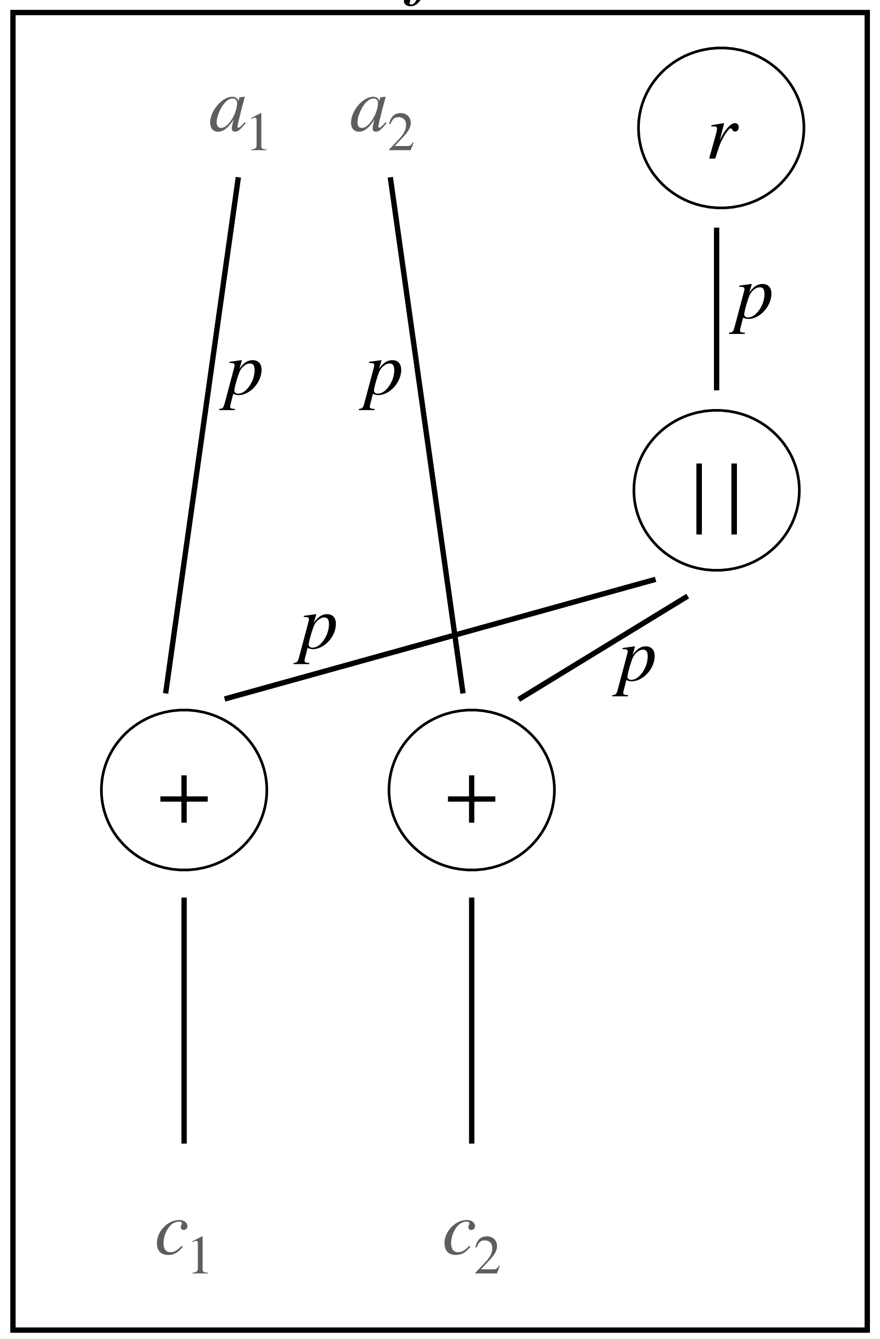
s: number of wires in gadget

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?

$G_{refresh}$



(p, ε) – random probing security

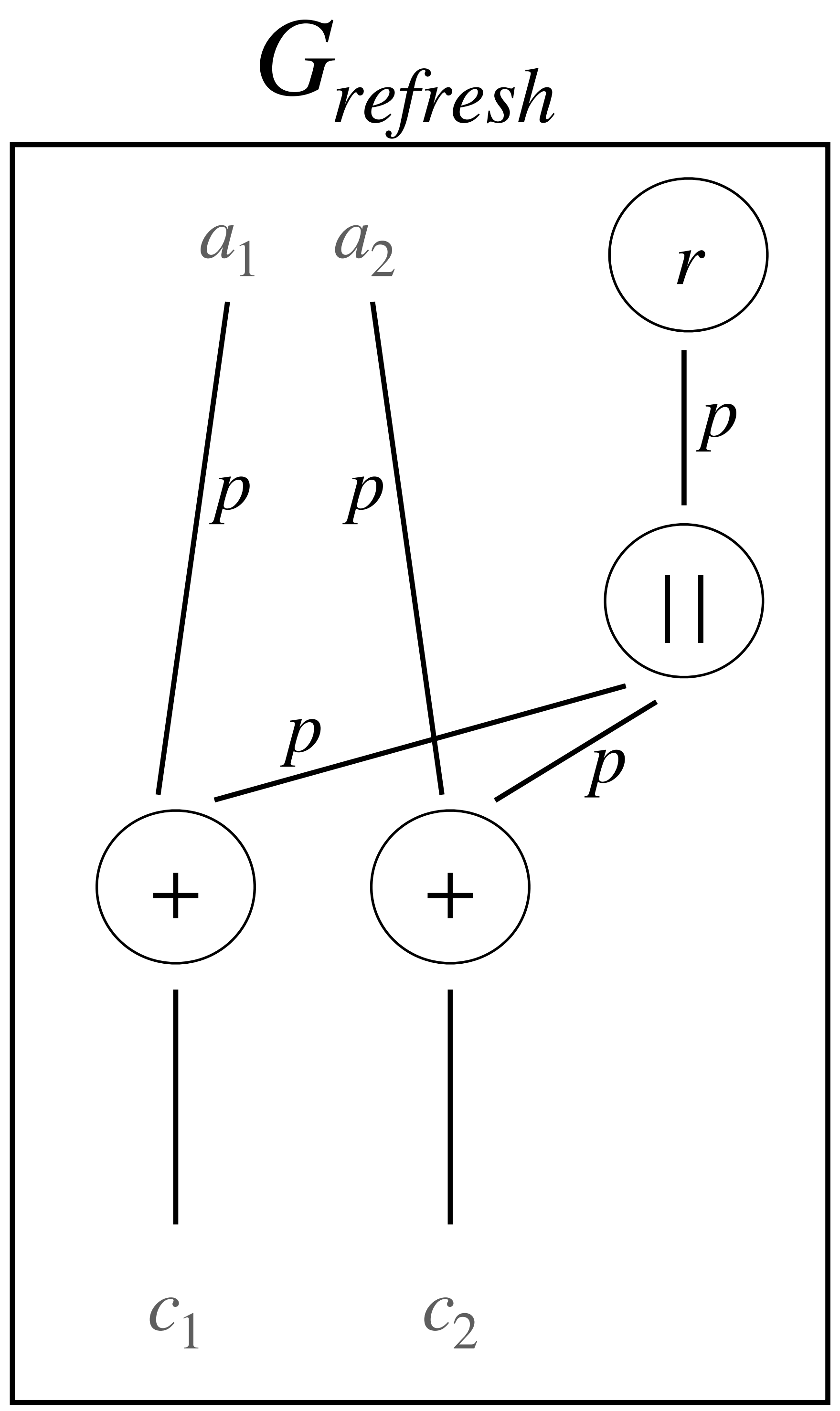
s: number of wires in gadget

$\varepsilon \leftarrow 0$

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



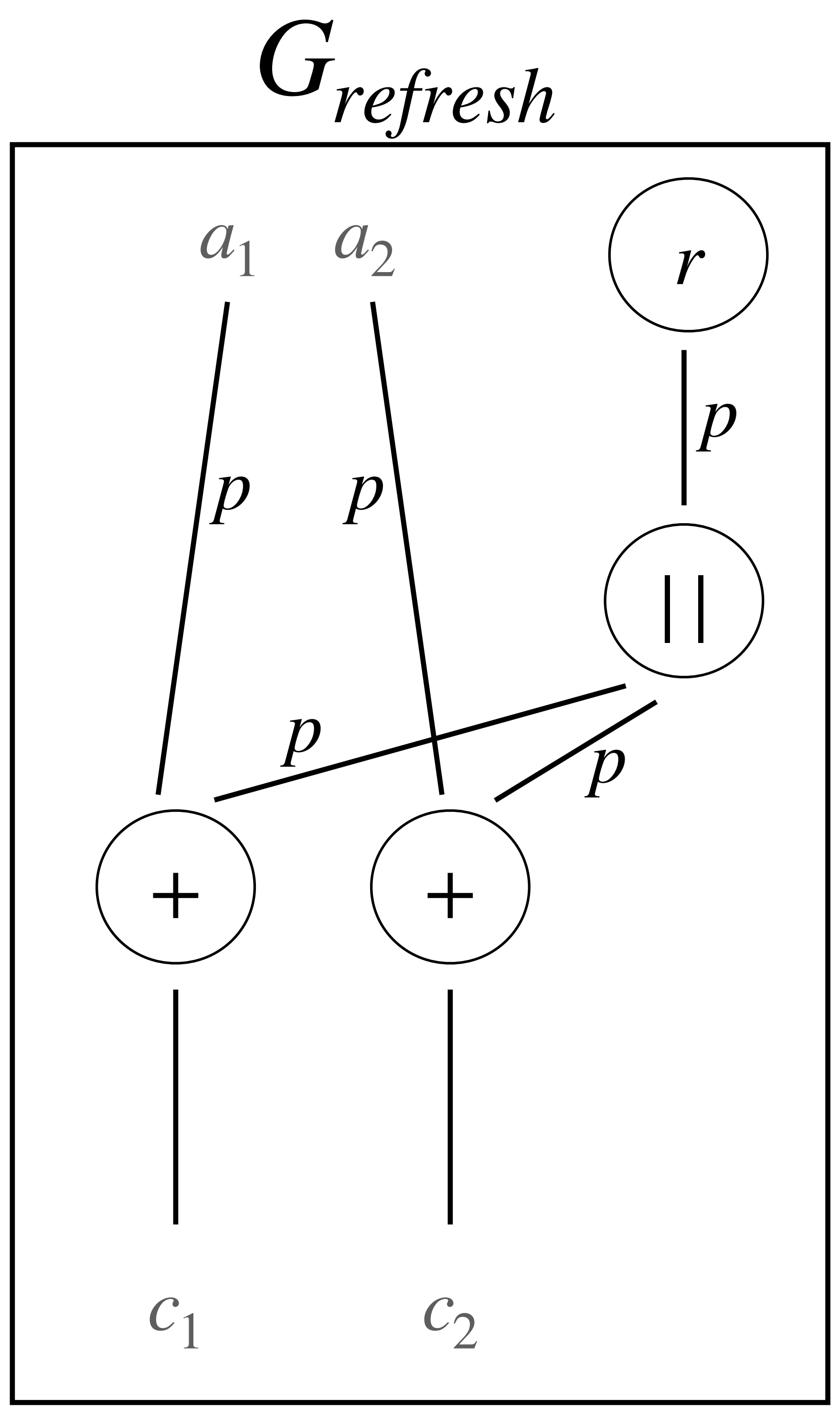
(p, ε) – random probing security

s: number of wires in gadget
$\varepsilon \leftarrow 0$
For $i = 1$ to s

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

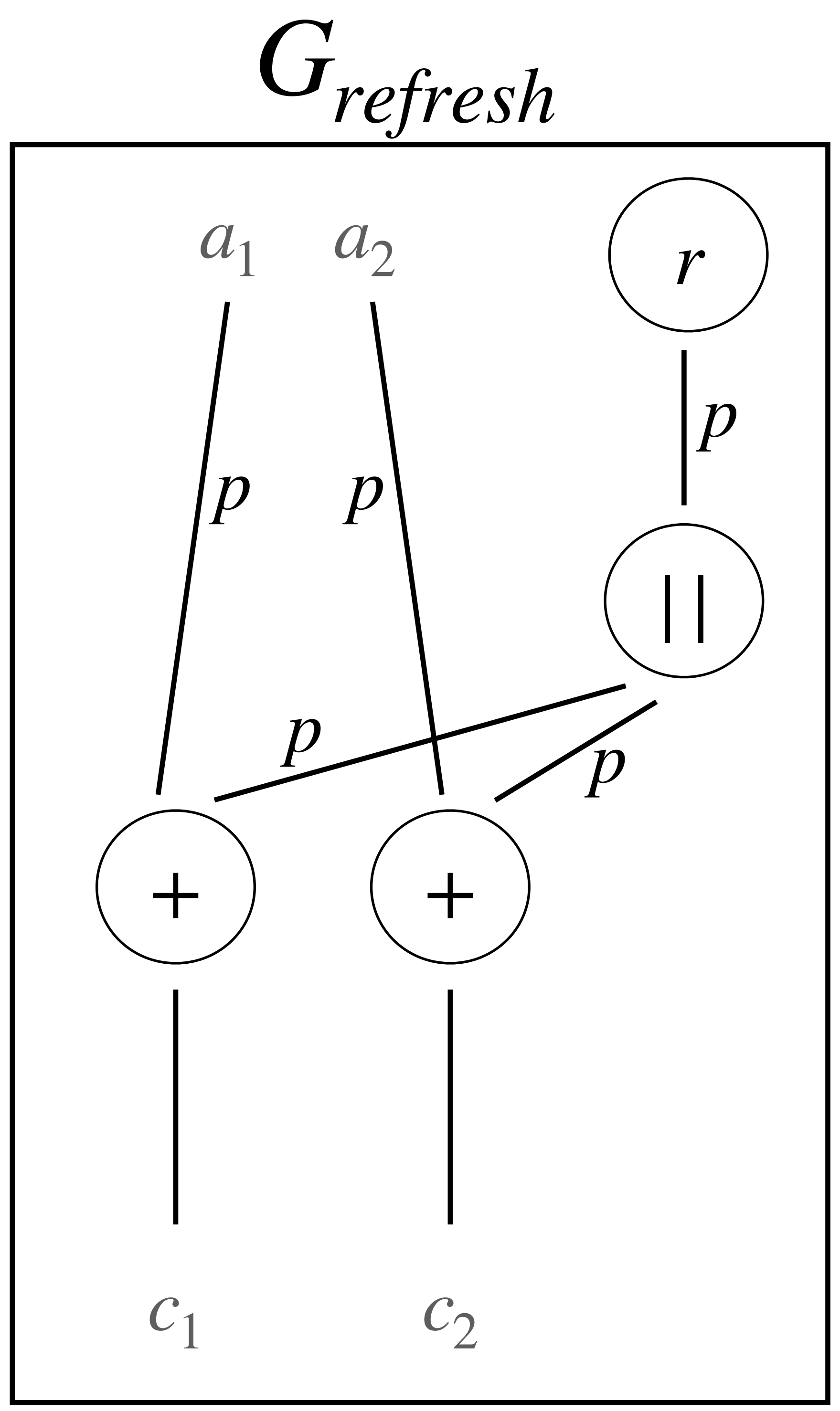
For $i = 1$ to s

- Enumerate all sets of wires of size i

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

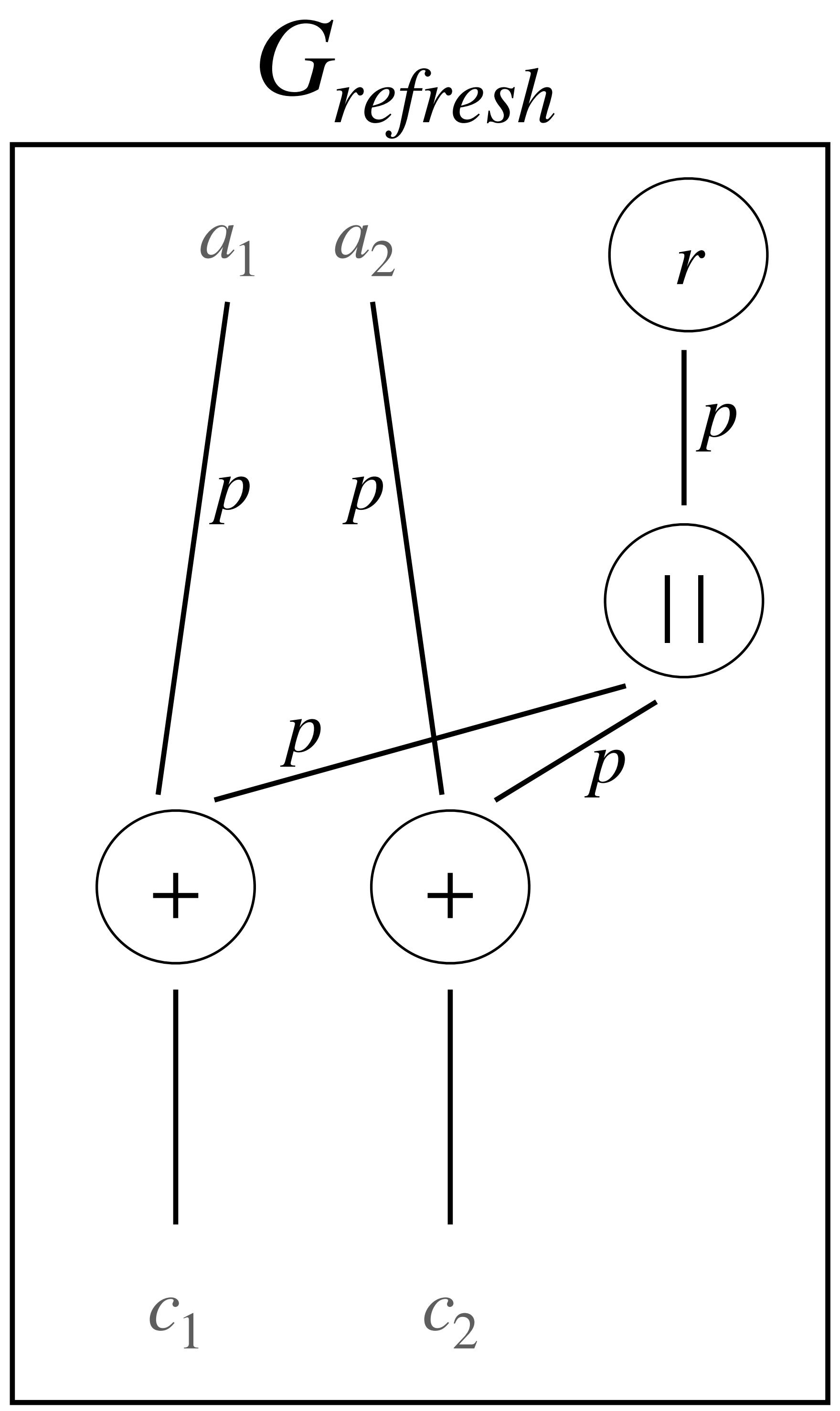
For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

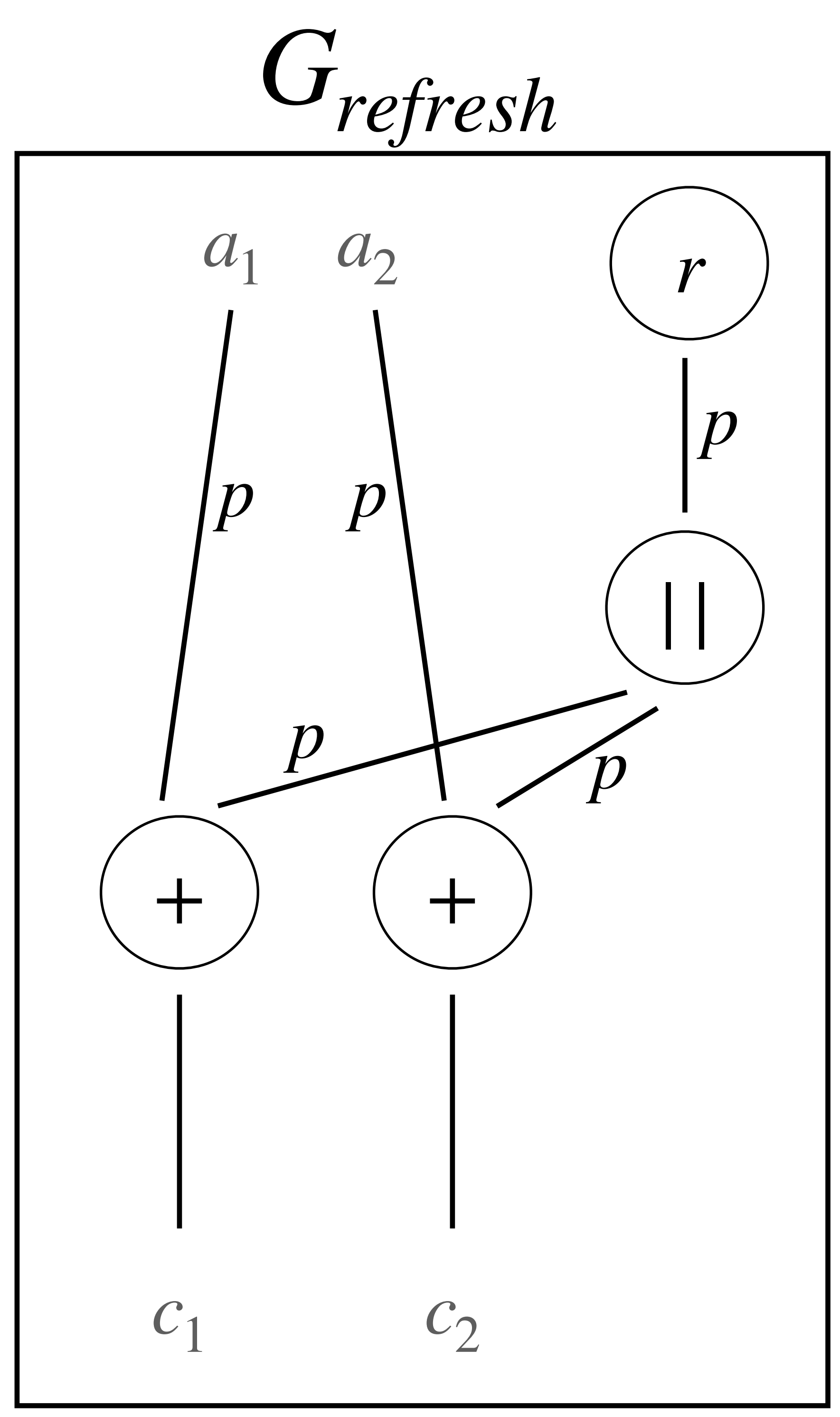
For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

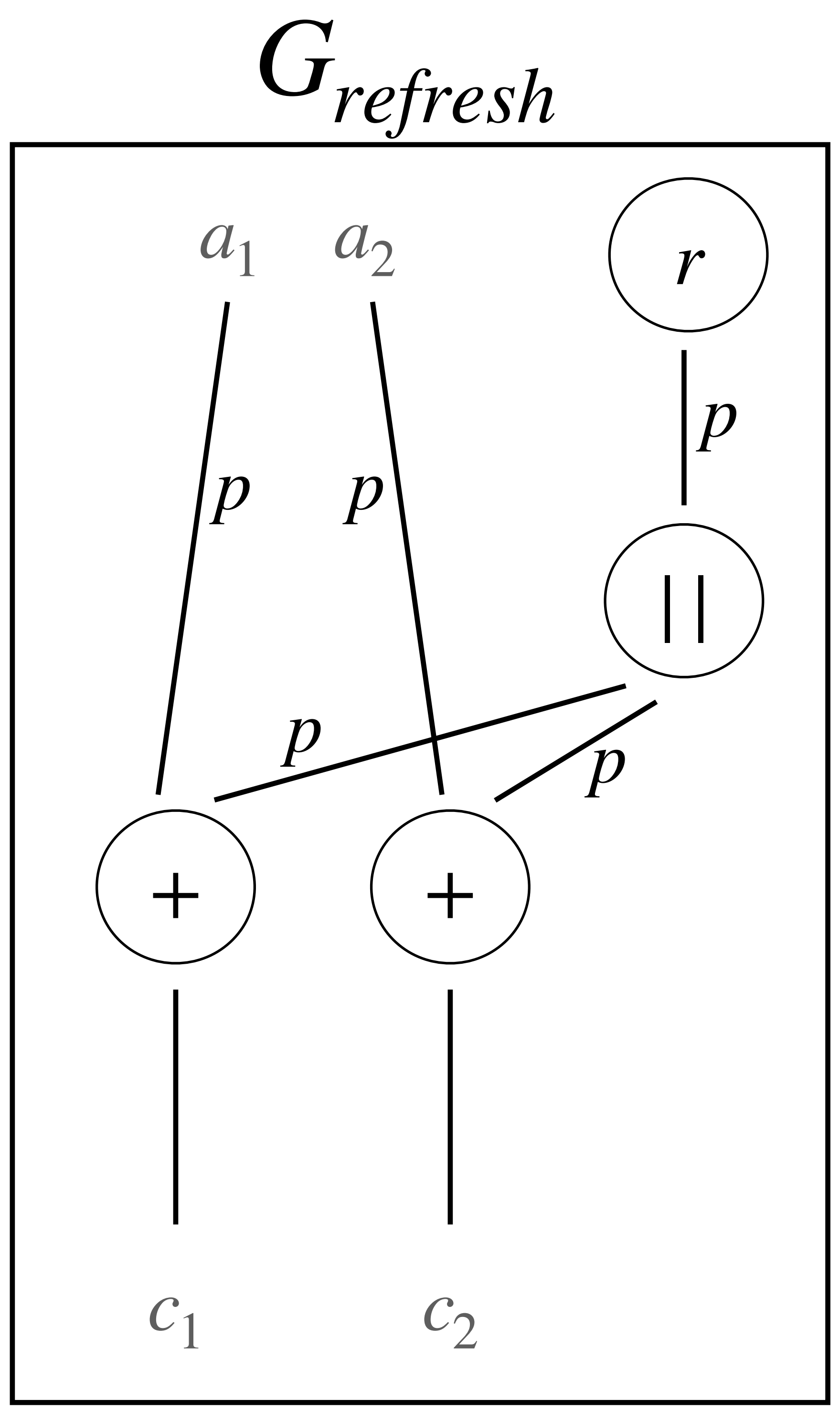
For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
  - $\varepsilon \leftarrow \varepsilon + \Pr(W)$

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

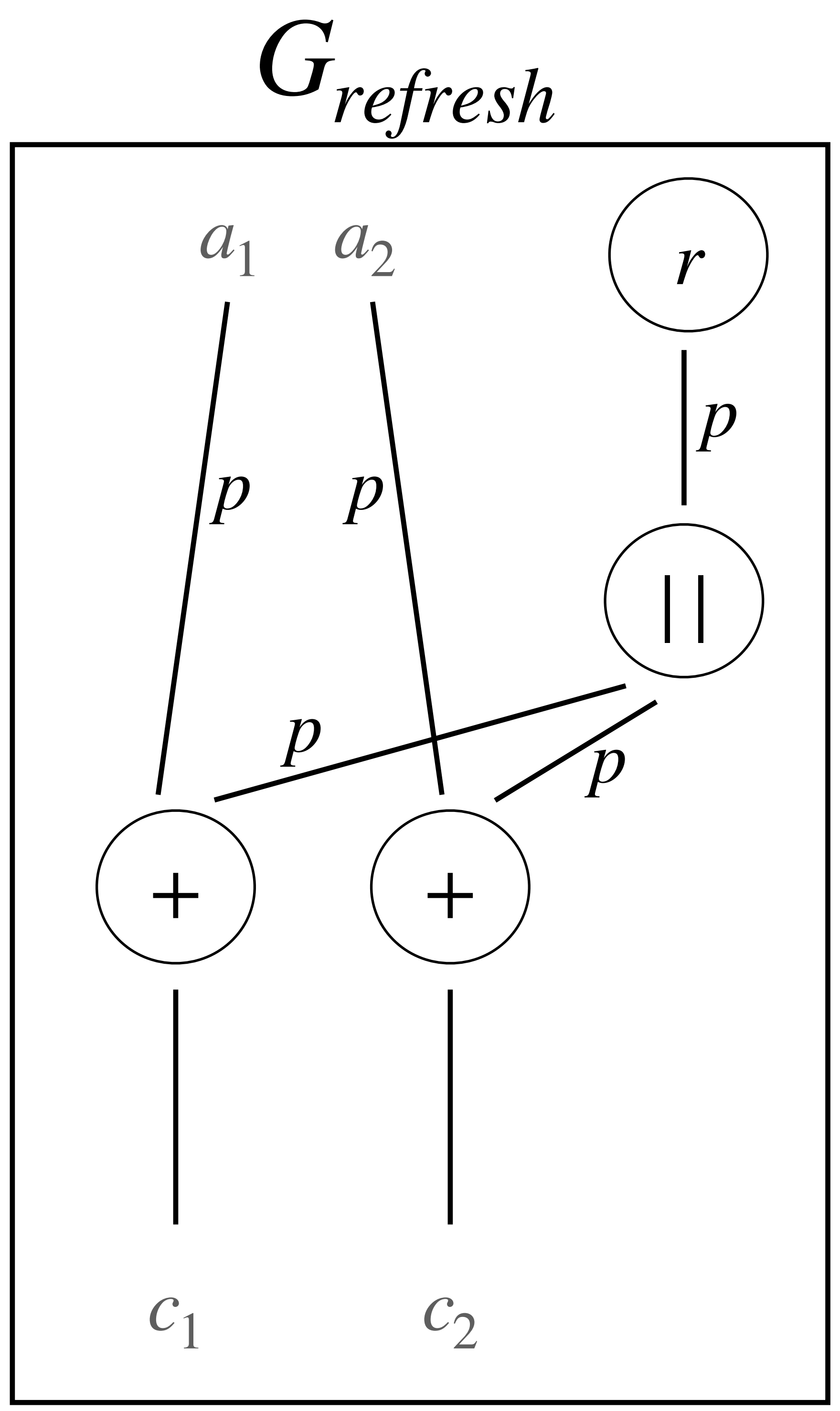
For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
  - $\varepsilon \leftarrow \varepsilon + \text{Pr}(W)$
- End If

