Unifying Freedom and Separation for Tight Probing-Secure Composition

Sonia Belaïd[1], Gaëtan Cassiers[3], Matthieu Rivain[1], **Abdel Rahman Taleb**[1,2]