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
    - $\varepsilon \leftarrow \varepsilon + \text{Pr}(W)$
- End If

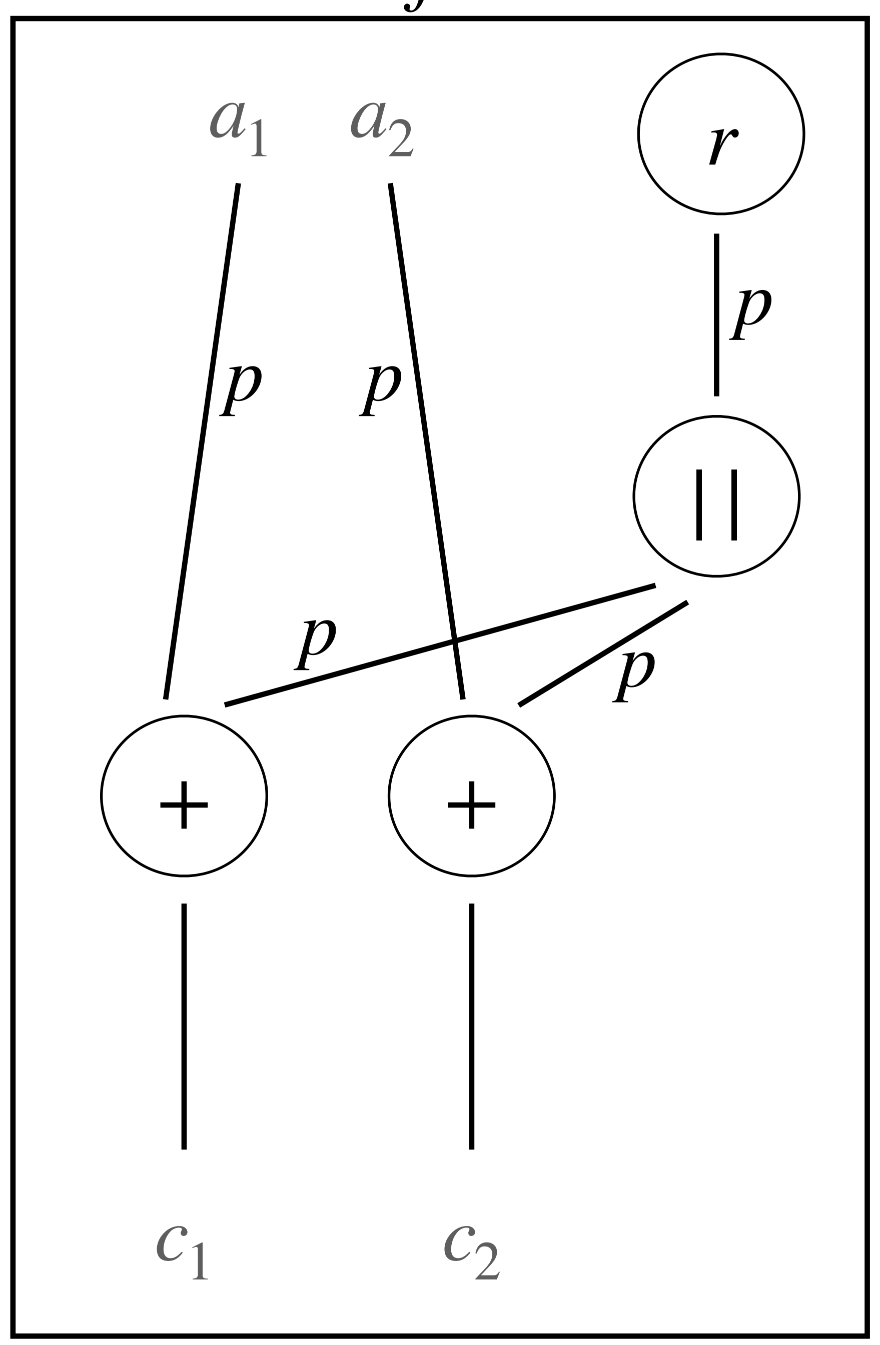
End For

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?

$G_{refresh}$



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
  - $\varepsilon \leftarrow \varepsilon + \mathrm{Pr}(W)$
- End If

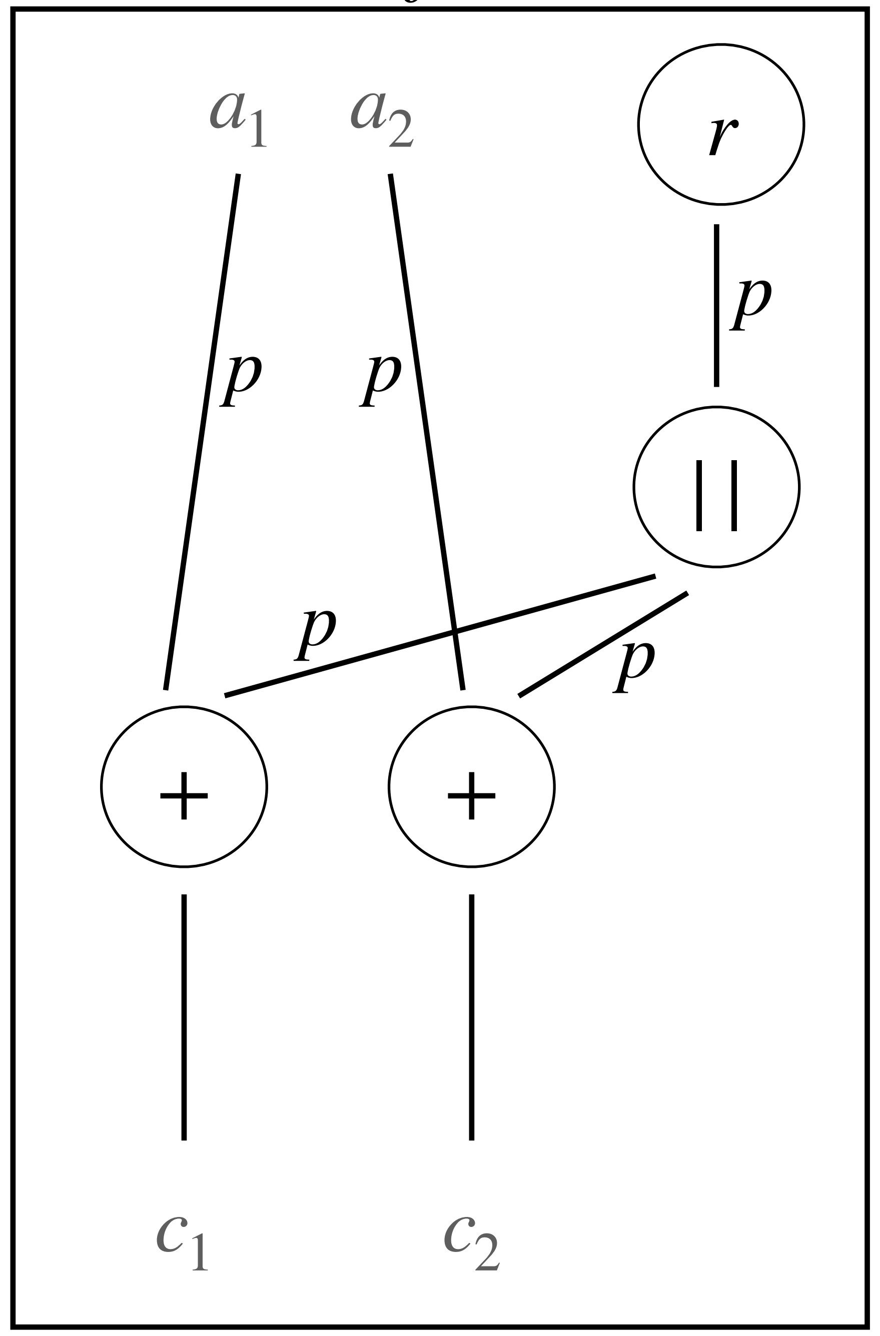
End For

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Random Probing Security

Definition: how to compute ε ?

$G_{refresh}$



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
  - $\varepsilon \leftarrow \varepsilon + \Pr(W)$
- End If

End For

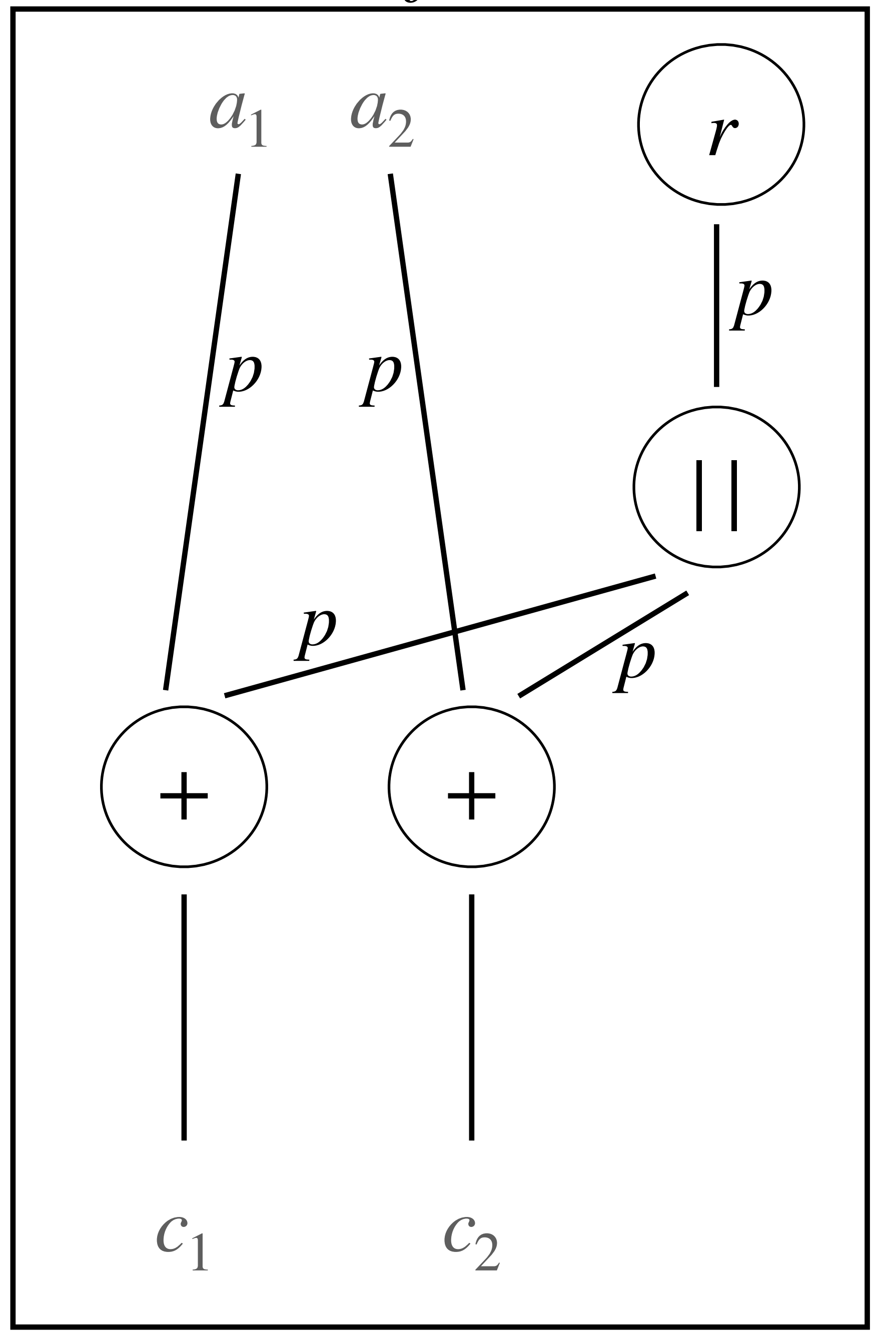
Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

$G_{refresh}$ contains 5 wires

Random Probing Security

Definition: how to compute ε ?

$G_{refresh}$



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
    - $\varepsilon \leftarrow \varepsilon + \Pr(W)$
- End If

End For

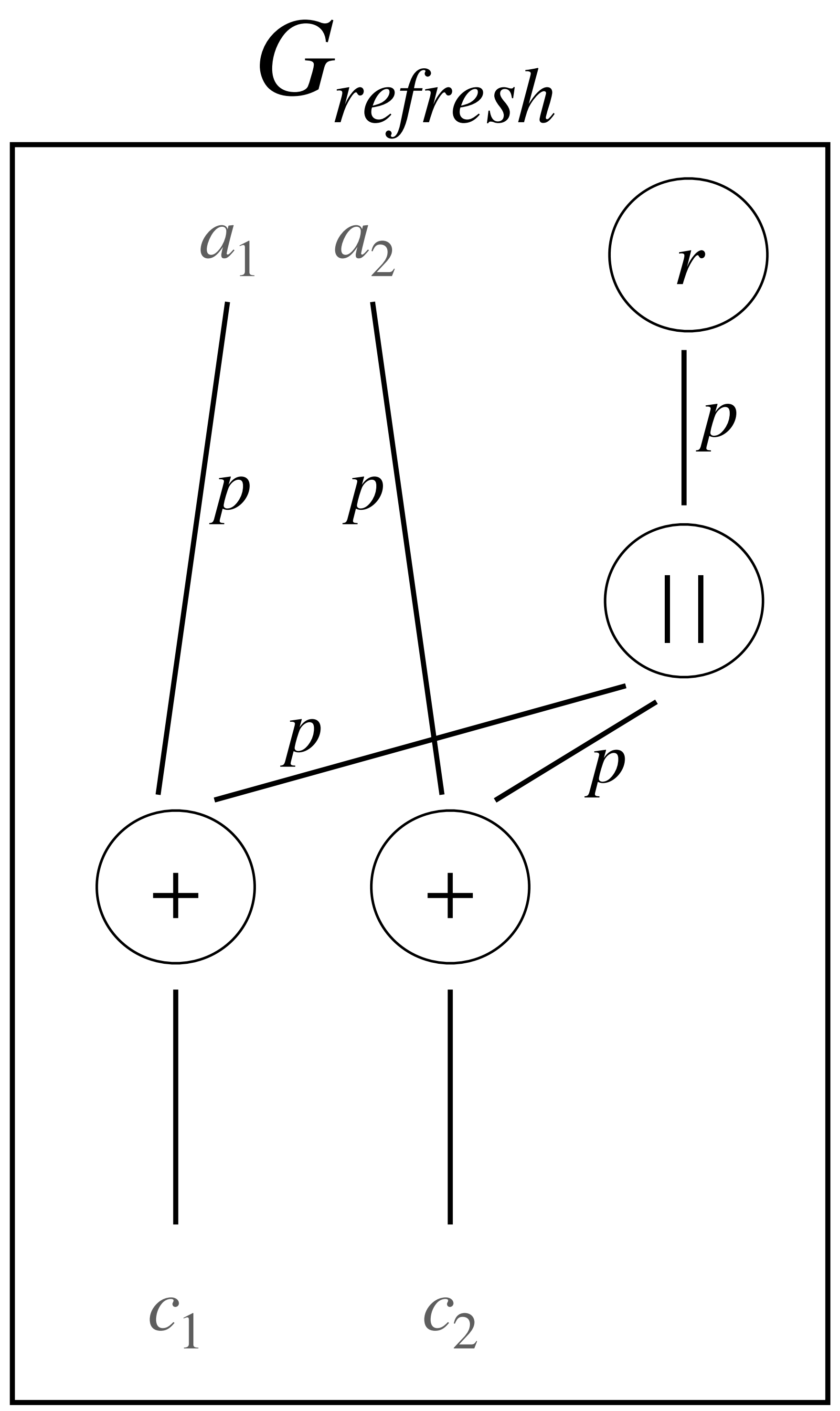
Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

$G_{refresh}$ contains 5 wires

31 sets to check

Random Probing Security

Definition: how to compute ε ?



(p, ε) – random probing security

s: number of wires in gadget

$\varepsilon \leftarrow 0$

For $i = 1$ to s

- Enumerate all sets of wires of size i
- Check if each set is independent from the secrets
- If Failure on set W, then
  - $\varepsilon \leftarrow \varepsilon + \Pr(W)$
- End If

End For

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

$G_{refresh}$ contains 5 wires

31 sets to check

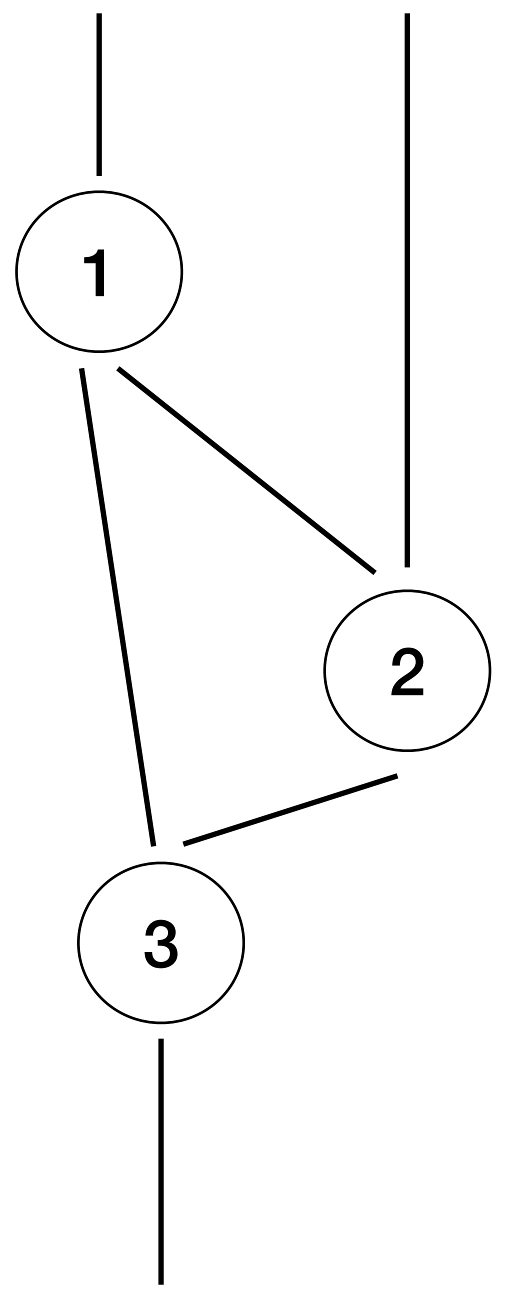
9 of them are Failures

Random Probing Security

Expansion: how to amplify the security ε ? Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*

Random Probing Security

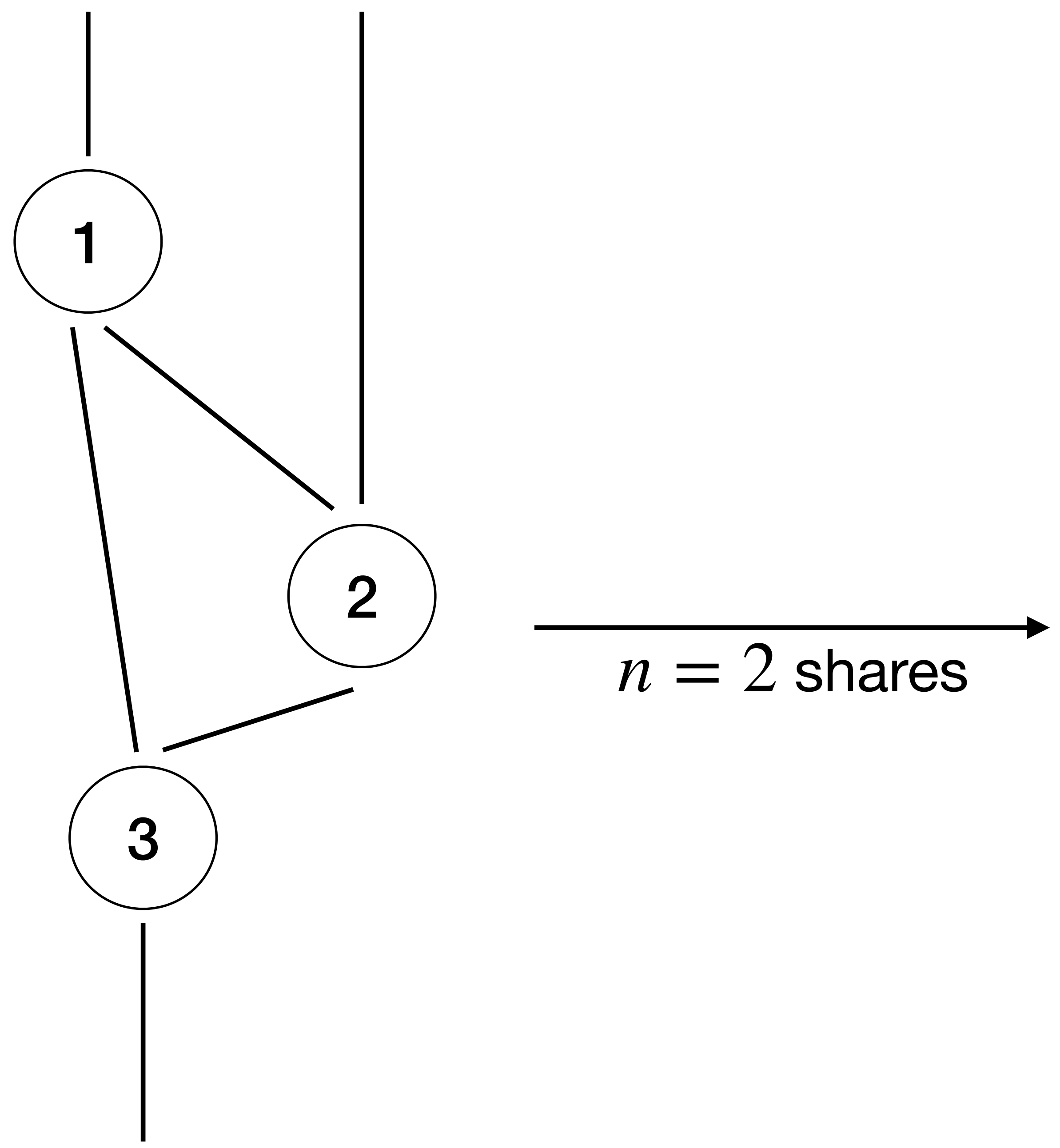
Expansion: how to amplify the security ε ? Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Leakage probability p

Random Probing Security

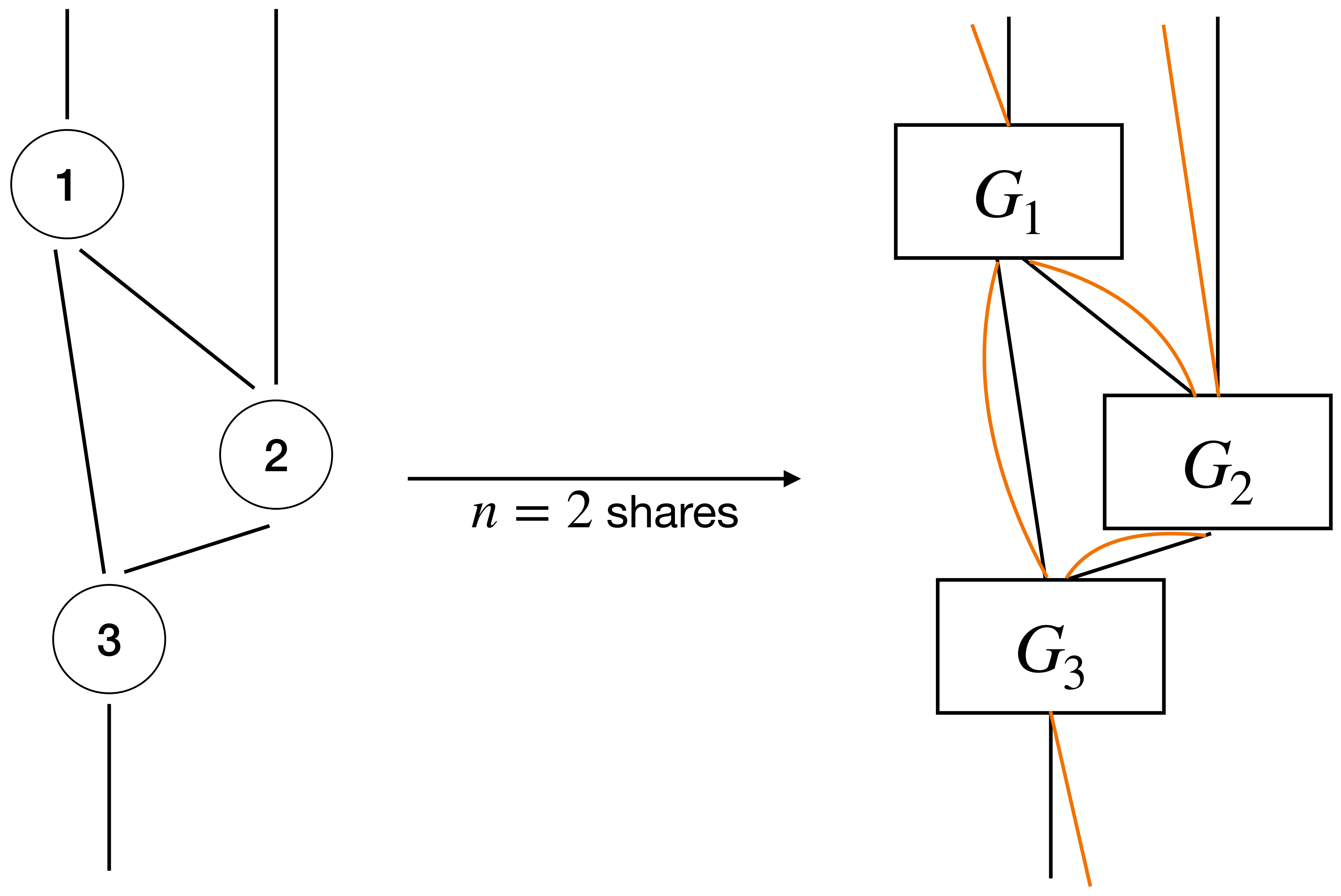
Expansion: how to amplify the security ε ? Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Leakage probability p

Random Probing Security

Expansion: how to amplify the security ε ? Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*

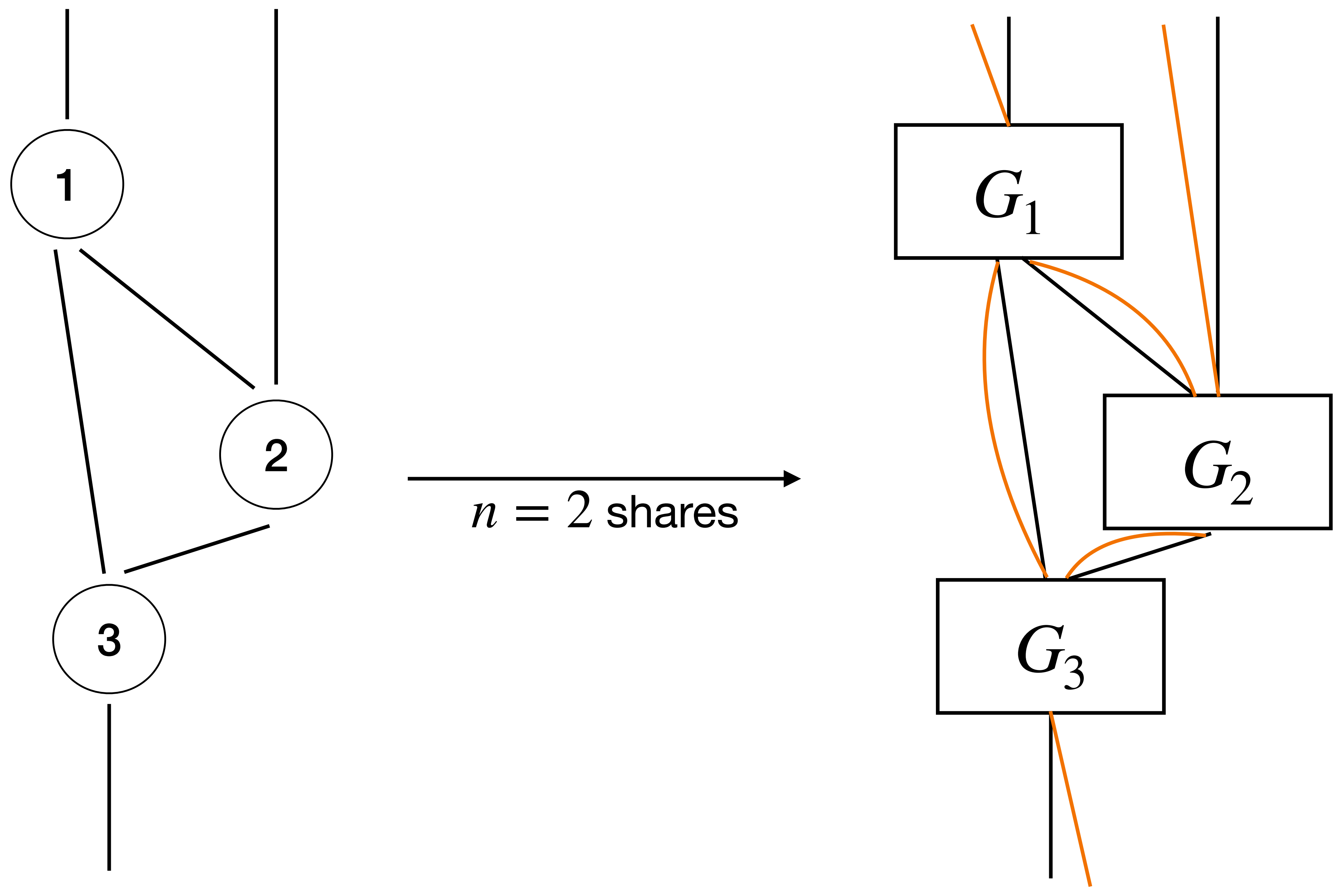


Leakage probability p

Random Probing Security

Expansion: how to amplify the security ε ?

Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



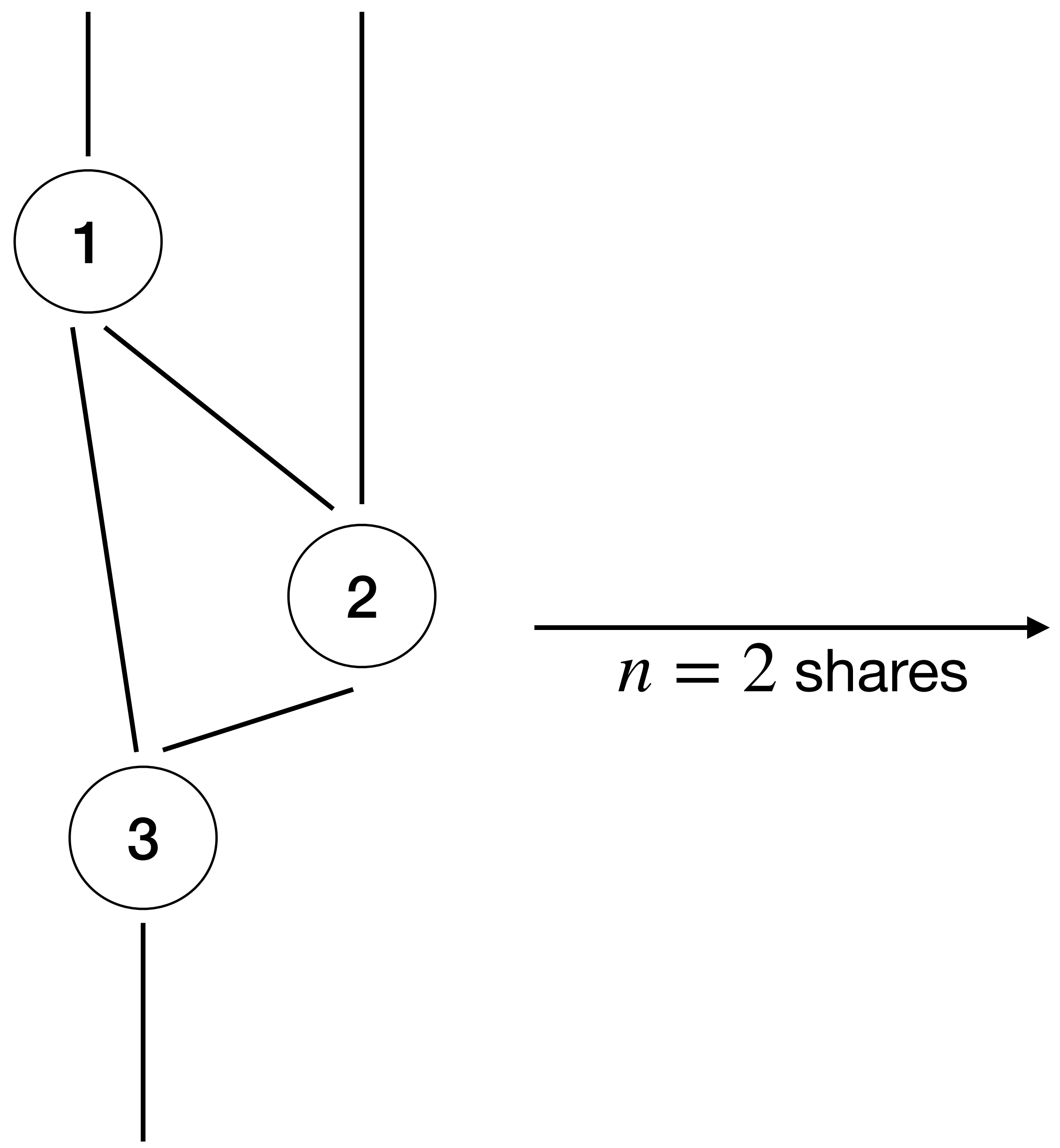
Leakage probability p

Failure probability ε

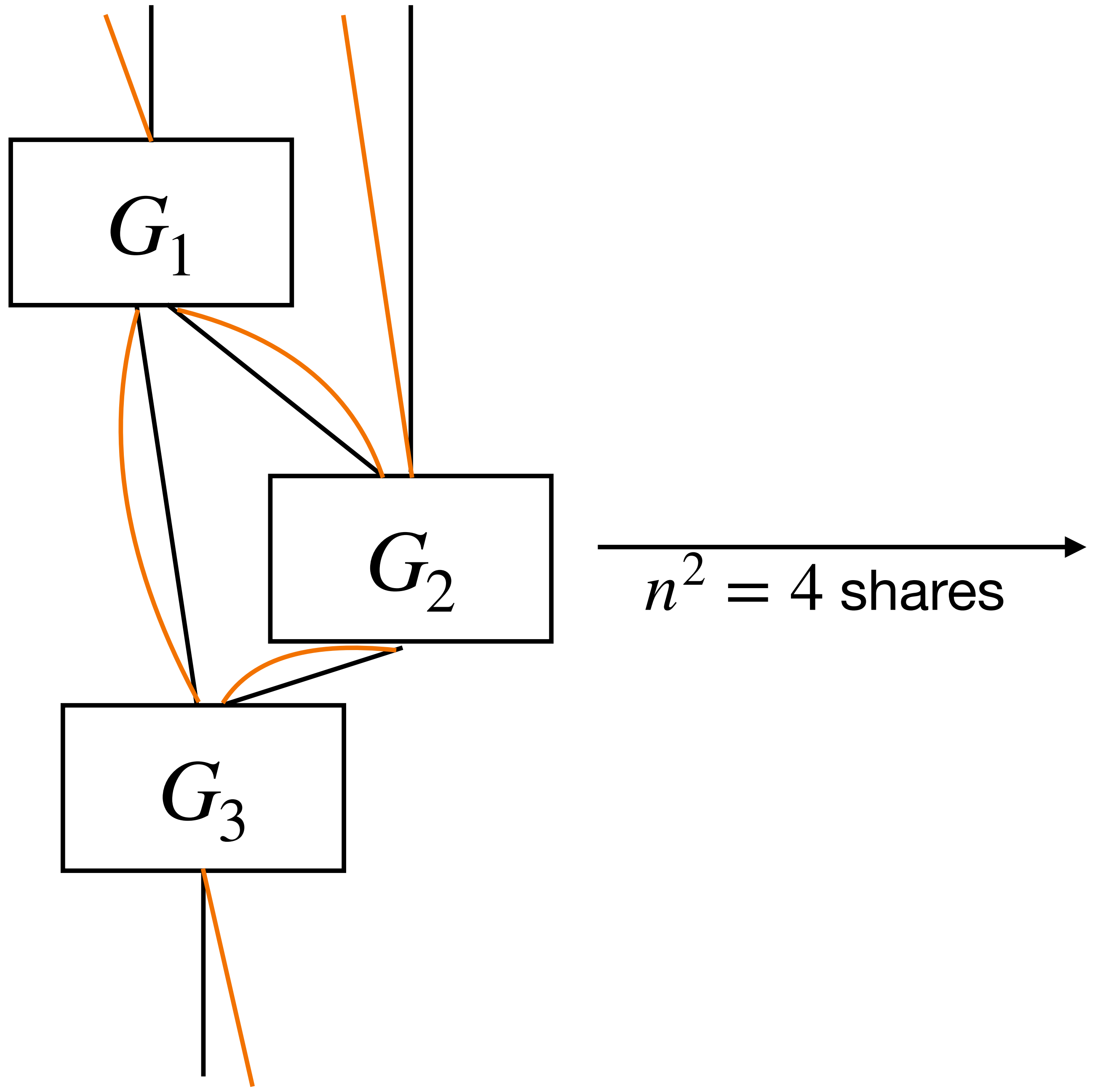
Random Probing Security

Expansion: how to amplify the security ε ?

Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Leakage probability p

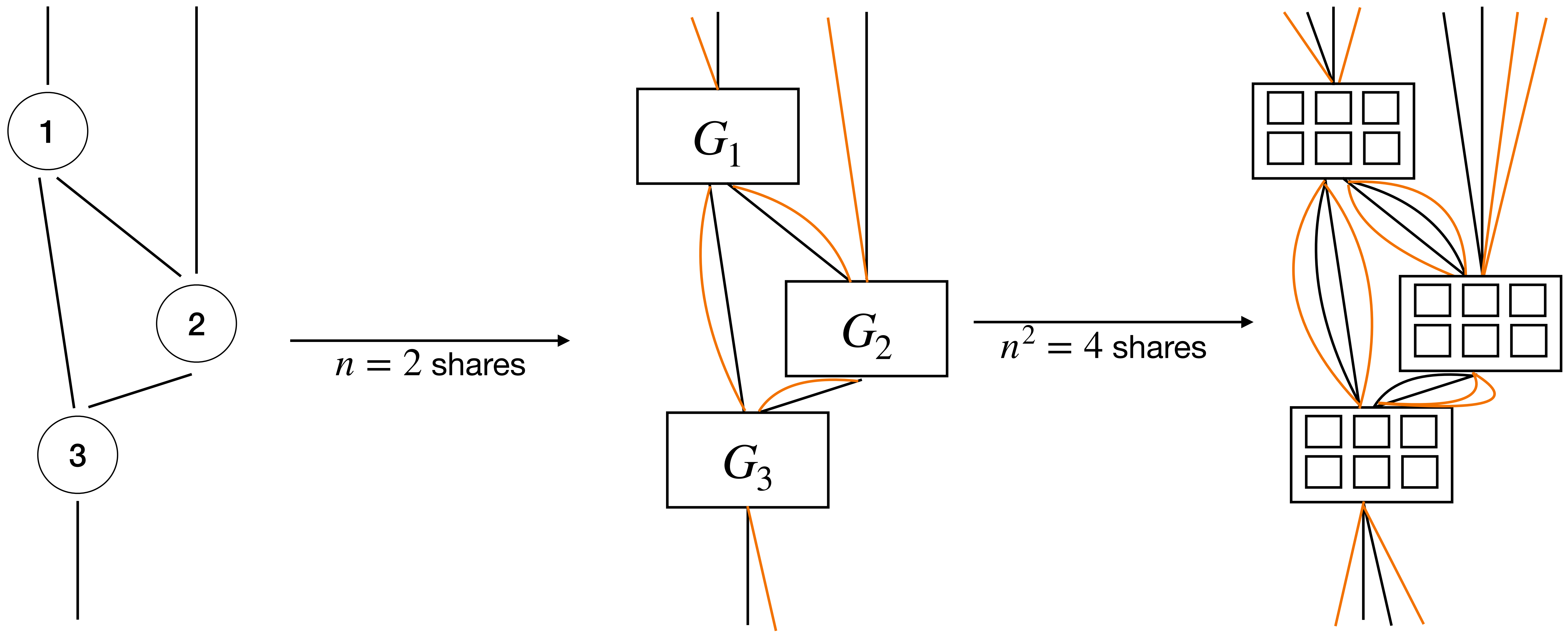


Failure probability ε

Random Probing Security

Expansion: how to amplify the security ε ?

Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



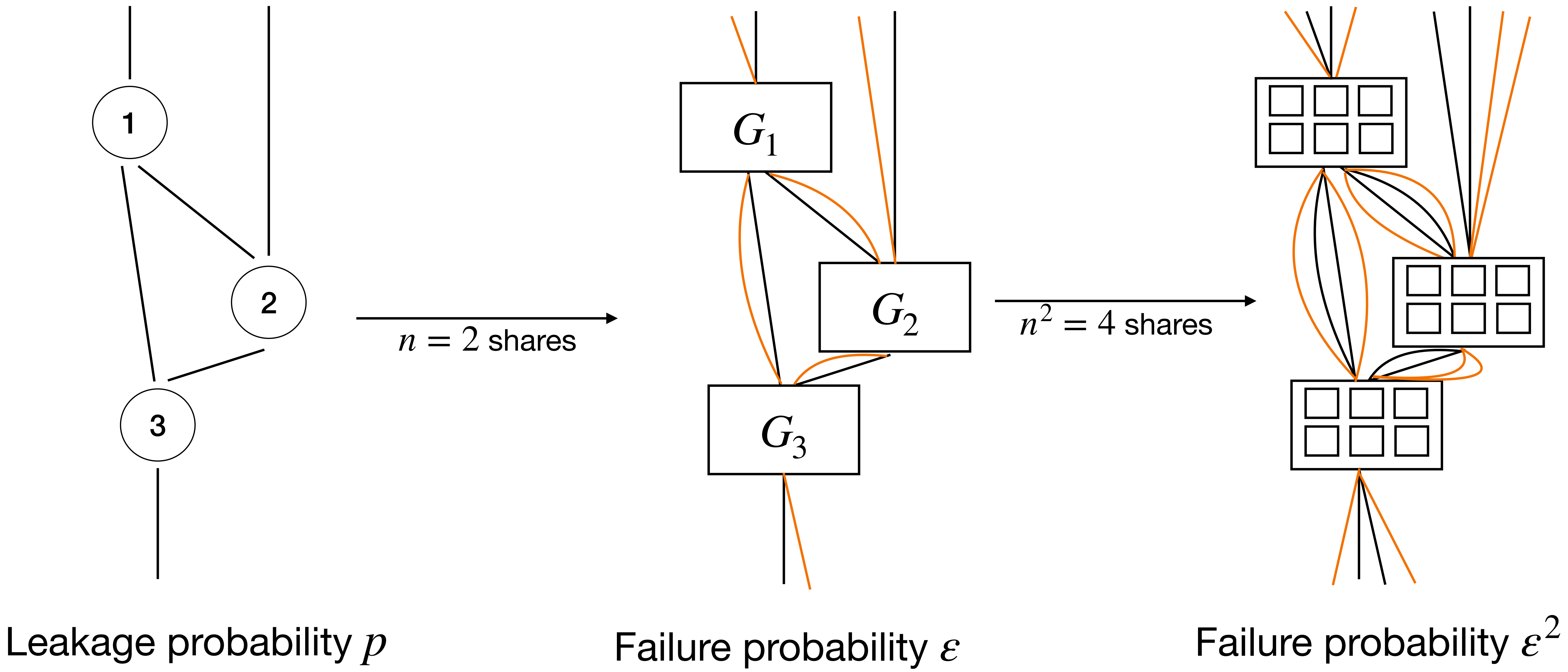
Leakage probability p

Failure probability ε

Random Probing Security

Expansion: how to amplify the security ε ?

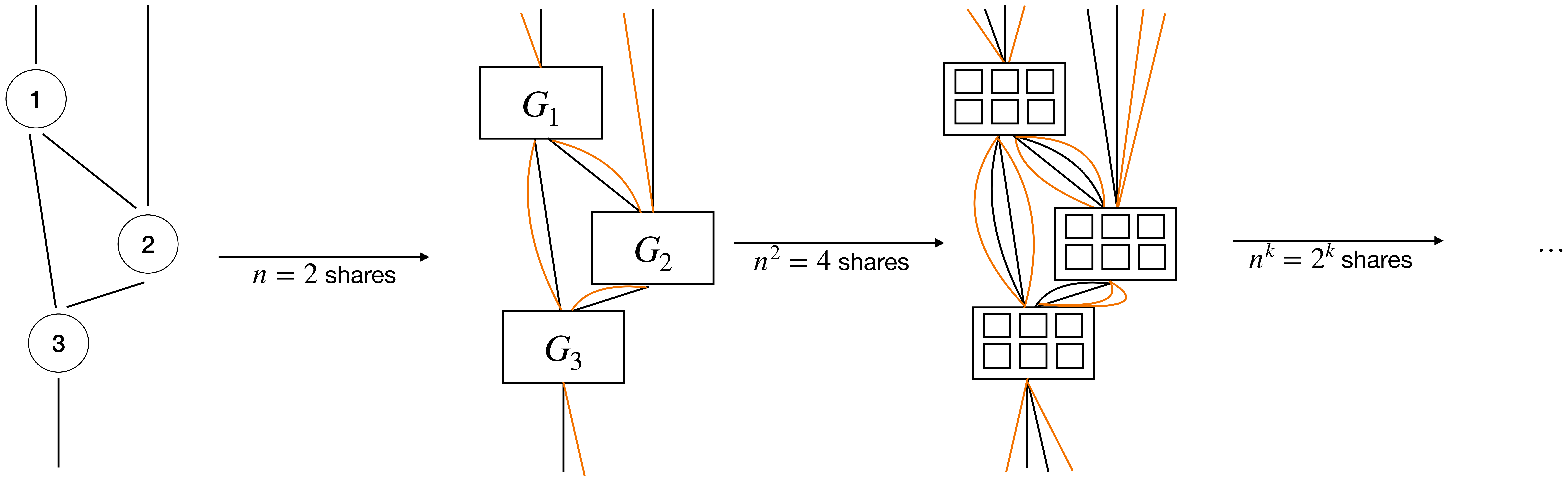
Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Random Probing Security

Expansion: how to amplify the security ε ?

Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Leakage probability p

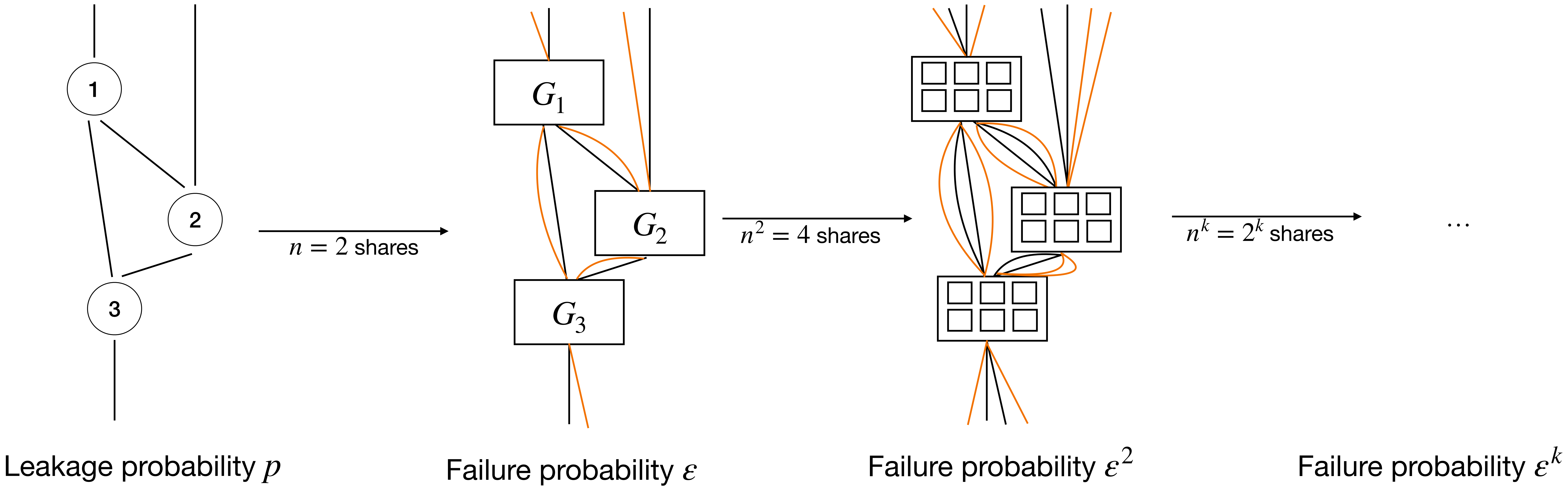
Failure probability ε

Failure probability ε^2

Random Probing Security

Expansion: how to amplify the security ε ?

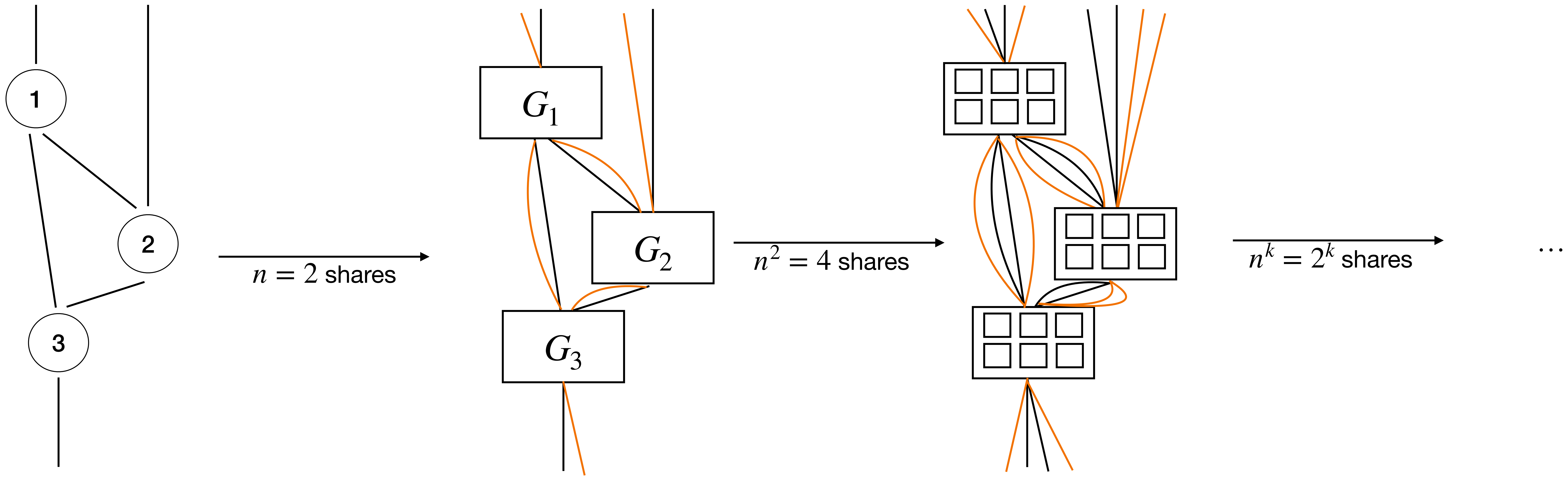
Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Random Probing Security

Expansion: how to amplify the security ε ?

Revisited approach from *Ananth, Ishai and Sahai [CRYPTO'18]*



Leakage probability p | Failure probability ε | Failure probability ε^2 | Failure probability ε^k

Condition: $\varepsilon < p$

Random Probing Expansion

Example

Random Probing Expansion

Example

$$c = a \times b$$

Random Probing Expansion

Example

$$c = a \times b$$

$$n = 2$$

Random Probing Expansion

Example

$$c = a \times b$$

$$n = 2$$

$$c_1 = a_1 \times b_1 + r_{12}$$

Random Probing Expansion

Example

$$c = a \times b$$

$$n = 2$$

$$c_1 = a_1 \times b_1 + r_{12}$$
$$c_2 = a_2 \times b_2 + ((a_1 \times b_2 + r_{12}) + a_2 \times b_1)$$

Random Probing Expansion

Example

$$c = a \times b$$

$n = 2$

2-share ISW multplication gadget

Ishai, Sahai, and Wagner [CRYPTO'03]

$$c_1 = a_1 \times b_1 + r_{12}$$
$$c_2 = a_2 \times b_2 + ((a_1 \times b_2 + r_{12}) + a_2 \times b_1)$$

Random Probing Expansion

Example

$$c = a \times b$$

$n = 2$

2-share ISW
multplication gadget

Ishai, Sahai, and Wagner
[CRYPTO'03]

$$c_1 = a_1 \times b_1 + r_{12}$$
$$c_2 = a_2 \times b_2 + ((a_1 \times b_2 + r_{12}) + a_2 \times b_1)$$

$n^2 = 4$

Random Probing Expansion

Example

$$c = a \times b$$

$n = 2$

2-share ISW multplication gadget

Ishai, Sahai, and Wagner [CRYPTO'03]

$$c_1 = a_1 \times b_1 + r_{12}$$
$$c_2 = a_2 \times b_2 + ((a_1 \times b_2 + r_{12}) + a_2 \times b_1)$$

$n^2 = 4$

$$(c_{1,1}, c_{1,2}) = G_+\Big(\; G_\times((a_{1,1}, a_{1,2}), (b_{1,1}, b_{1,2})) \;\; , \;\; (r_{12,1}, r_{12,2}) \; \Big)$$

Random Probing Expansion

Example

$$c = a \times b$$

$n = 2$

2-share ISW multplication gadget

Ishai, Sahai, and Wagner [CRYPTO'03]

$$c_1 = a_1 \times b_1 + r_{12}$$
$$c_2 = a_2 \times b_2 + ((a_1 \times b_2 + r_{12}) + a_2 \times b_1)$$

$n^2 = 4$

$$(c_{1,1}, c_{1,2}) = G_+\Big(\; G_\times((a_{1,1}, a_{1,2}), (b_{1,1}, b_{1,2})) \; , \; (r_{12,1}, r_{12,2}) \; \Big)$$
$$(c_{2,1}, c_{2,2}) = \ldots$$

Random Probing Expansion

Definition

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)

Random Probing Expansion
Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

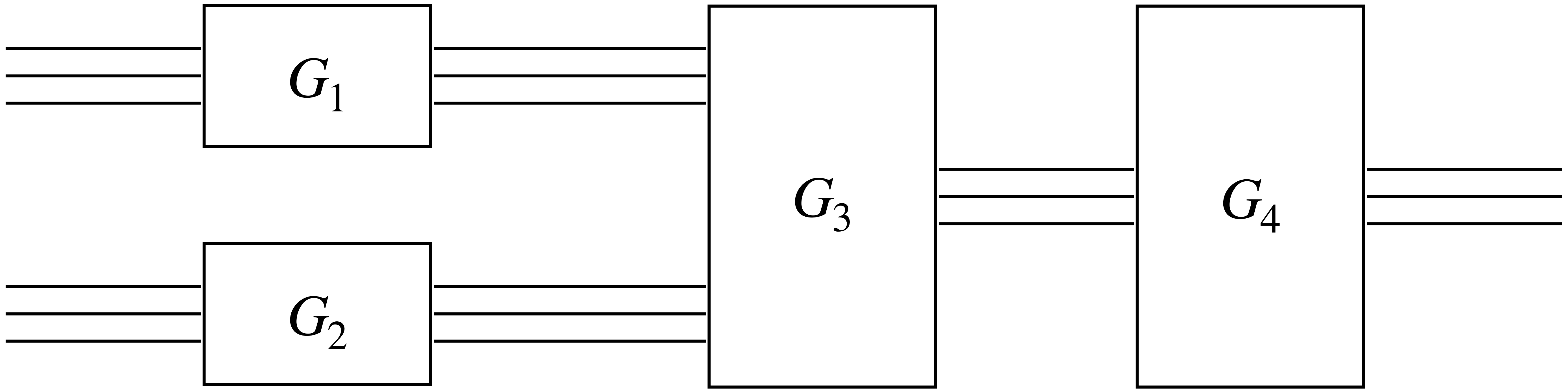
- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

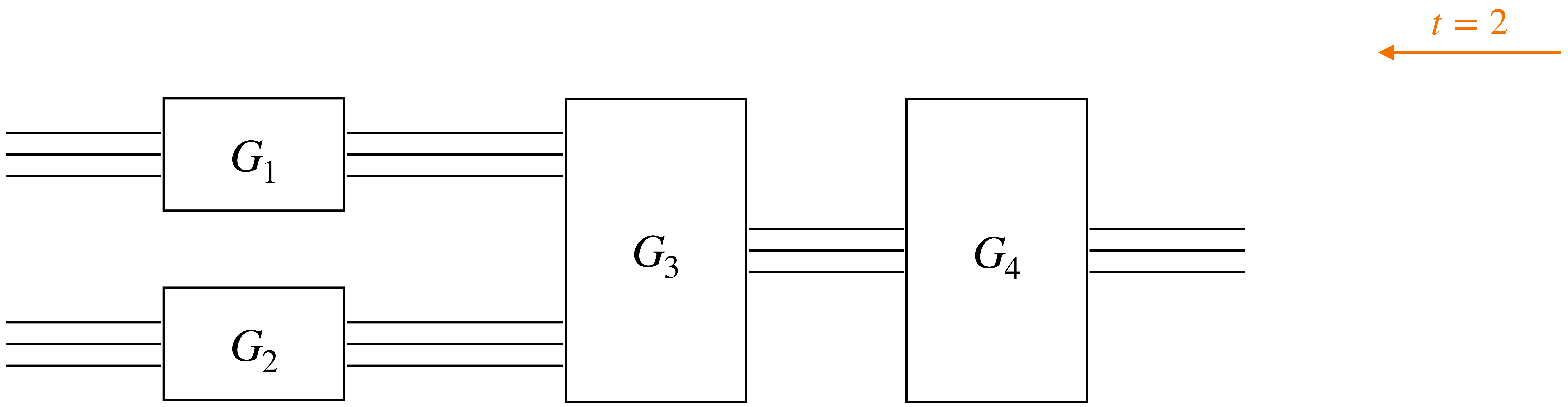


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

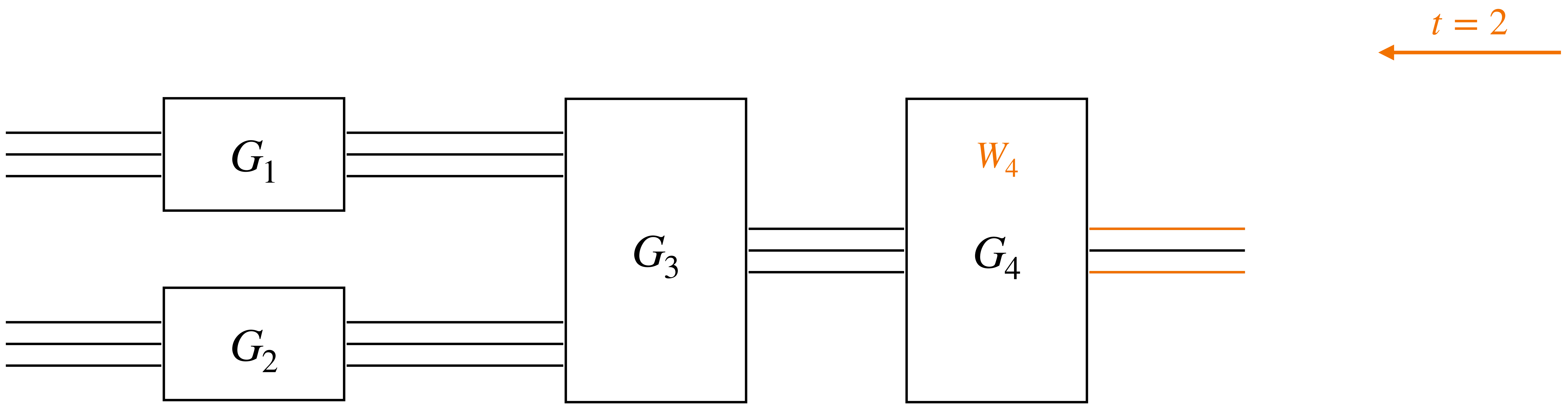


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

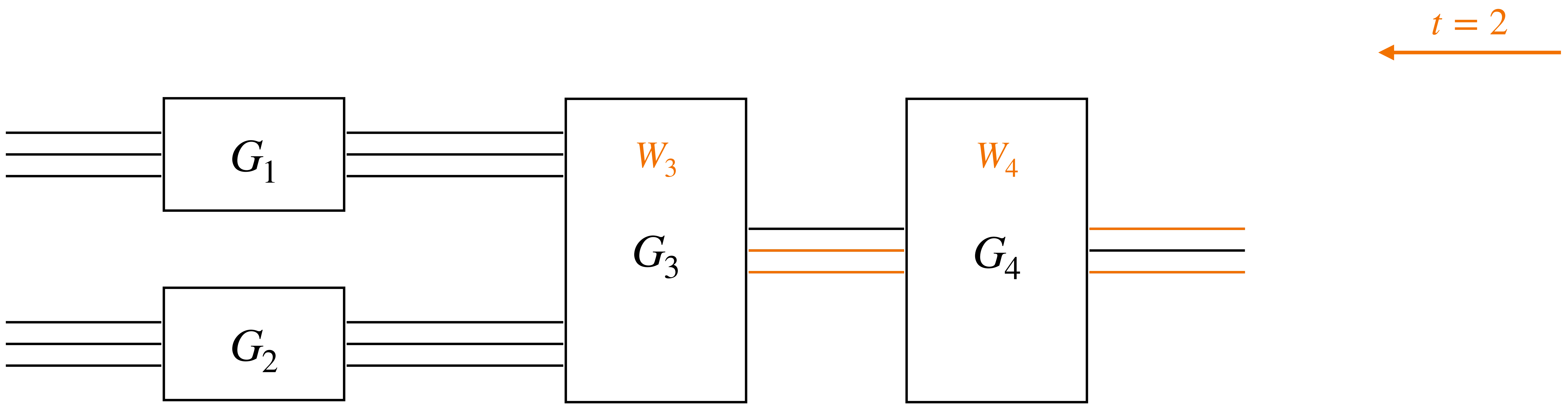


Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

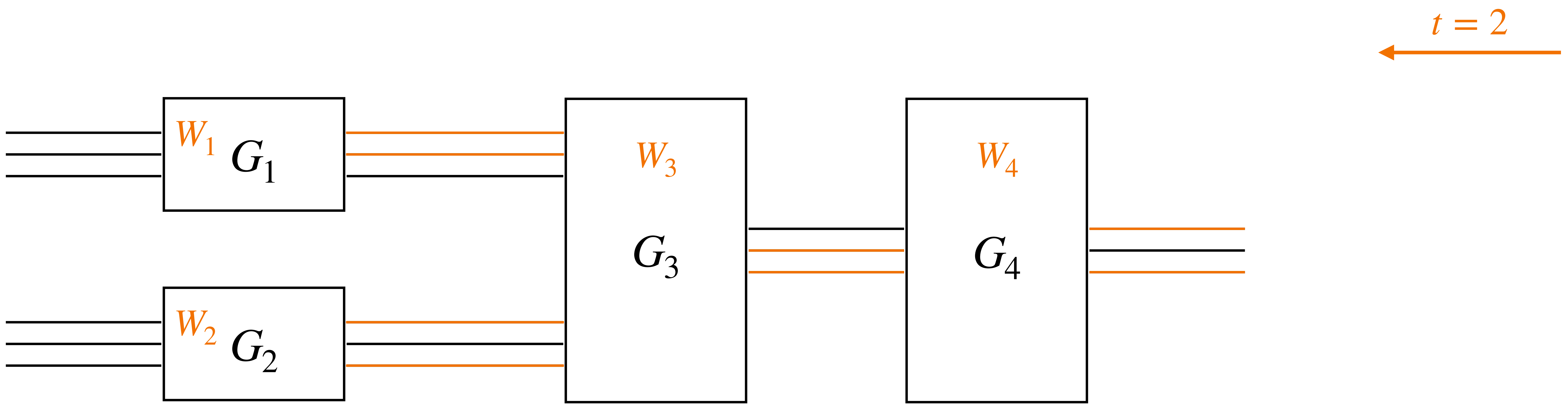


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

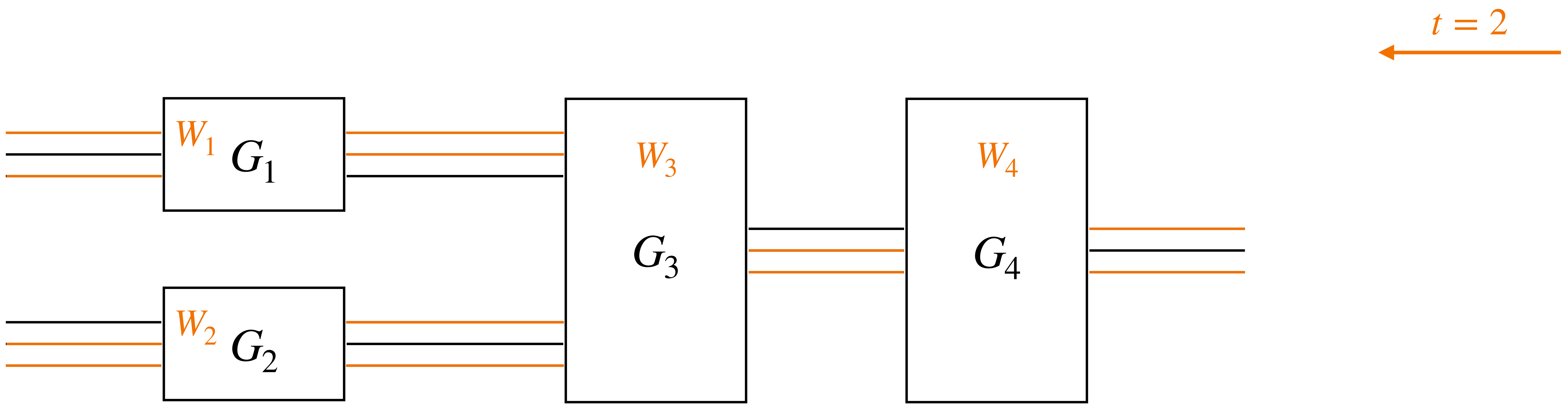


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

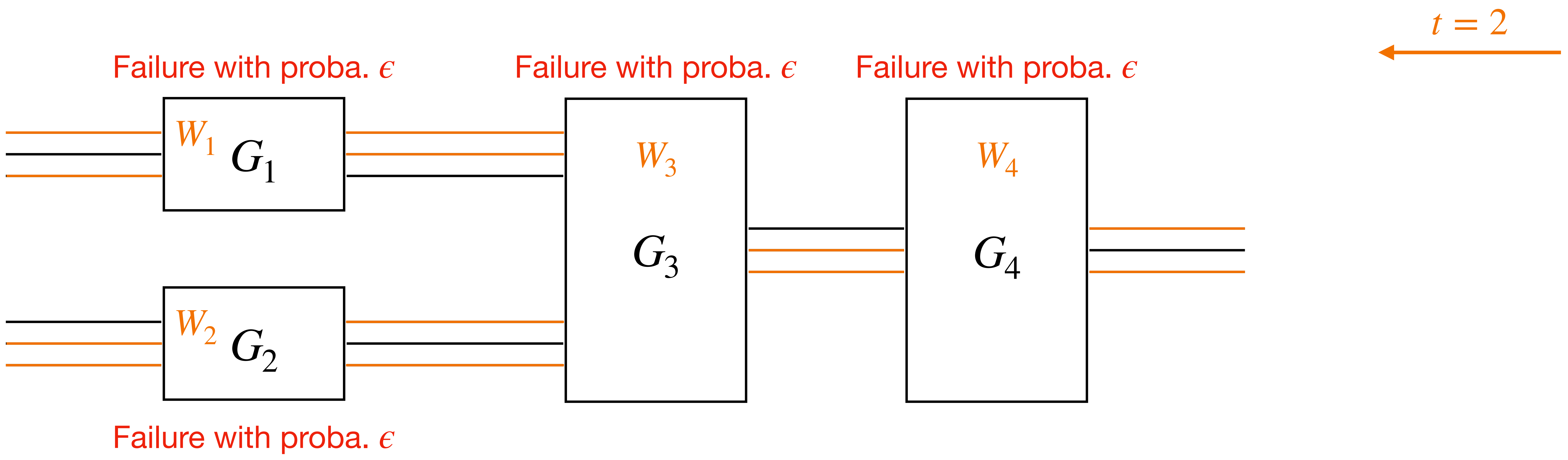


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

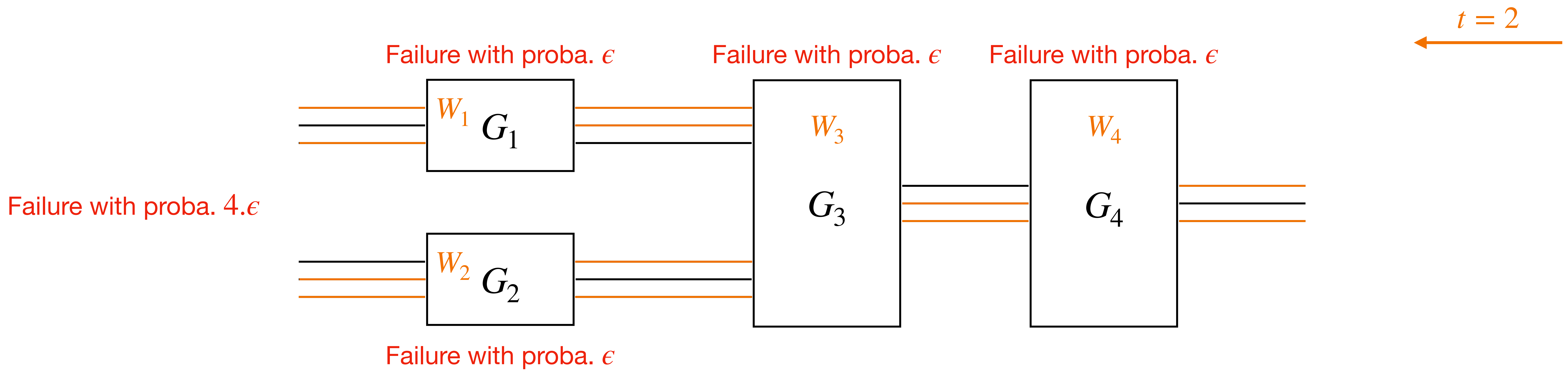


Random Probing Expansion

Definition

(t, p, ε) - **Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



Random Probing Expansion
Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

All gadgets are (t, p, ε) - random probing expandable

Random Probing Expansion

Definition

(t, p, ε) **- Random Probing Expandability** of a gadget G garantees:

- G is (p, ε) - RP secure (RPE >> RP)
- Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

All gadgets are (t, p, ε) - random probing expandable → A circuit C compiled from scratch is $(p,\ 2.|C|.\varepsilon^k)$ - random probing secure

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C| . \kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Example $t = 1, n = 2$:

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C| \,.\, \kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Example $t = 1, n = 2$:

G_+^2

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

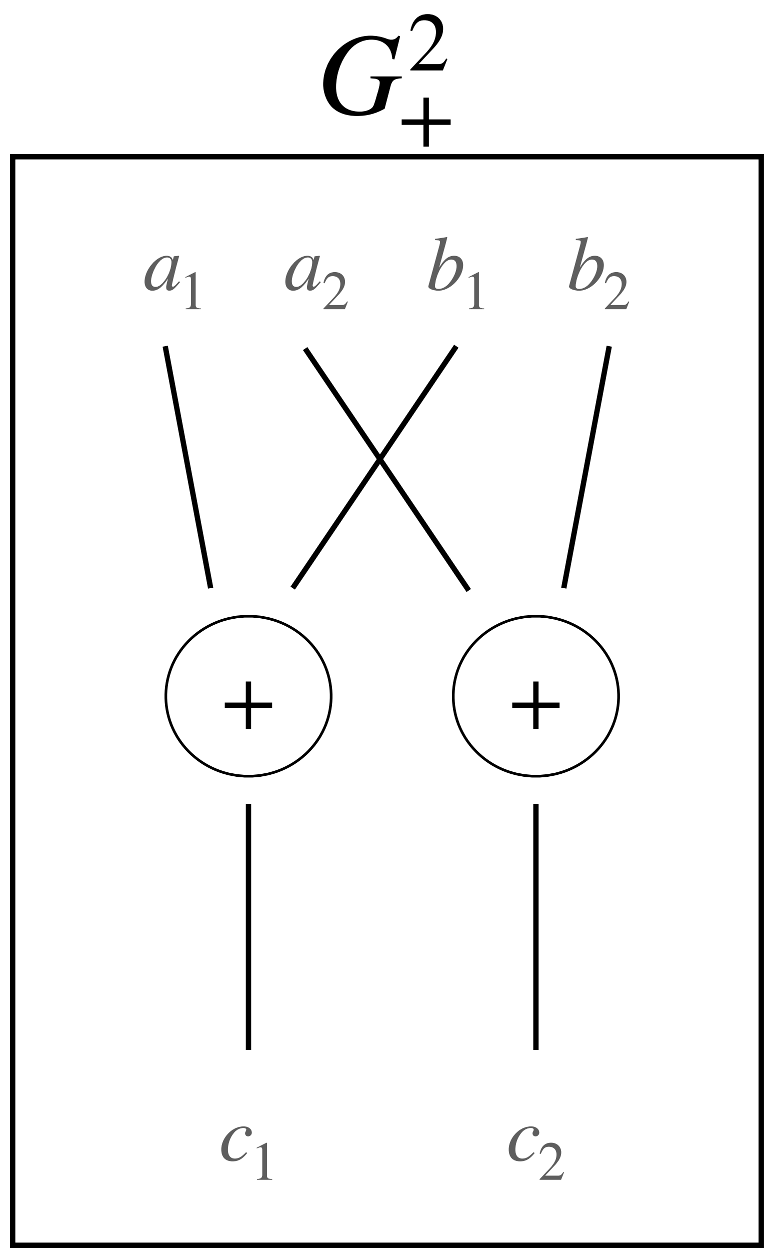
$$\mathcal{O}(|C| . \kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Example $t = 1, n = 2$:



Ability to simulate any set of internal wires **and t output shares** using at most t input shares of each input

Random Probing Expansion

Complexity

Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \qquad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

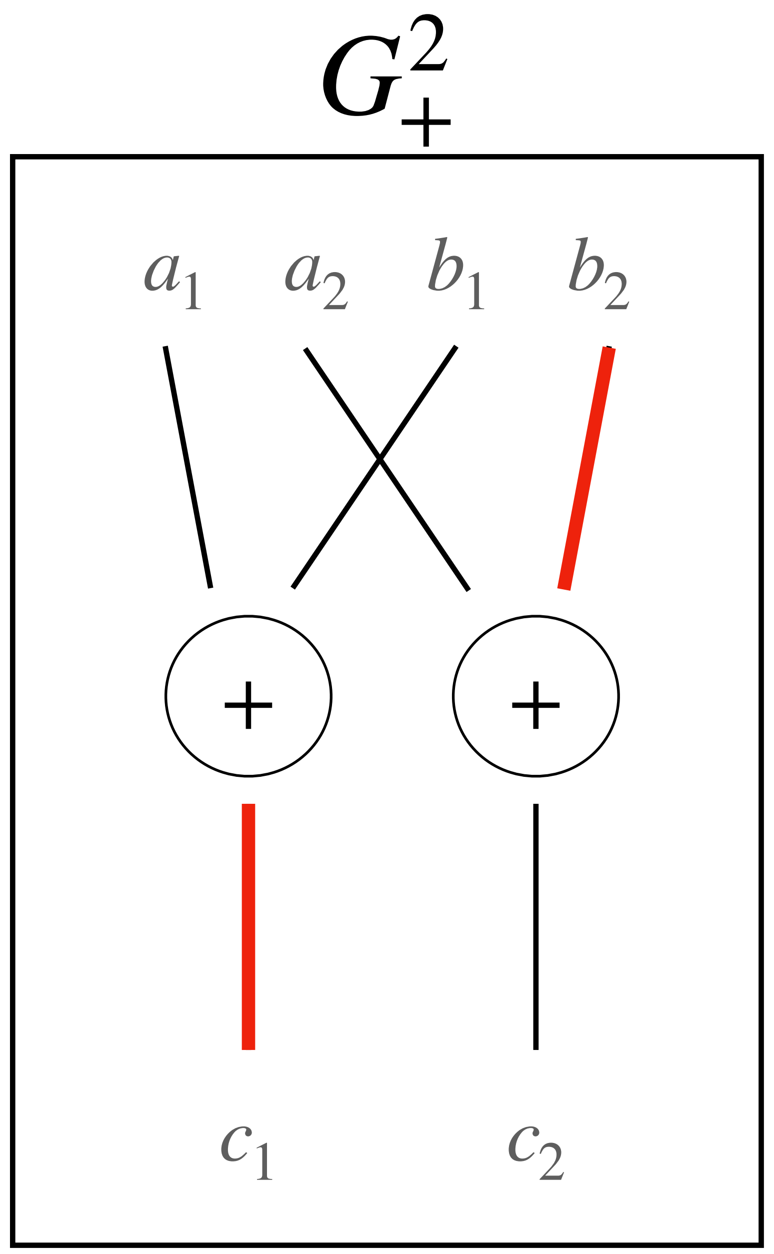
Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Example $t = 1, n = 2$:

output c_1 and internal wire $W = \{b_2\}$ together reveal information about b



Ability to simulate any set of internal wires **and t output shares** using at most t input shares of each input

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|\,.\,\kappa^e), \qquad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

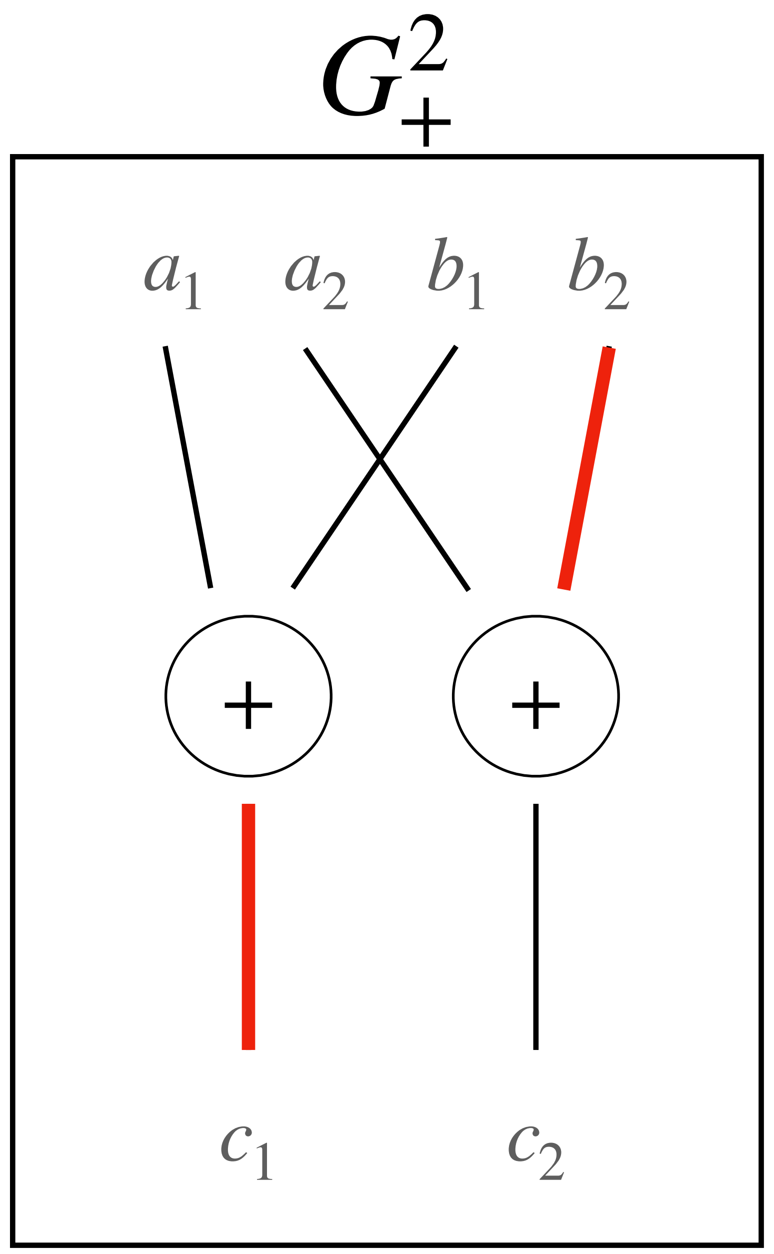
$\mathbf{N}_{\max} \approx \max(\#\times$ in $G_\times$, $\#(+,||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Example $t = 1, n = 2$:

output c_1 and internal wire $W = \{b_2\}$ together reveal information about b

W is a failure of 1 wire, d = 1



Ability to simulate any set of internal wires **and t output shares** using at most t input shares of each input

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C| . \kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, #$(+ , ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

$\varepsilon = f(p) = c_d p^d (1-p)^{s-d} + \mathcal{O}(p^{d+1})$

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C| . \kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

$\varepsilon = f(p) = c_d p^d (1-p)^{s-d} + \mathcal{O}(p^{d+1})$

during expansion, $\varepsilon^k = f^k(p) = f(f(\ldots f(p)\ldots))$

Random Probing Expansion

Complexity *Belaïd, Coron, Prouff, Rivain, Taleb [CRYPTO'20]*

Expanding a circuit C to achieve a desired security level κ costs

$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{\log(\mathbf{N}_{\max})}{\log(\mathbf{d})}$$

Considering practical cases with $G_+, G_\times, G_{||}$

$\mathbf{N}_{\max} \approx \max(\# \times$ in $G_\times$, $\#(+, ||)$ in $G_+, G_{||})$

$\mathbf{d} =$ smallest failure set of internal wires (amplification order)

$\varepsilon = f(p) = c_d p^d (1-p)^{s-d} + \mathcal{O}(p^{d+1})$

during expansion, $\varepsilon^k = f^k(p) = f(f(\ldots f(p)\ldots))$

higher amplification order $d \Longrightarrow$ faster decrease in failure probability ($d_{\max} = \dfrac{n+1}{2}$)

Random Probing Expansion

Results Overview

Random Probing Expansion

Results Overview

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based		
[BCPRT CRYPTO'20] 3-share		
[BelRivTal EUROCRYPT'21] 3-share		
[BelRivTal EUROCRYPT'21] 5-share		
[BRTV ASIACRYPT'21] theoretical		

Random Probing Expansion

Results Overview

Maximum probability such that $\varepsilon < p_{\max}$

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based		
[BCPRT CRYPTO'20] 3-share		
[BelRivTal EUROCRYPT'21] 3-share		
[BelRivTal EUROCRYPT'21] 5-share		
[BRTV ASIACRYPT'21] theoretical		

Random Probing Expansion

Results Overview

Maximum probability such that $\varepsilon < p_{\max}$

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based		
[BCPRT CRYPTO'20] 3-share		
[BelRivTal EUROCRYPT'21] 3-share		
[BelRivTal EUROCRYPT'21] 5-share		
[BRTV ASIACRYPT'21] theoretical		

All values are computed using automatic verification tools

Random Probing Expansion

Results Overview

Maximum probability such that $\varepsilon < p_{max}$

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based	$\mathcal{O}(\lvert C \rvert . \kappa^{7.87})$	$p_{max} = 2^{-25}$
[BCPRT CRYPTO'20] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{7.5})$	$p_{max} = 2^{-8}$
[BelRivTal EUROCRYPT'21] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.9})$	$p_{max} = 2^{-7.5}$
[BelRivTal EUROCRYPT'21] 5-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.2})$	$2^{-9.67} < p_{max} < 2^{-7.66}$
[BRTV ASIACRYPT'21] theoretical	$\approx \mathcal{O}(\lvert C \rvert . \kappa^{2})$	–

All values are computed using automatic verification tools

Random Probing Expansion

Results Overview

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based	$\mathcal{O}(\lvert C \rvert . \kappa^{7.87})$	$p_{\max} = 2^{-25}$
[BCPRT CRYPTO'20] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{7.5})$	$p_{\max} = 2^{-8}$
[BelRivTal EUROCRYPT'21] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.9})$	$p_{\max} = 2^{-7.5}$
[BelRivTal EUROCRYPT'21] 5-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.2})$	$2^{-9.67} < p_{\max} < 2^{-7.66}$
[BRTV ASIACRYPT'21] theoretical	$\approx \mathcal{O}(\lvert C \rvert . \kappa^{2})$	–

Maximum probability such that $\varepsilon < p_{\max}$

All values are computed using automatic verification tools

Gadgets too large to have precise results by current tools

Random Probing Expansion

Results Overview

Maximum probability such that $\varepsilon < p_{max}$

Construction	Complexity	Tolerated Leakage rate
[AIS CRYPTO'18] MPC based	$\mathcal{O}(\lvert C \rvert . \kappa^{7.87})$	$p_{max} = 2^{-25}$
[BCPRT CRYPTO'20] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{7.5})$	$p_{max} = 2^{-8}$
[BelRivTal EUROCRYPT'21] 3-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.9})$	$p_{max} = 2^{-7.5}$
[BelRivTal EUROCRYPT'21] 5-share	$\mathcal{O}(\lvert C \rvert . \kappa^{3.2})$	$2^{-9.67} < p_{max} < 2^{-7.66}$
[BRTV ASIACRYPT'21] theoretical	$\approx \mathcal{O}(\lvert C \rvert . \kappa^{2})$	–

All values are computed using automatic verification tools

Gadgets too large to have precise results by current tools

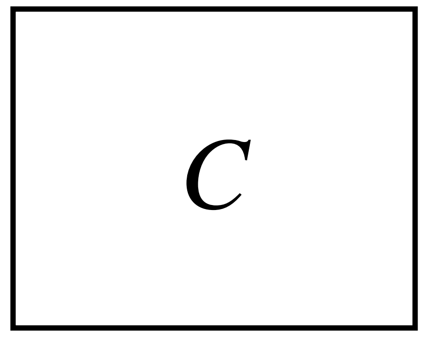
theoretical construction on large fields, not taken into account by current tools

Automatic Verification Tools

Goal

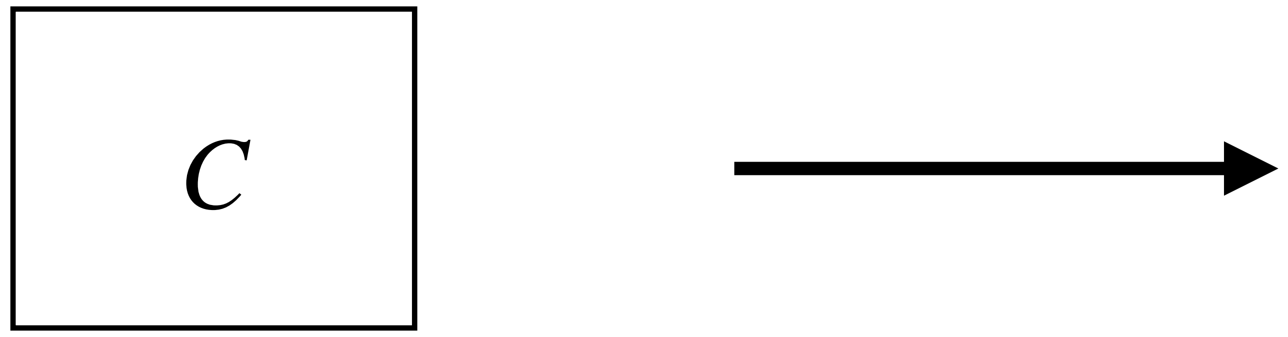
Automatic Verification Tools

Goal



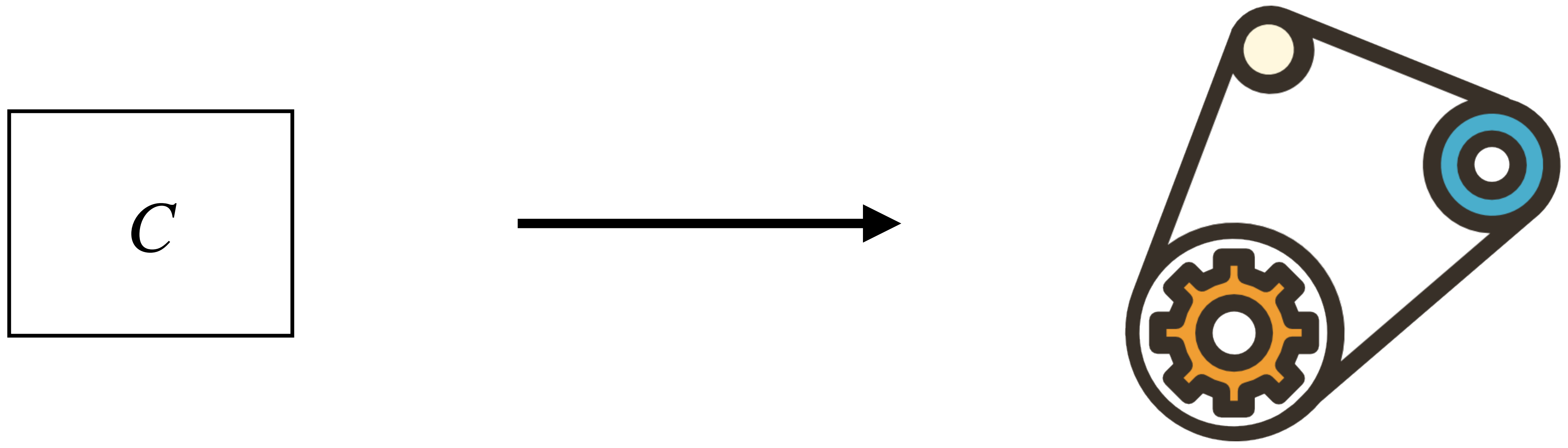
Automatic Verification Tools

Goal



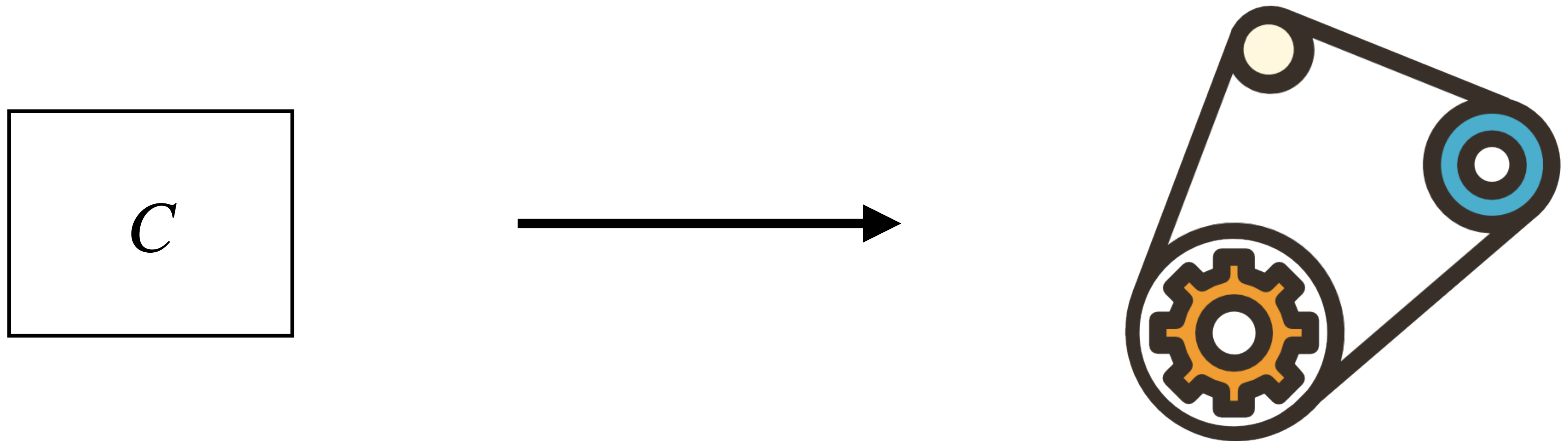
Automatic Verification Tools

Goal



Automatic Verification Tools

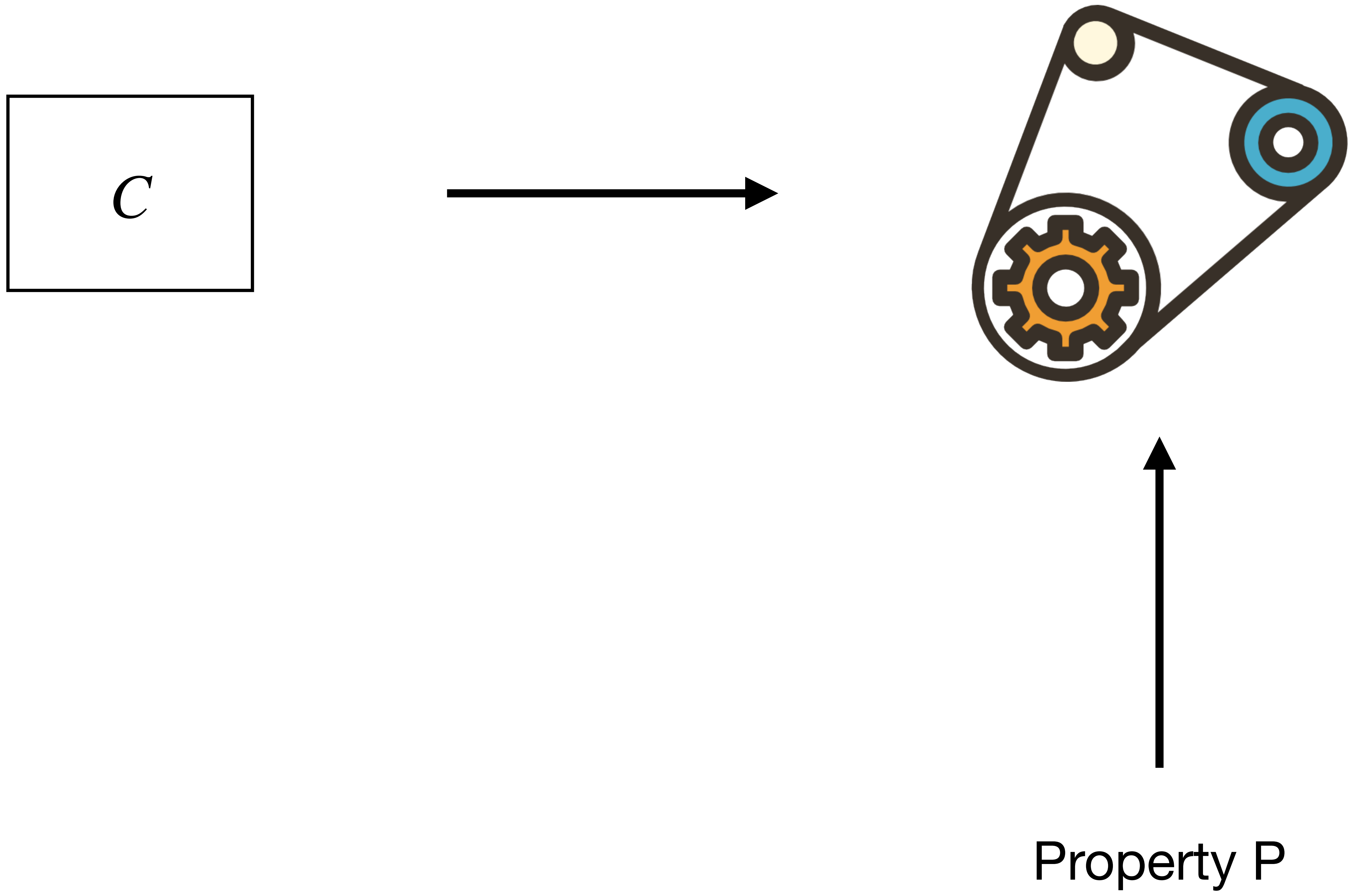
Goal



Property P

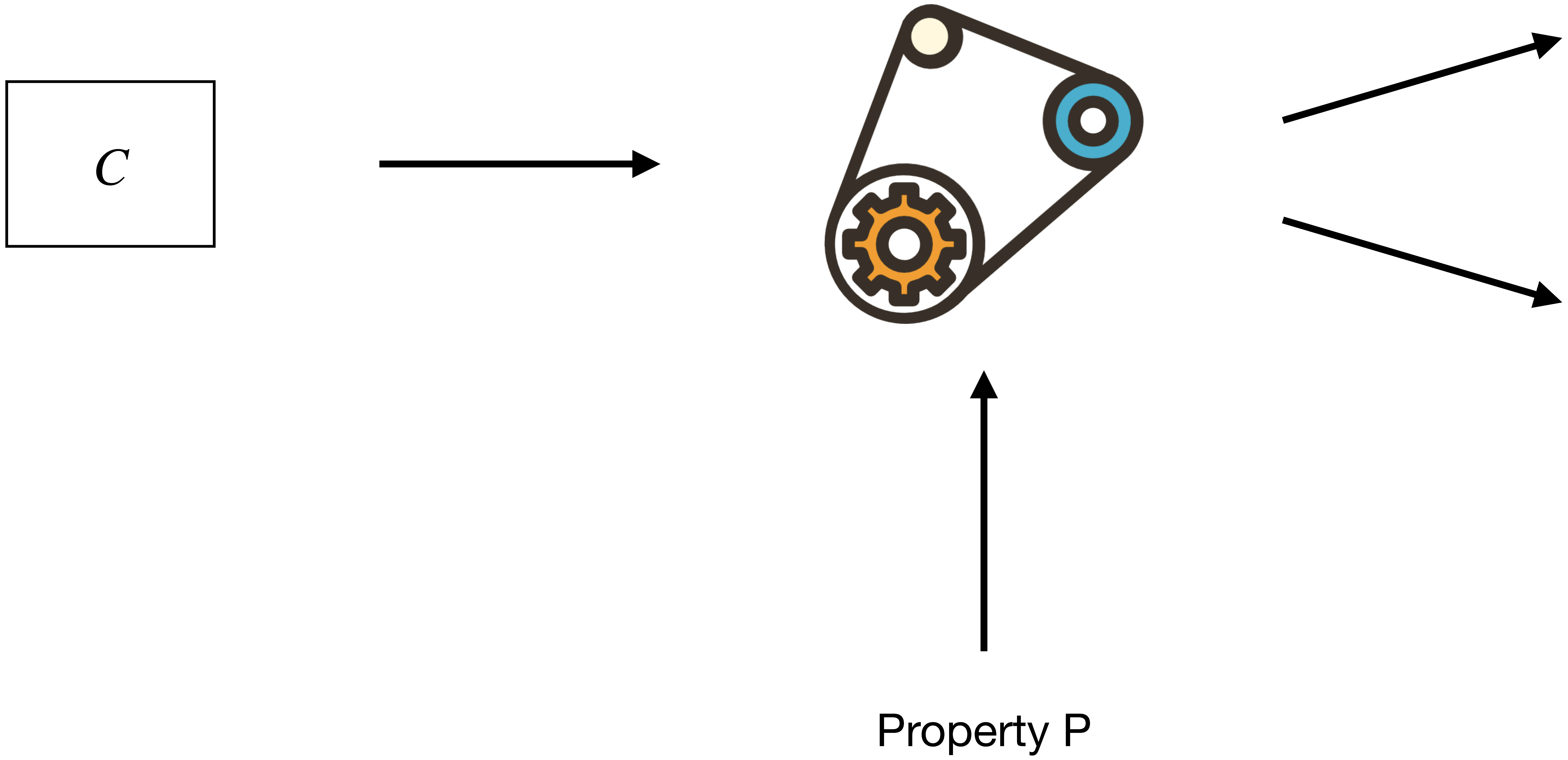
Automatic Verification Tools

Goal



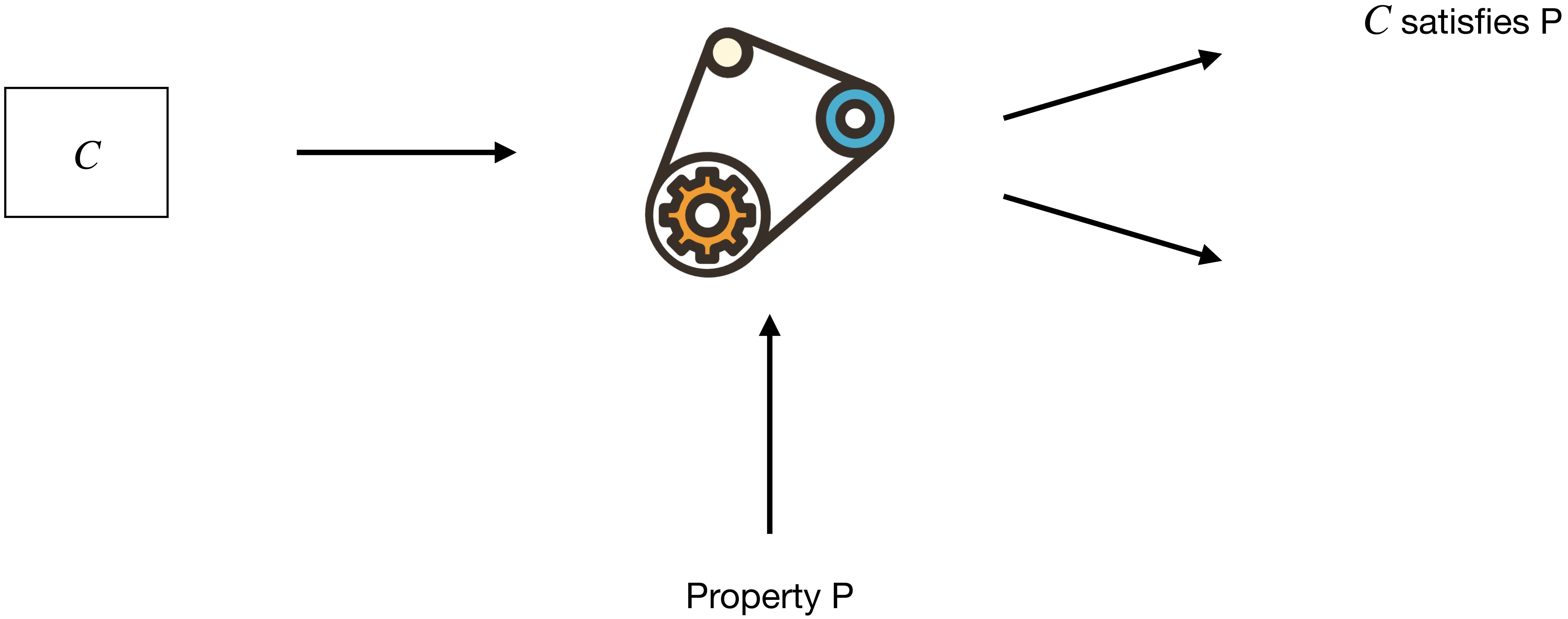
Automatic Verification Tools

Goal



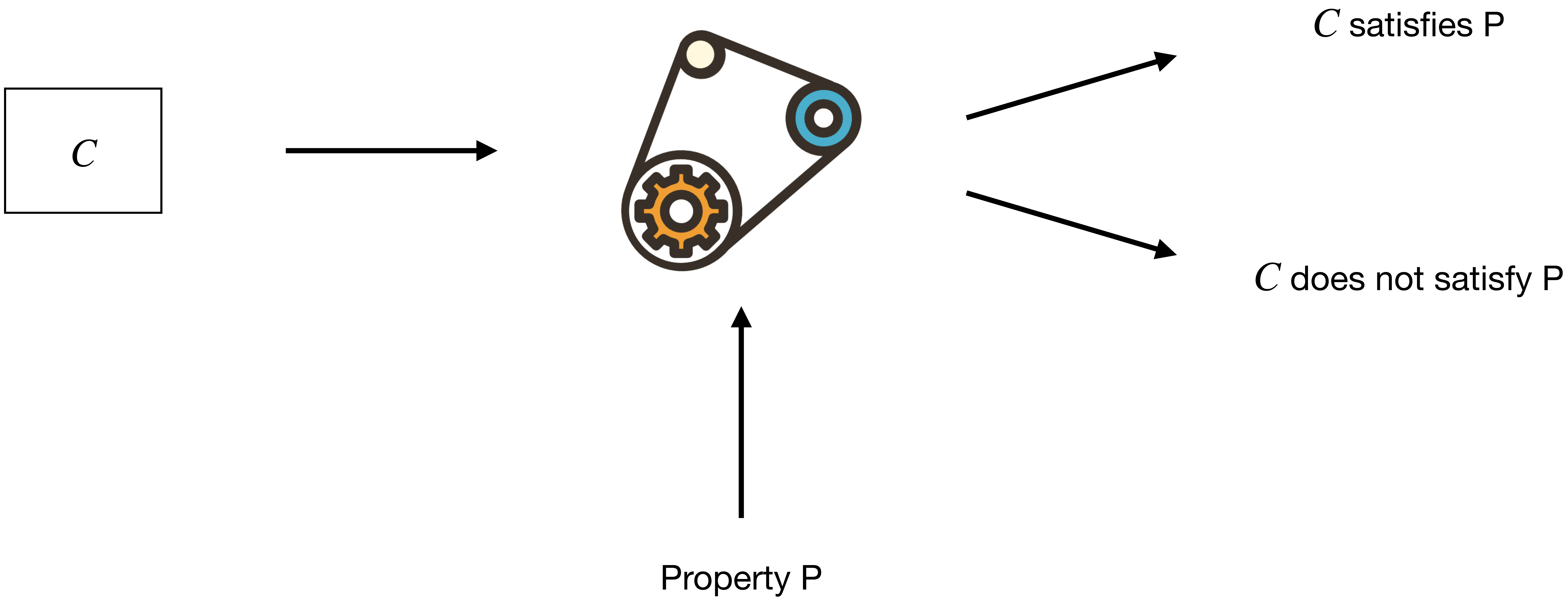
Automatic Verification Tools

Goal



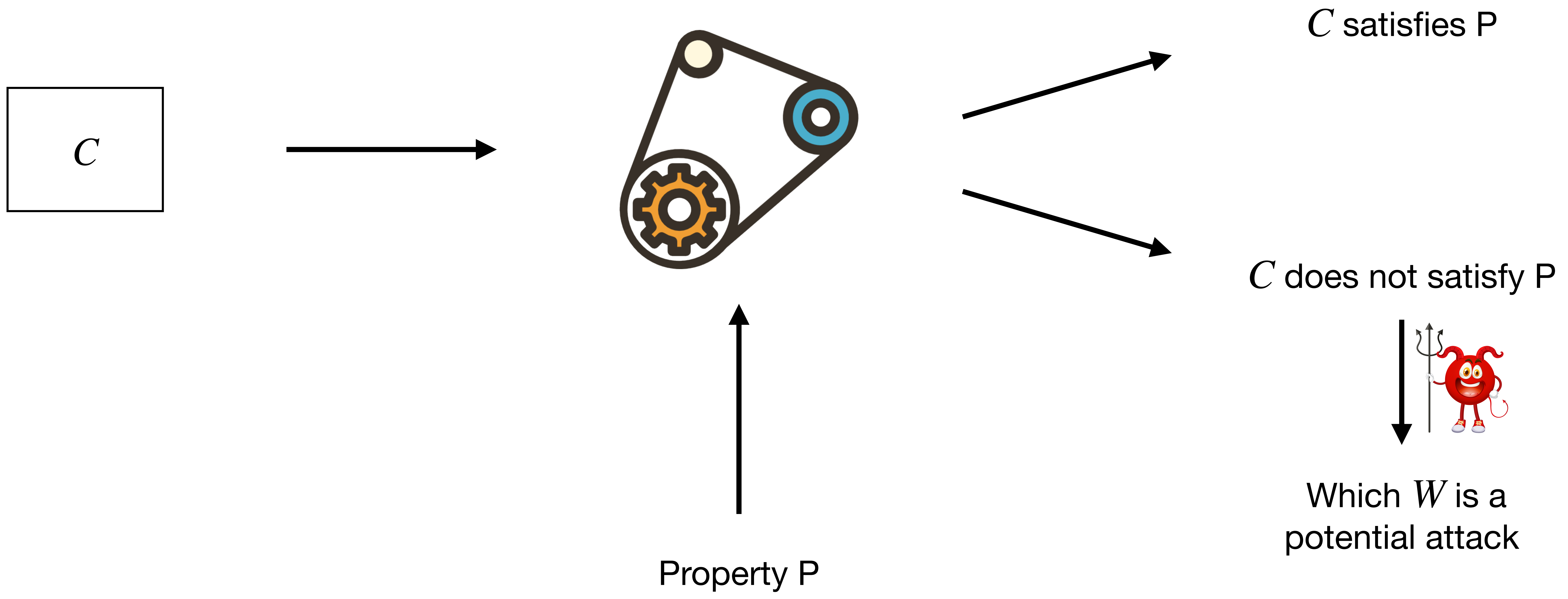
Automatic Verification Tools

Goal



Automatic Verification Tools

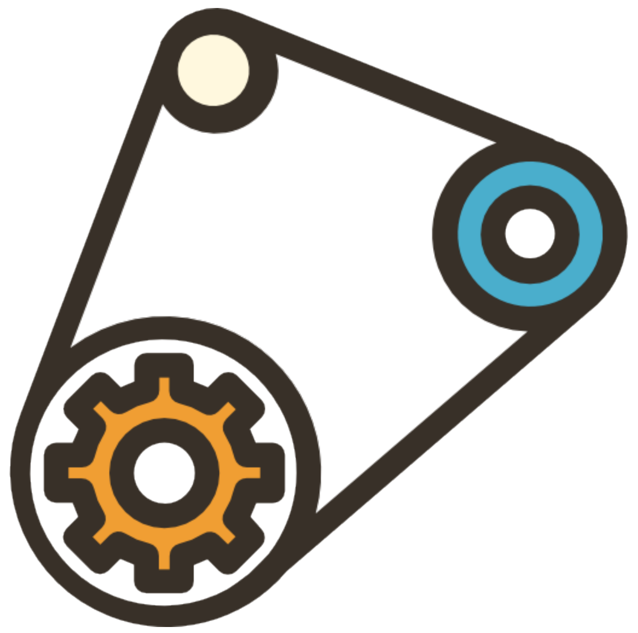
Goal



Automatic Verification Tools

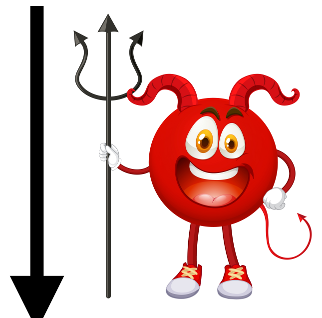
Goal

C



C satisfies P

C does not satisfy P



Which W is a potential attack

Property P

Probing

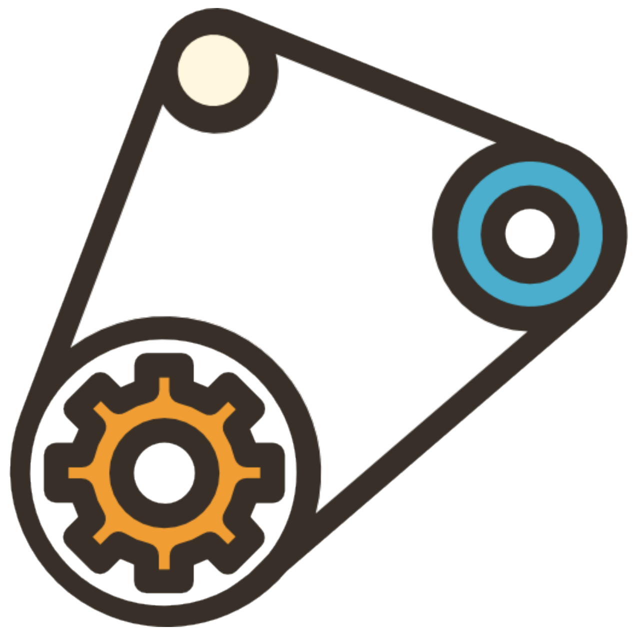
- t-NI
- t-SNI
- t-PINI

…

Automatic Verification Tools

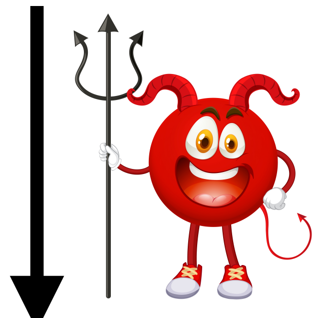
Goal

C



C satisfies P

C does not satisfy P



Which W is a potential attack

Property P

Probing
- t-NI
- t-SNI
- t-PINI

…

Random Probing
- (p, ε)- RP security
- (p, ε)- RP composition
- (t, p, ε)- RP expansion

…

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block

- Exact verification method for almost all gadgets in the state-of-the-art

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block
- Exact verification method for almost all gadgets in the state-of-the-art
- IronMask: an **exact** verification tool for **(all)** probing and random probing properties

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block
- Exact verification method for almost all gadgets in the state-of-the-art
- IronMask: an **exact** verification tool for **(all)** probing and random probing properties

Tool	Properties		Fast Verification
	Probing	Random Probing	
SILVER	✔ (green)	✘	✘
MaskVerif	✔ (orange)	✘	✔ (green)
MatVerif	✔ (orange)	✘	✔ (green)
VRAPS	✔ (orange)	✔ (orange)	✘
STRAPS	✘	✔ (orange)	✔ (green)
IronMask	✔ (green)	✔ (green)	✔ (green)

✔ (green) handled
✔ (orange) handled but inexact
✘ not handled

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

- Formalization of all of the probing and random probing properties from a single standard building block
- Exact verification method for almost all gadgets in the state-of-the-art
- IronMask: an **exact** verification tool for **(all)** probing and random probing properties

Tool	Properties		Fast Verification
	Probing	Random Probing	
SILVER	✔	✘	✘
MaskVerif	✓	✘	✔
MatVerif	✓	✘	✔
VRAPS	✓	✓	✘
STRAPS	✘	✓	✔
IronMask	✔	✔	✔

MatVerif → Limited to specific types of circuits

IronMask → Limited to specific types of circuits / Covers all gadgets in the state-of-the-art

✔ handled
✓ handled but inexact
✘ not handled

IronMask

Building Block for Security Properties

IronMask

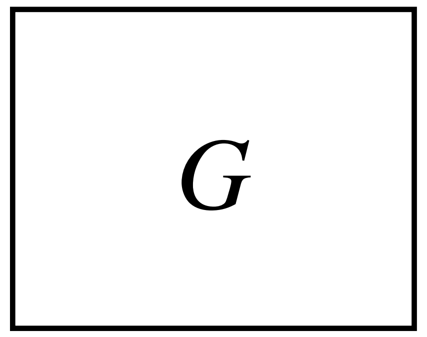
Building Block for Security Properties

Simulation based definitions of all (random) probing properties

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties



ℓ input sharings
1 output sharing

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

G

ℓ input sharings
1 output sharing

SIS

Sets of Input Shares

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

Set of internal probes W

G

ℓ input sharings
1 output sharing

SIS

Sets of Input Shares

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

Set of internal probes W

G

ℓ input sharings
1 output sharing

Set of output shares O

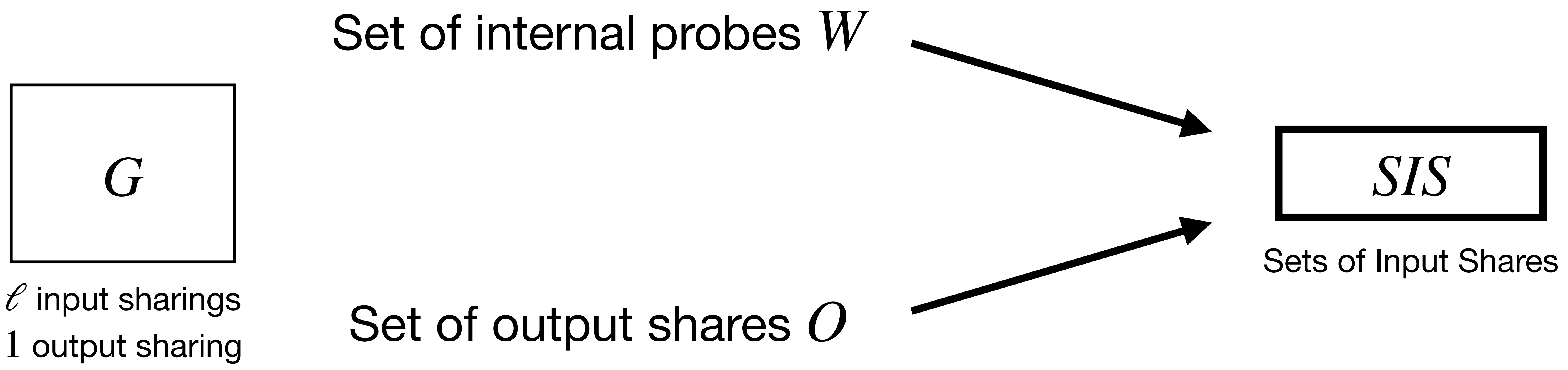
SIS

Sets of Input Shares

IronMask

Building Block for Security Properties

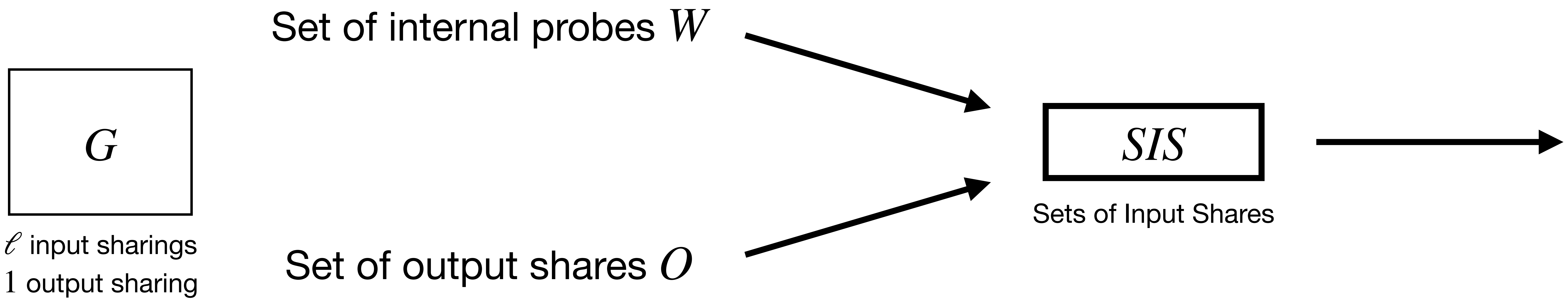
Simulation based definitions of all (random) probing properties



IronMask

Building Block for Security Properties

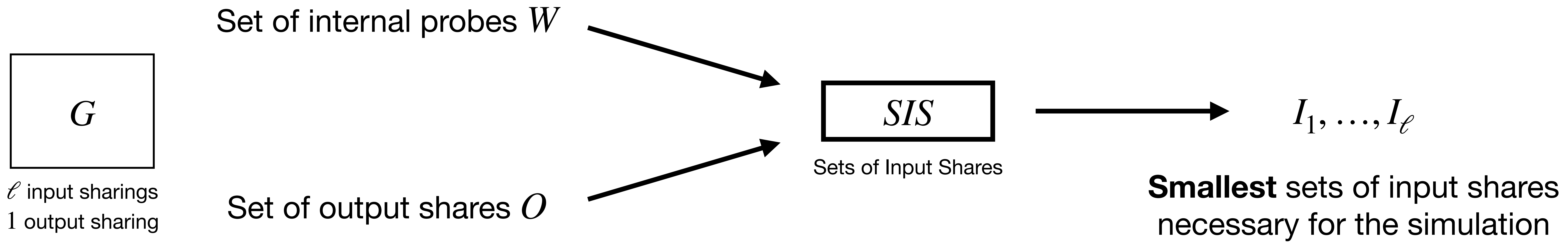
Simulation based definitions of all (random) probing properties



IronMask

Building Block for Security Properties

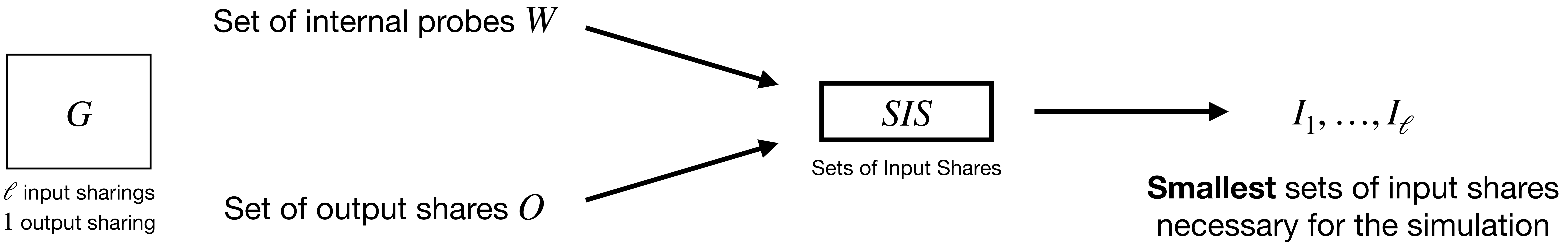
Simulation based definitions of all (random) probing properties



IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties



$$SIS_G(W, O) = (I_1, \ldots, I_\ell)$$

IronMask

Building Block for Security Properties

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$$SIS_G(W, O) = (I_1, \ldots, I_\ell)$$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$$SIS_G(W, O) = (I_1, \ldots, I_\ell)$$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$SIS_G(W, O) = (I_1, \ldots, I_\ell)$

t-NI: $\forall W, O, |W| + |O| \leq t, \quad |I_i| \leq t, \quad \forall i \in [1 : \ell]$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$SIS_G(W, O) = (I_1, \ldots, I_\ell)$

t-NI: $\forall W, O, |W| + |O| \leq t, \ \ |I_i| \leq t, \ \ \forall i \in [1 : \ell]$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$SIS_G(W, O) = (I_1, \ldots, I_\ell)$

t-NI: $\forall W, O, |W| + |O| \leq t, \quad |I_i| \leq t, \quad \forall i \in [1 : \ell]$

(t, p, ε)-RP composability: $\forall W, O, |O| \leq t, \quad Pr\big(|I_1| > t \vee \ldots \vee |I_\ell| > t \big) \leq \varepsilon$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$SIS_G(W, O) = (I_1, \dots, I_\ell)$

t-NI: $\forall W, O, |W| + |O| \leq t, \;\; |I_i| \leq t, \;\; \forall i \in [1 : \ell]$

(t, p, ε)-RP composability: $\forall W, O, |O| \leq t, \;\; Pr\Big(|I_1| > t \vee \dots \vee |I_\ell| > t \Big) \leq \varepsilon$

IronMask

Building Block for Security Properties

Simulation based definitions of all (random) probing properties

$SIS_G(W, O) = (I_1, \ldots, I_\ell)$

t-NI: $\forall W, O, |W| + |O| \leq t, \;\; |I_i| \leq t, \;\; \forall i \in [1 : \ell]$

(t, p, ε)-RP composability: $\forall W, O, |O| \leq t, \;\; Pr\big(\, |I_1| > t \vee \ldots \vee |I_\ell| > t\big) \leq \varepsilon$

$\ldots$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness

$\overrightarrow{x_1}$
$\vdots$
$\overrightarrow{x_\ell}$

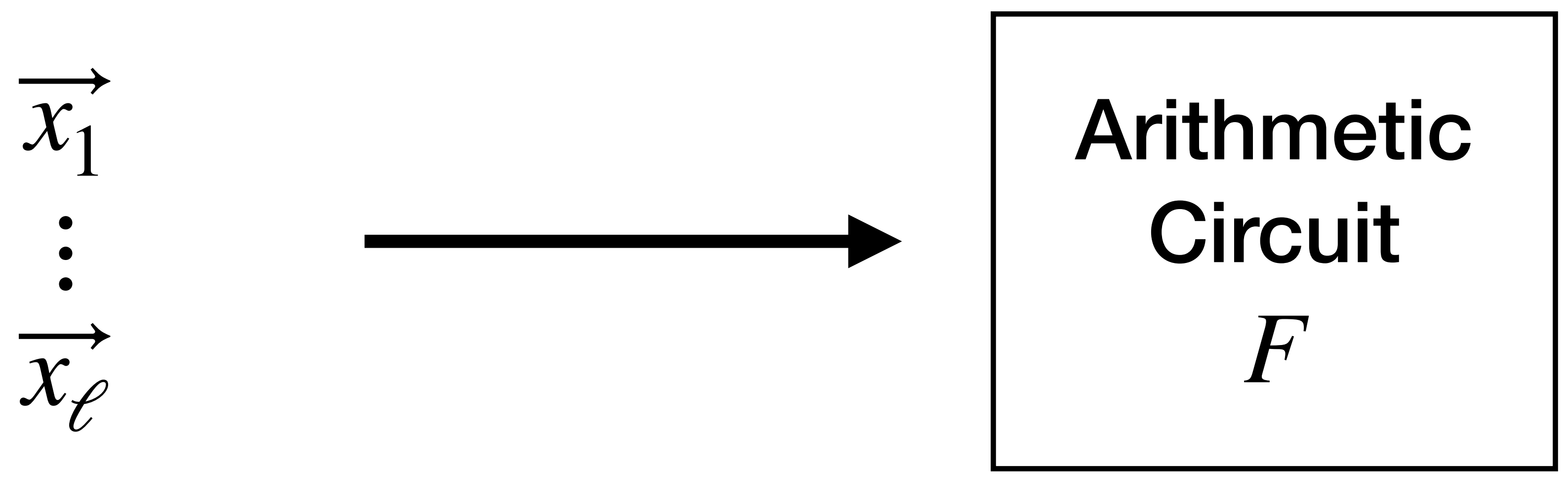
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness

$$\begin{matrix} \overrightarrow{x_1} \\ \vdots \\ \overrightarrow{x_\ell} \end{matrix} \quad \longrightarrow$$

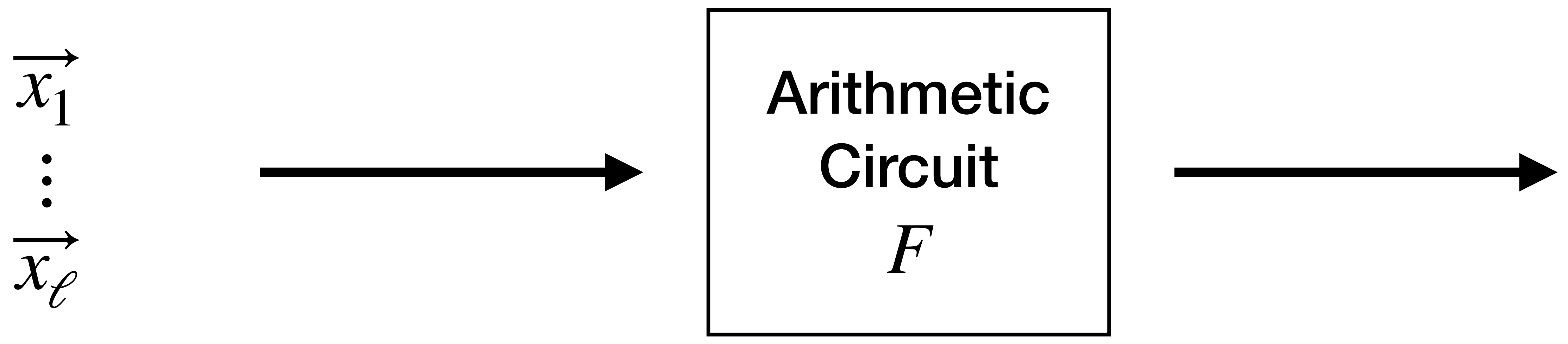
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



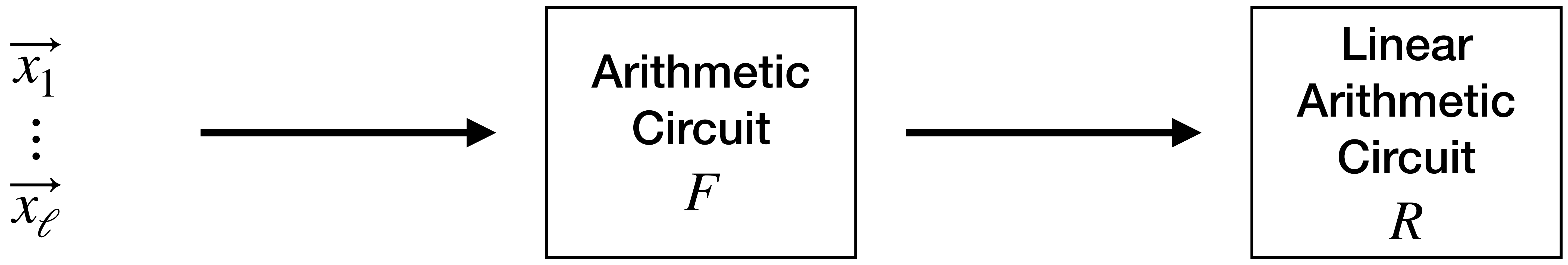
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



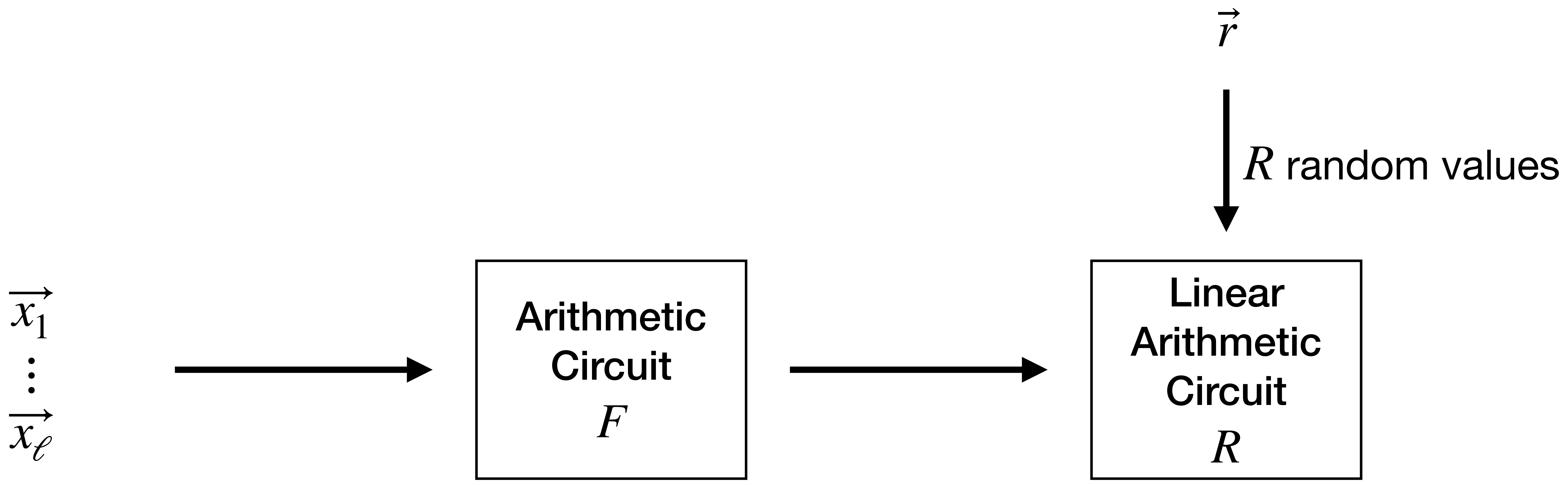
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



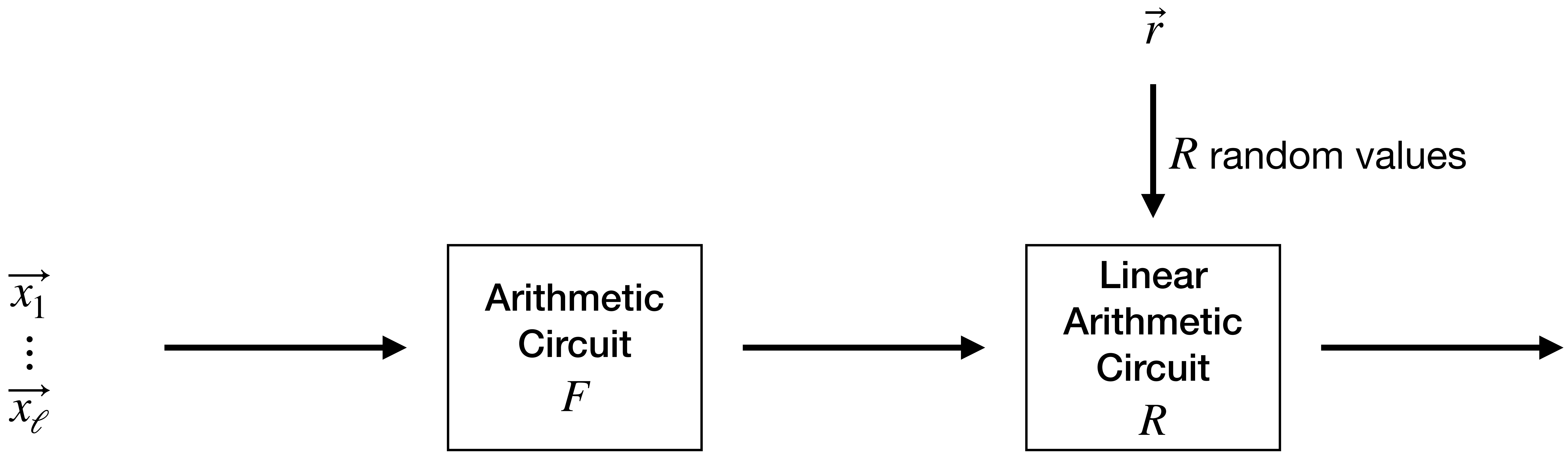
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



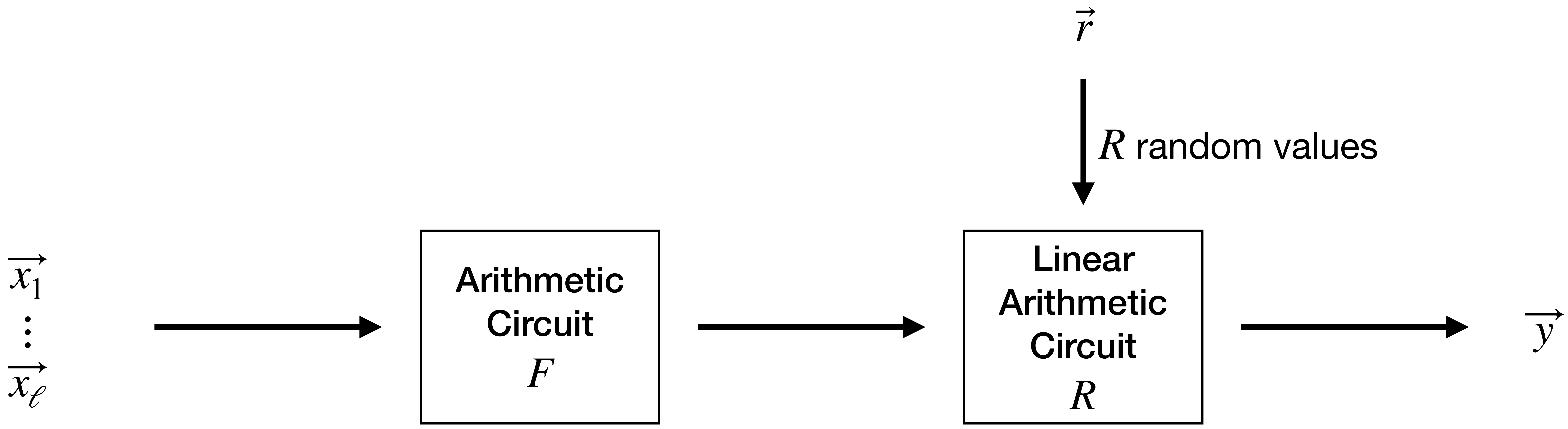
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



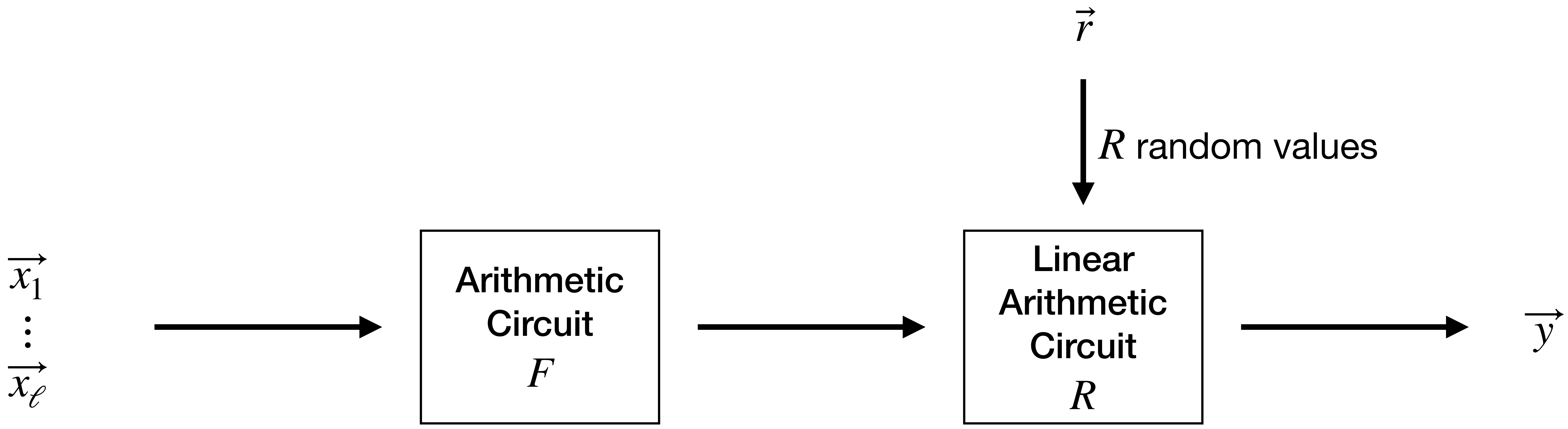
Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



Algebraic Characterization of Gadgets

Gadgets with Linear Randomness

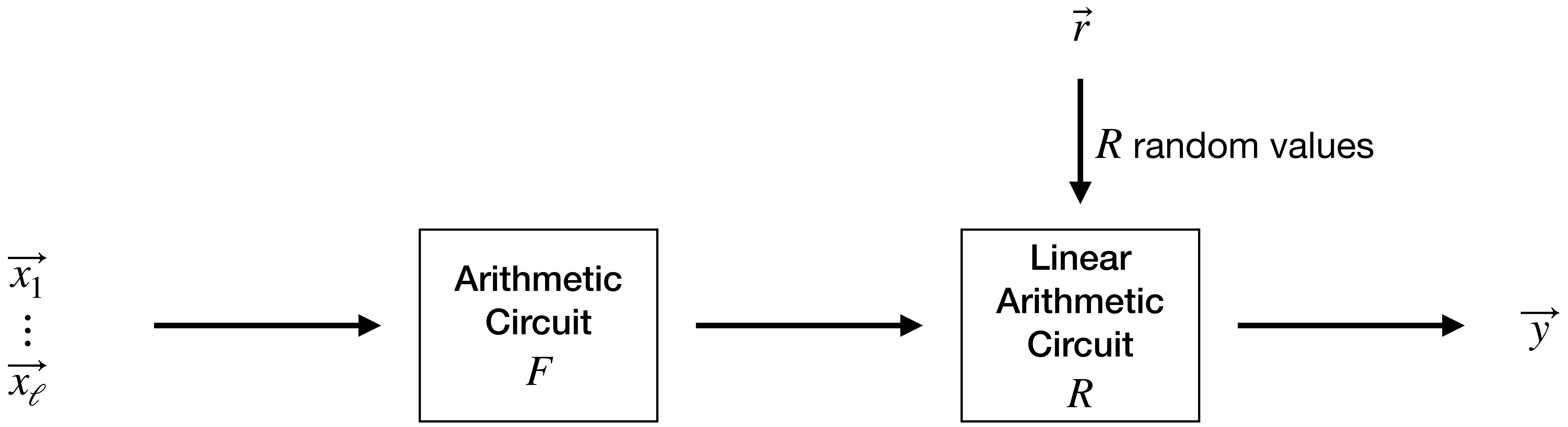


Probe p on such a gadget

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness



Probe p on such a gadget

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

→ Coefficient i of $\overrightarrow{s_p}$ is 1 if p contains r_i

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix}$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix}$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow \quad S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

$$p'_i = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{0}$$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

$$p'_i = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \vec{0}$$

To perfectly simulate probes in $\overrightarrow{W}$, we need exactly the input shares appearing in $(p'_1, \ldots, p'_m)$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

$$p'_i = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \vec{0}$$

To perfectly simulate probes in $\overrightarrow{W}$, we need exactly the input shares appearing in $(p'_1, \ldots, p'_m)$ $\longrightarrow$

Algebraic Characterization of Gadgets

Gadgets with Linear Randomness: Exact Verification

$$p = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Vector of probes $\overrightarrow{W} = (p_1, \ldots, p_k)$

$$S = \begin{pmatrix} \overrightarrow{s_{p_1}} \\ \vdots \\ \overrightarrow{s_{p_k}} \end{pmatrix} \longrightarrow S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix} \longrightarrow \overrightarrow{W'} = N \cdot \overrightarrow{W} = (p'_1, \ldots, p'_m, p'_{m+1}, \ldots, p'_d)$$

simulated from uniform random values

$$p'_i = f_p(\overrightarrow{x_1}, \ldots, \overrightarrow{x_\ell}) + \vec{r}^T \cdot \vec{0}$$

To perfectly simulate probes in $\overrightarrow{W}$, we need exactly the input shares appearing in $(p'_1, \ldots, p'_m) \longrightarrow (I_1, \ldots, I_\ell)$

Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)

Algebraic Characterization of Gadgets

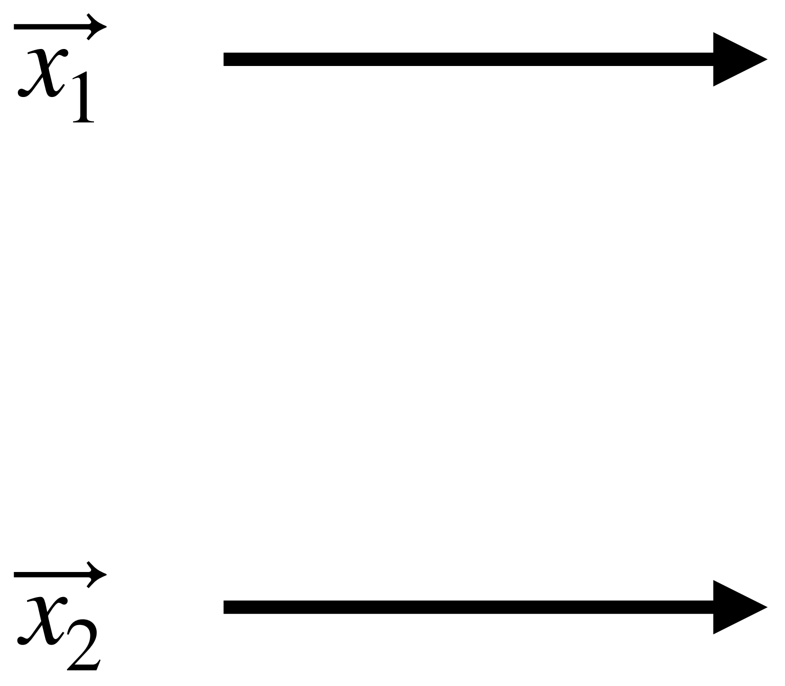
Gadgets with Non-Linear Randomness (2 inputs)

$\overrightarrow{x_1}$

$\overrightarrow{x_2}$

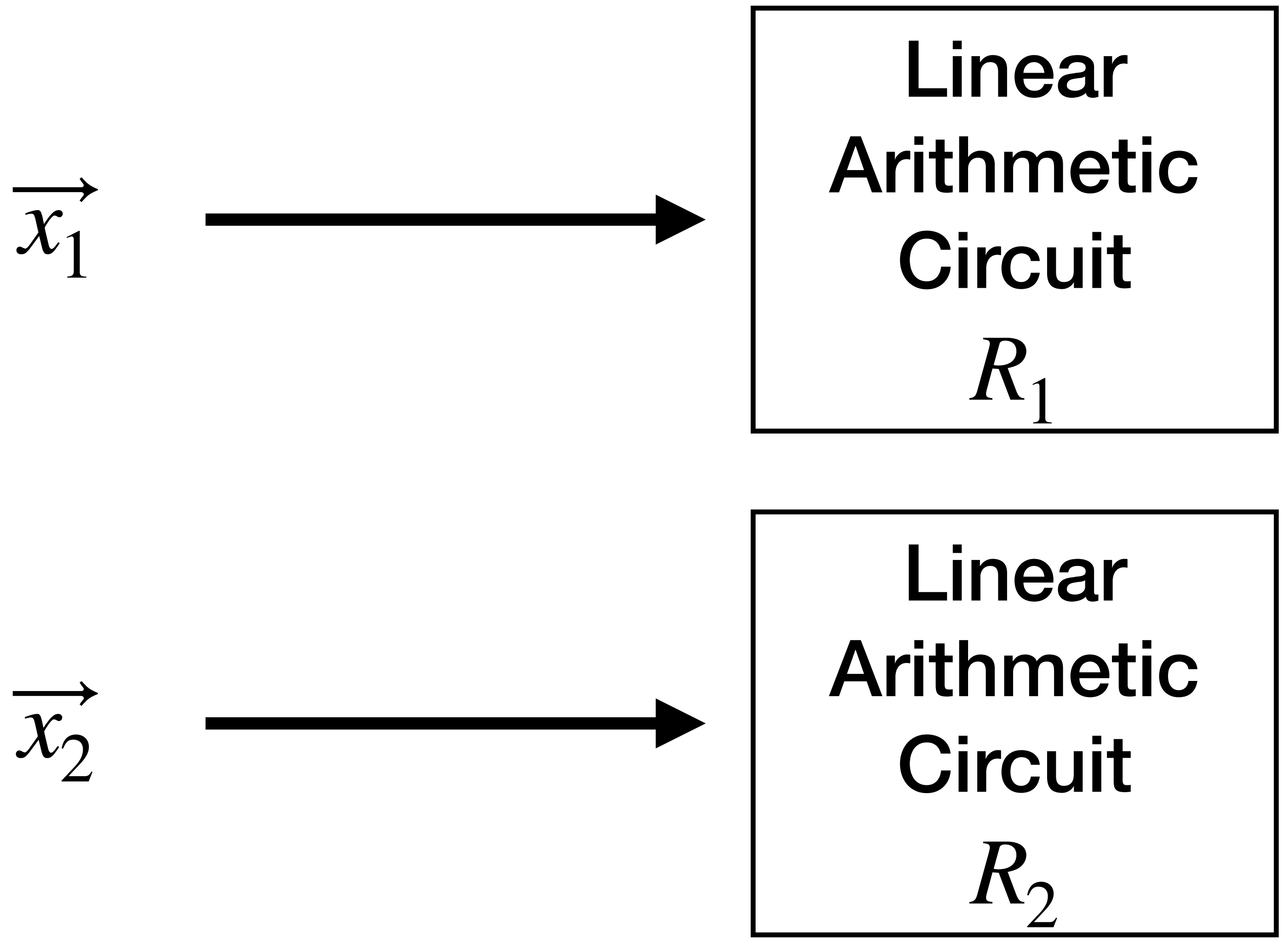
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



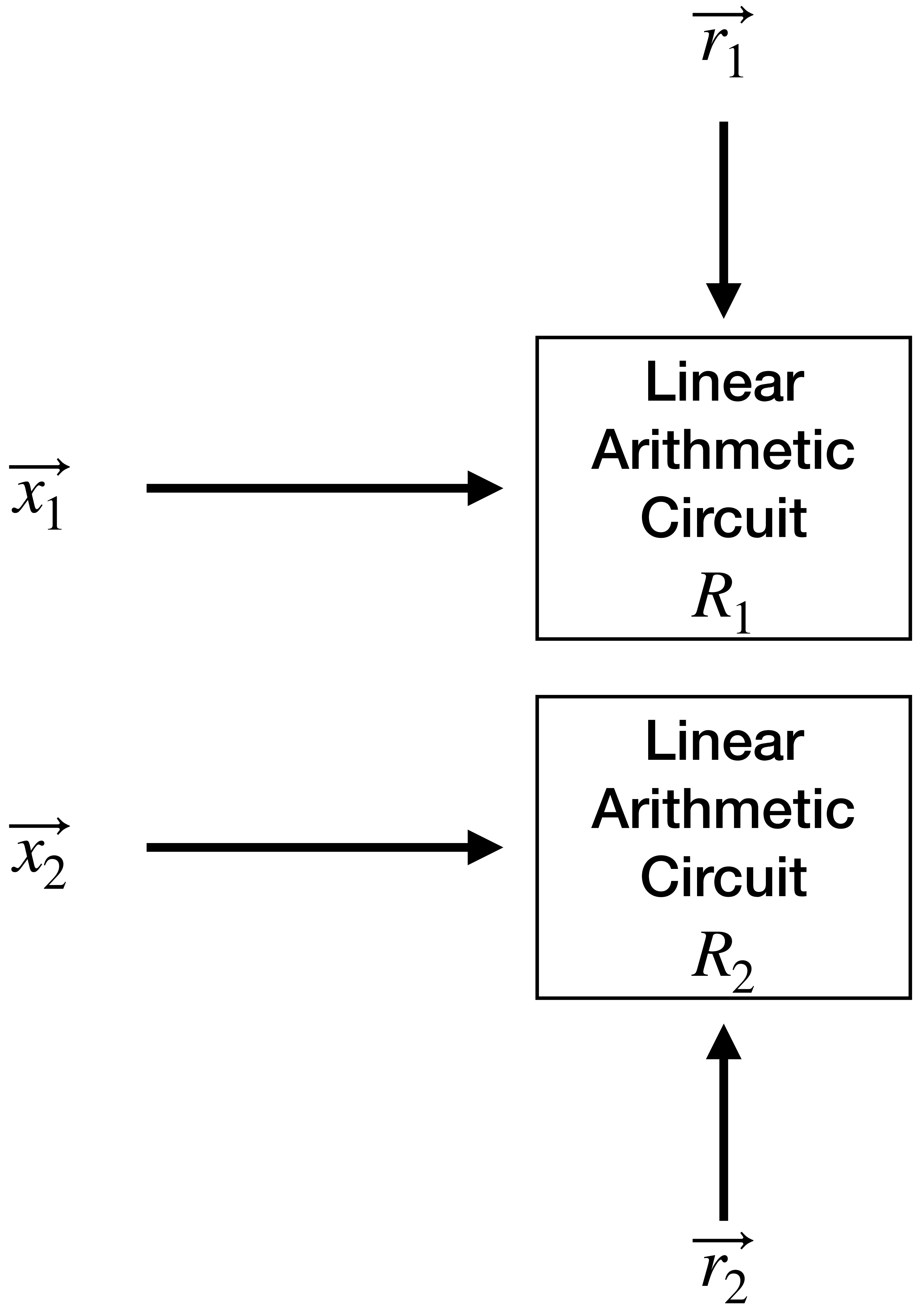
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



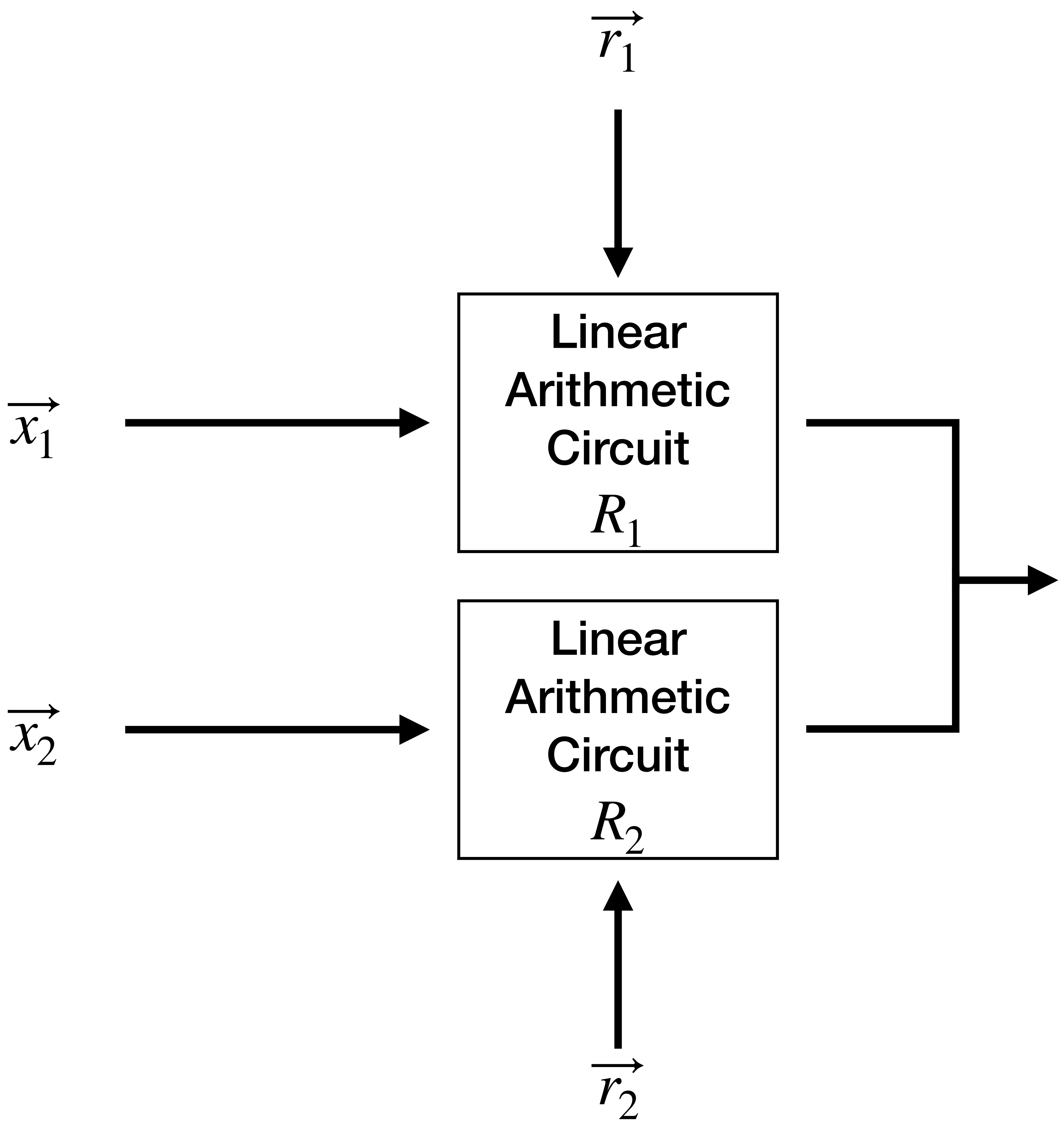
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



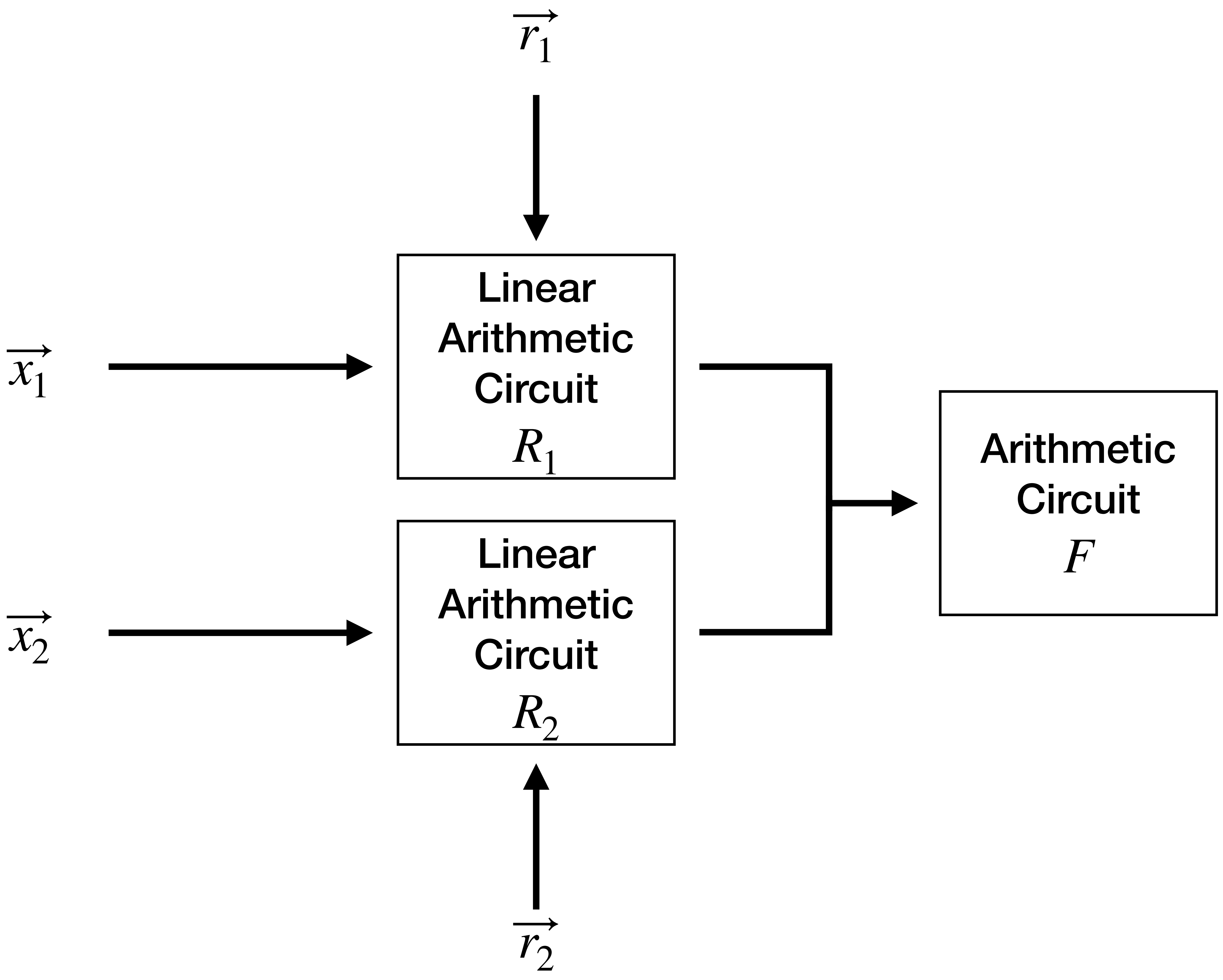
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



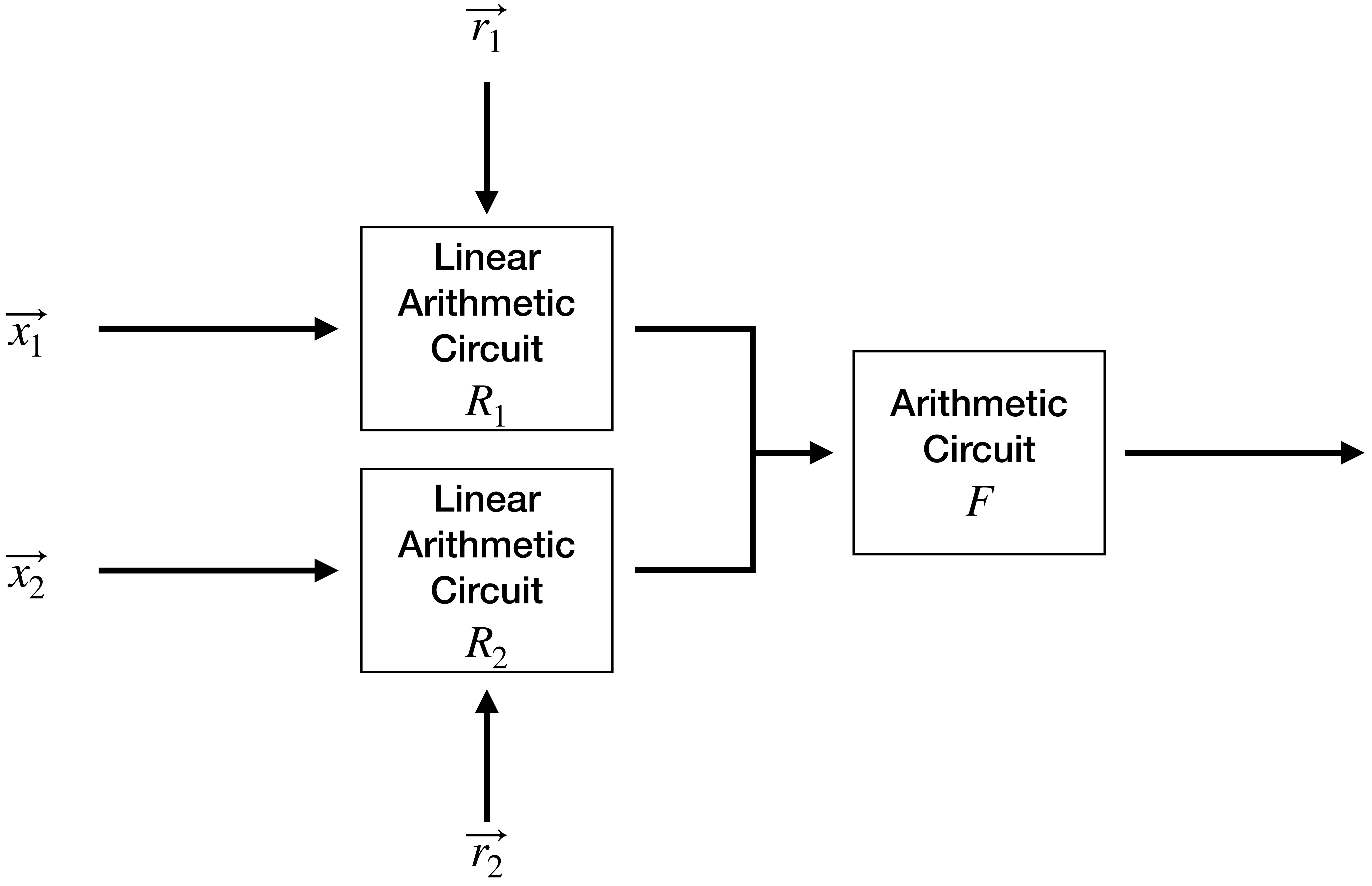
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



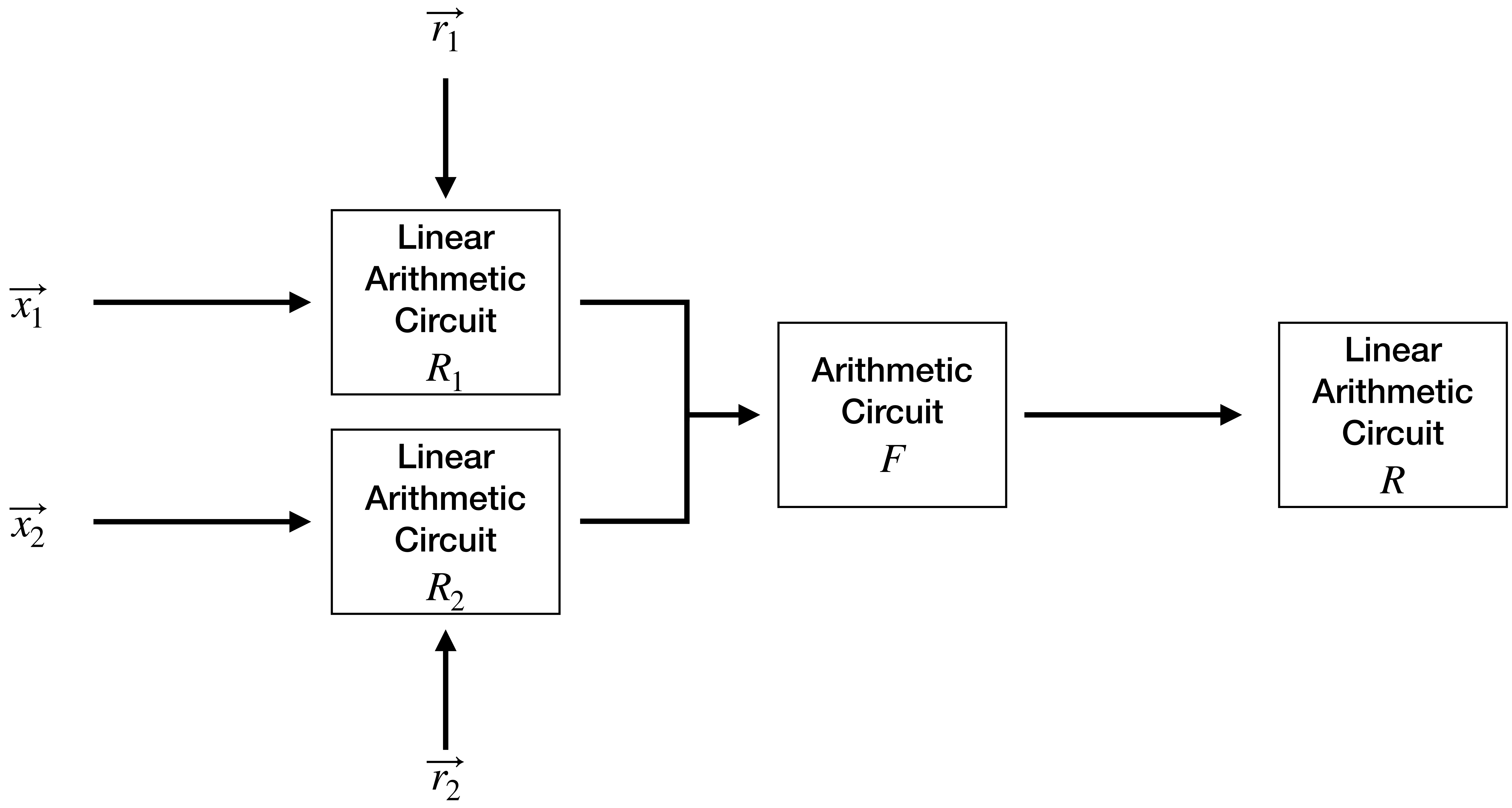
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



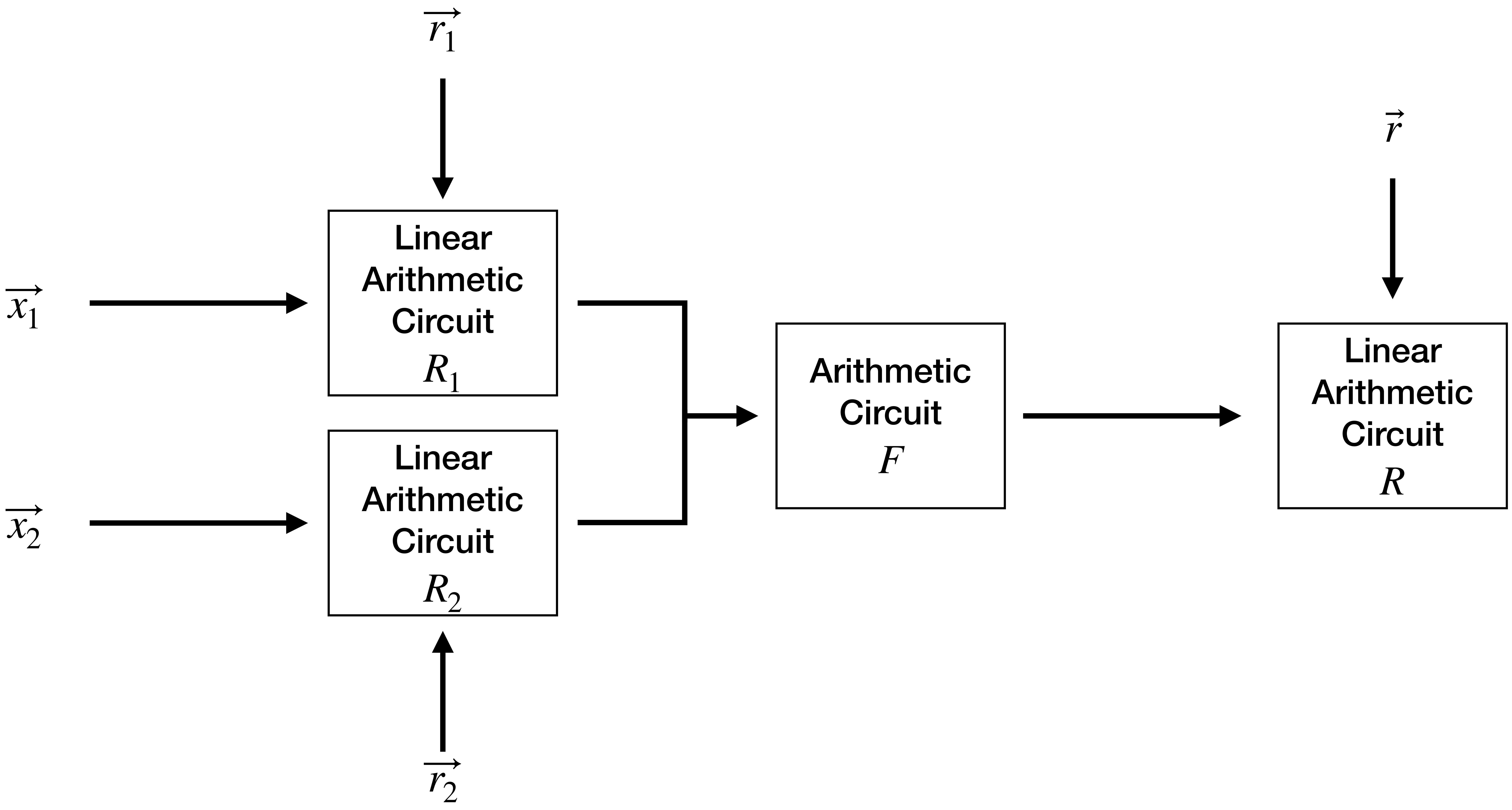
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



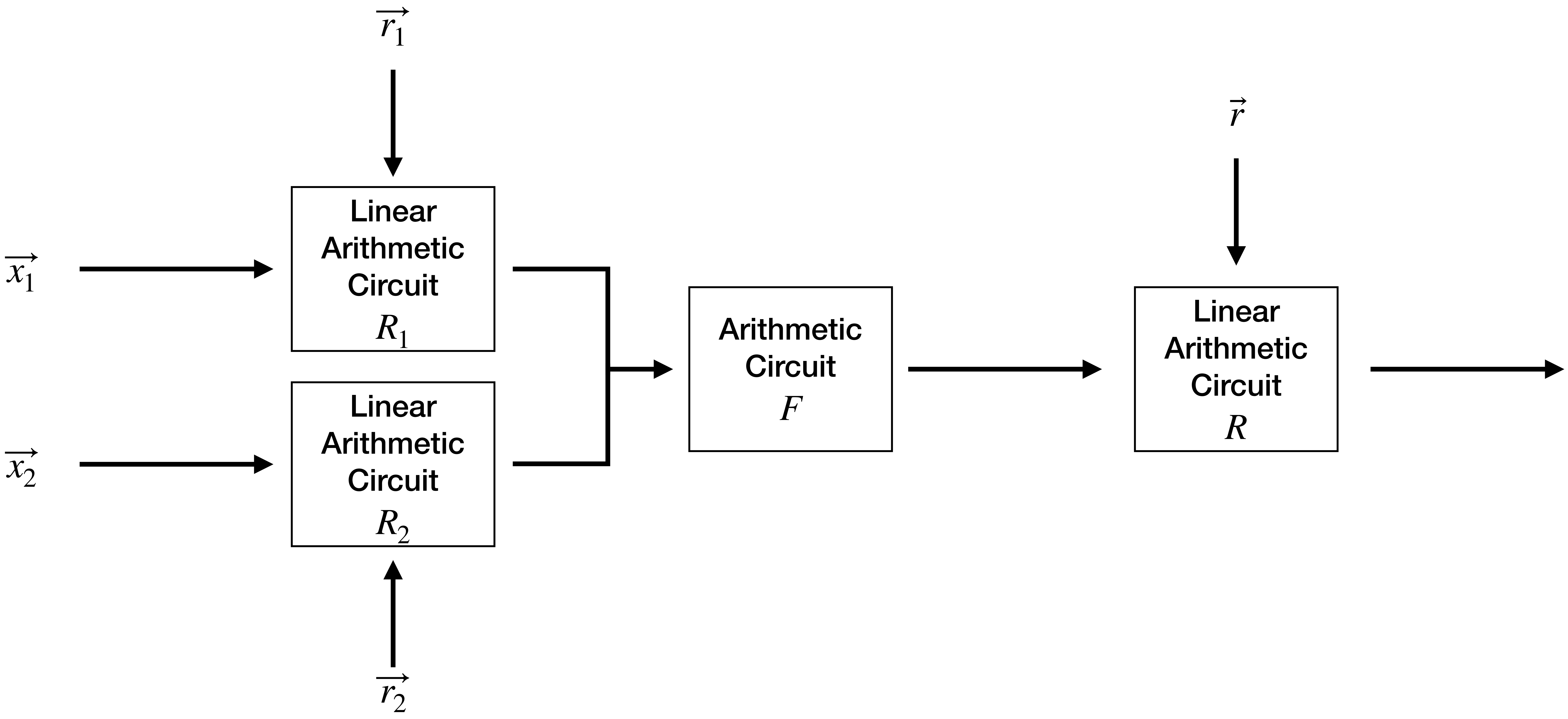
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



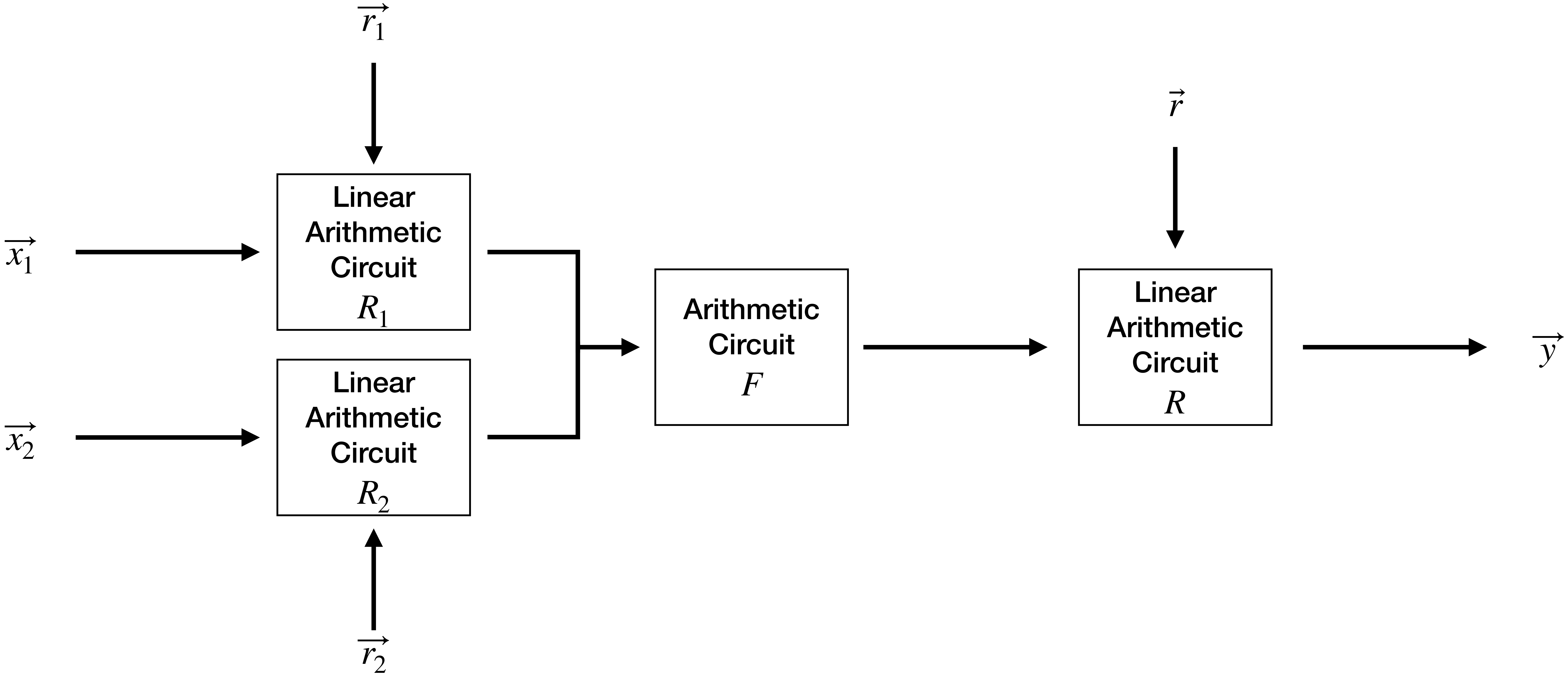
Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness (2 inputs)



Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness: Exact Verification

Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness: Exact Verification

Probe p on such a gadget

$$p = f_p\left(R_1(\overrightarrow{x_1}, \overrightarrow{r_1}), R_2(\overrightarrow{x_2}, \overrightarrow{r_2})\right) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Perform three row reductions

- First with respect to $\vec{r}$
- Then with respect to $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$

Algebraic Characterization of Gadgets

Gadgets with Non-Linear Randomness: Exact Verification

Probe p on such a gadget

$$p = f_p\Big(R_1(\overrightarrow{x_1}, \overrightarrow{r_1}), R_2(\overrightarrow{x_2}, \overrightarrow{r_2})\Big) + \vec{r}^T \cdot \overrightarrow{s_p}$$

Perform three row reductions

- First with respect to $\vec{r}$
- Then with respect to $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$

<u>Proven Result:</u> the strategy is an exact verification method for such gadgets

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

IronMask

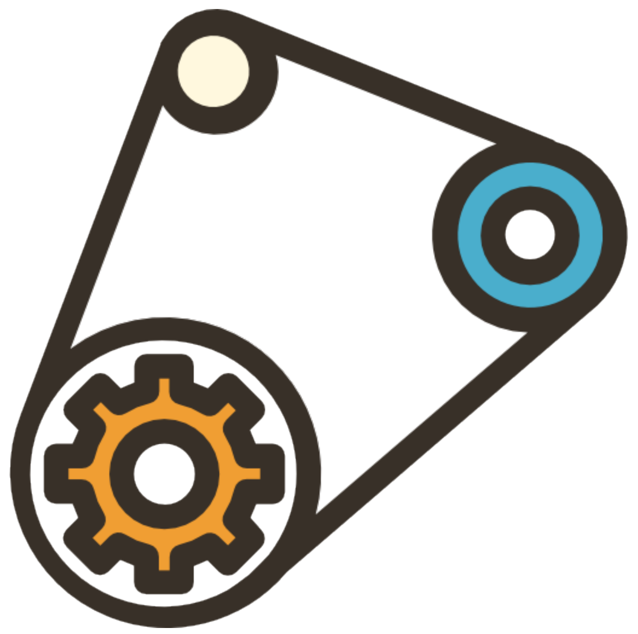
Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

→

IronMask



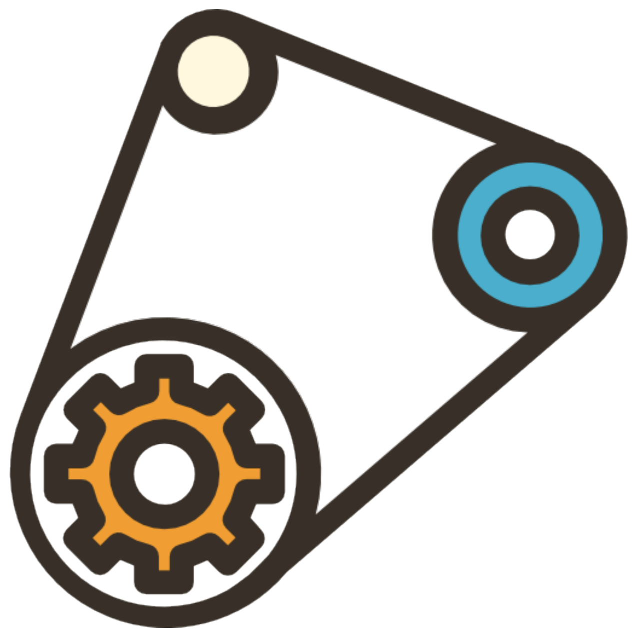
IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

IronMask



$ ./ironmask gadget.sage SNI -t 1

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

$ ./ironmask gadget.sage SNI -t 1

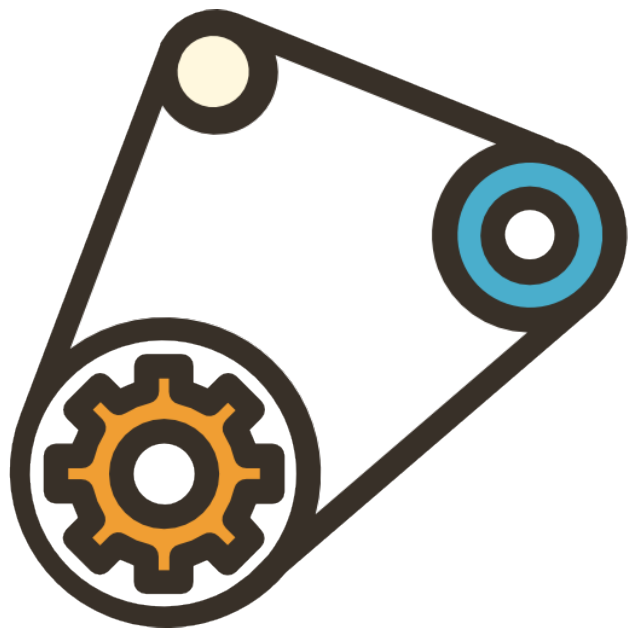
IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

IronMask



P

t-NI / t-SNI / …

```
$ ./ironmask gadget.sage SNI -t 1
```

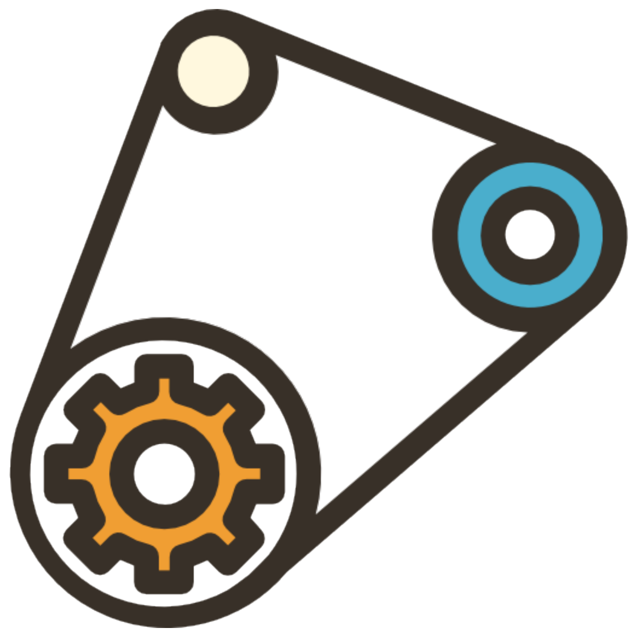
IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

→ IronMask



→ P

t-NI / t-SNI / …
or

$ ./ironmask gadget.sage SNI -t 1

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

t-NI / t-SNI / …
or
counter-example

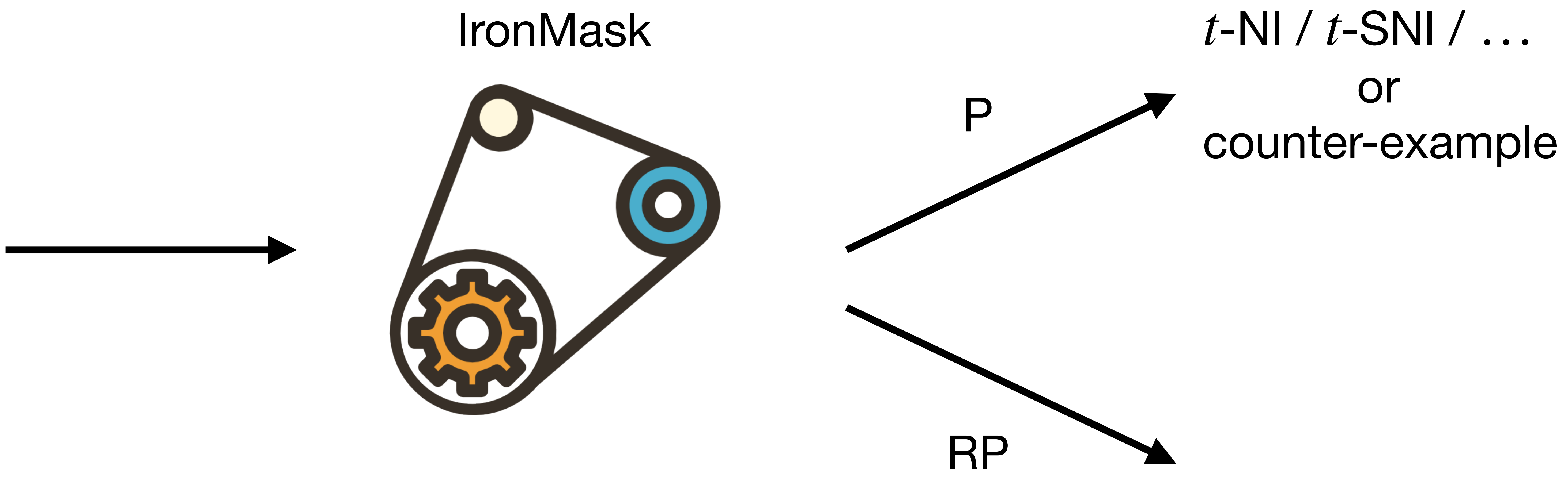
```
$ ./ironmask gadget.sage SNI -t 1
```

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file



$ ./ironmask gadget.sage SNI -t 1

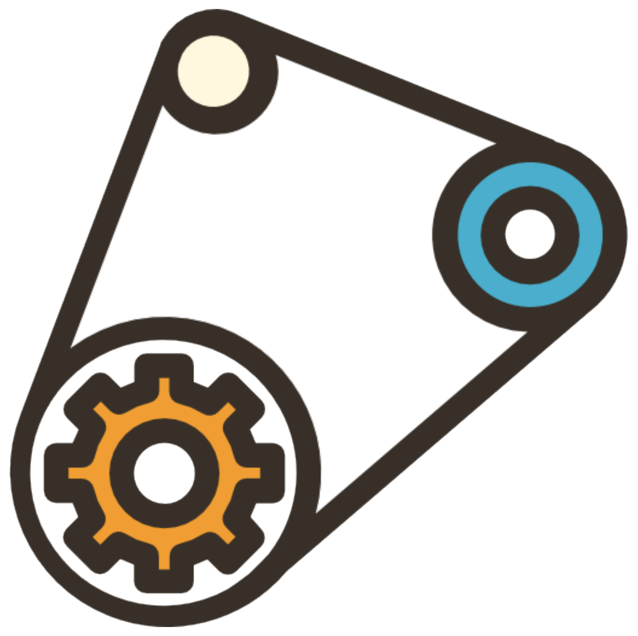
IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

→ IronMask



P → t-NI / t-SNI / … or counter-example

RP → $\varepsilon = f(p)$

$ ./ironmask gadget.sage SNI -t 1

IronMask

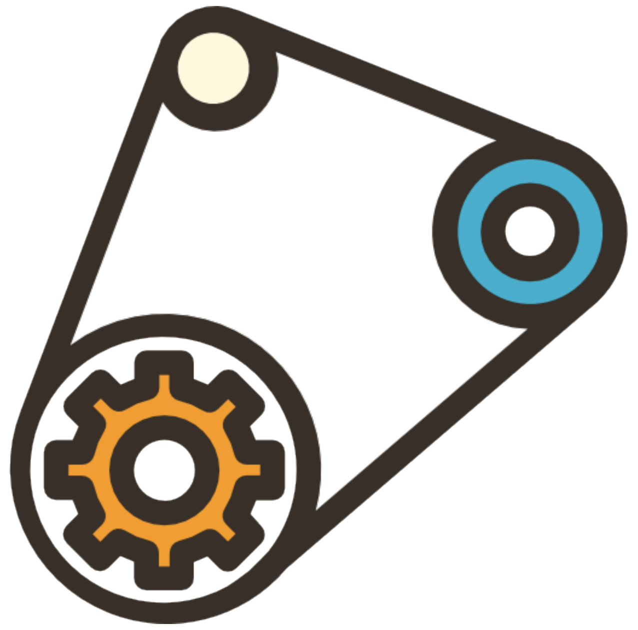
Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

→

IronMask



In C

P → t-NI / t-SNI / … or counter-example

RP → $\varepsilon = f(p)$

$ ./ironmask gadget.sage SNI -t 1

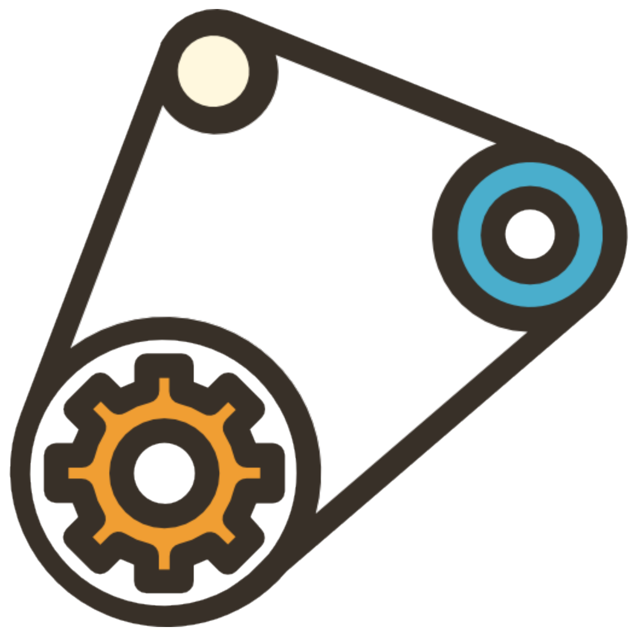
IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

IronMask



In C
+ multithreading

P → t-NI / t-SNI / …
or
counter-example

RP → $\varepsilon = f(p)$

$ ./ironmask gadget.sage SNI -t 1

IronMask

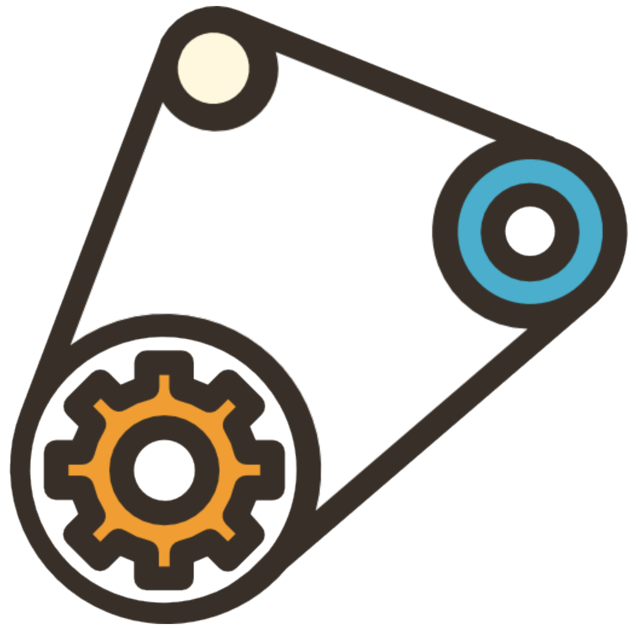
Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file

→

IronMask



In C
+ multithreading
+ several optimizations

P → t-NI / t-SNI / … or counter-example

RP → $\varepsilon = f(p)$

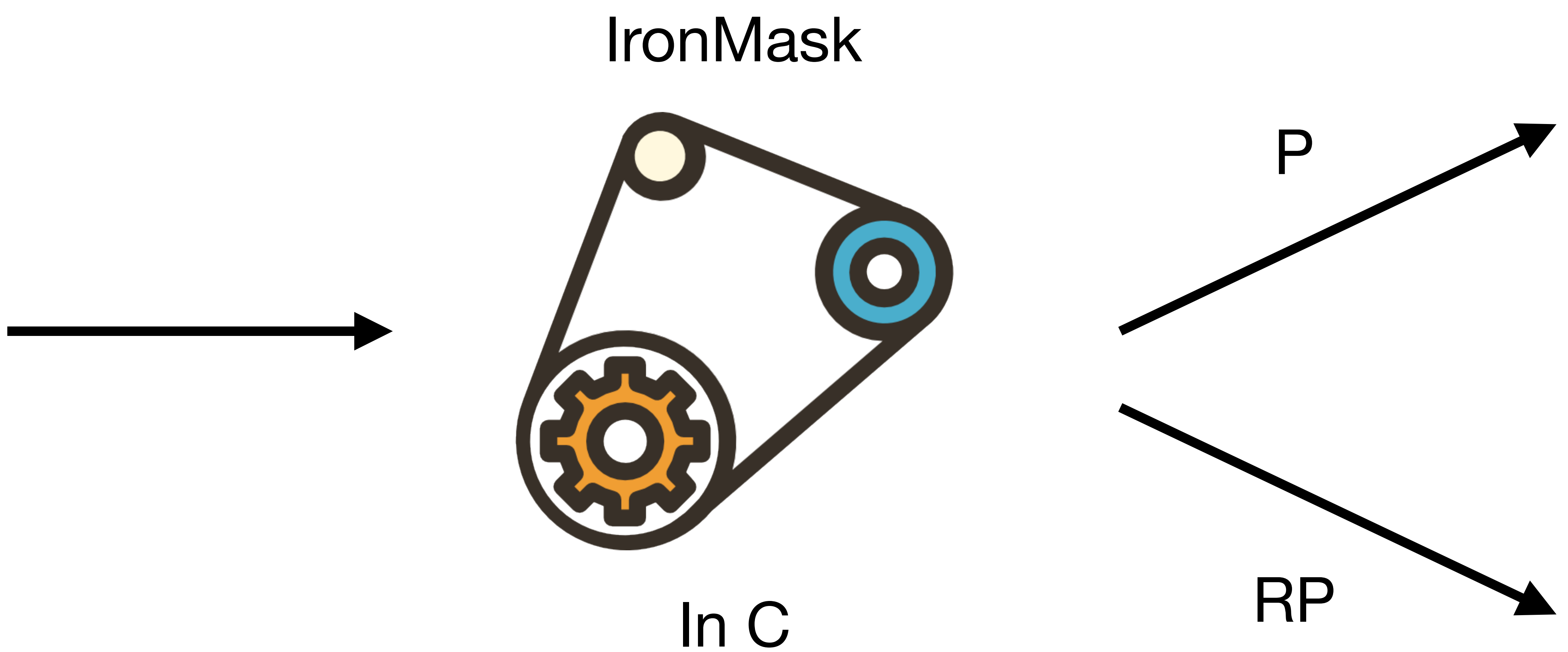
$ ./ironmask gadget.sage SNI -t 1

IronMask

Versatile Automatic Verification Tool *Belaïd, Mercadier, Rivain, Taleb [S&P'22]*

```
#shares 2
#in a b
#randoms r0
#out c

m0 = a0 * b1
t0 = ![ r0 + m0 ]
m1 = a1 * b0
t1 = ![ t0 + m1 ]

m2 = a0 * b0
c0 = m2 + r0

m3 = a1 * b1
c1 = m3 + t1
```

gadget file



In C
+ multithreading
+ several optimizations
+ compact data representation

t-NI / t-SNI / …
or
counter-example

$\varepsilon = f(p)$

$ ./ironmask gadget.sage SNI -t 1

IronMask

Performance

IronMask

Performance

Probing

IronMask

Performance

Probing

IronMask

Performance

Probing

Competitive with the fastest verification tools for probing-like properties (MaskVerif, MatVerif, …)

IronMask
Performance

Probing

Competitive with the fastest verification tools for probing-like properties (MaskVerif, MatVerif, …)

IronMask

Performance

Probing

Competitive with the fastest verification tools for probing-like properties (MaskVerif, MatVerif, …)

Random Probing

IronMask
Performance

Probing

Competitive with the fastest verification tools for probing-like properties (MaskVerif, MatVerif, …)

Random Probing

IronMask

Performance

Probing

Competitive with the fastest verification tools for probing-like properties (MaskVerif, MatVerif, …)

Random Probing

Gadget	Verification time	
	IronMask	VRAPS
5-share ISW mult.	3 sec	1h 15min
6-share ISW mult.	17 sec	> 24h
7-share ISW mult.	24 sec	> 24h

Conclusion

Conclusion

- The random probing model is gaining popularity

Conclusion

- The random probing model is gaining popularity

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*
- IronMask: https://github.com/CryptoExperts/IronMask

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*
- IronMask: https://github.com/CryptoExperts/IronMask
  - Formalization of all (random) probing properties in the state-of-the-art

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*
- IronMask: https://github.com/CryptoExperts/IronMask
  - Formalization of all (random) probing properties in the state-of-the-art
  - Exact proven verification methods for most gadgets with 2 inputs

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*

- IronMask: https://github.com/CryptoExperts/IronMask
    - Formalization of all (random) probing properties in the state-of-the-art
    - Exact proven verification methods for most gadgets with 2 inputs
    - Probing: similar performance as other fast verification tools

Conclusion

- The random probing model is gaining popularity
- The expansion strategy is a promising approach
- A recent work provides better composition, but without expansion *Cassiers, Faust, Orlt, and Standaert [CRYPTO'21]*
- **Dynamic** expansion strategy $\Longrightarrow$ better asymptotic complexities *Belaïd, Rivain, Taleb and Vergnaud [ASIACRYPT'21]*
- IronMask: https://github.com/CryptoExperts/IronMask
  - Formalization of all (random) probing properties in the state-of-the-art
  - Exact proven verification methods for most gadgets with 2 inputs
  - Probing: similar performance as other fast verification tools
  - Random probing: much faster than VRAPS

Thank you and see you next time !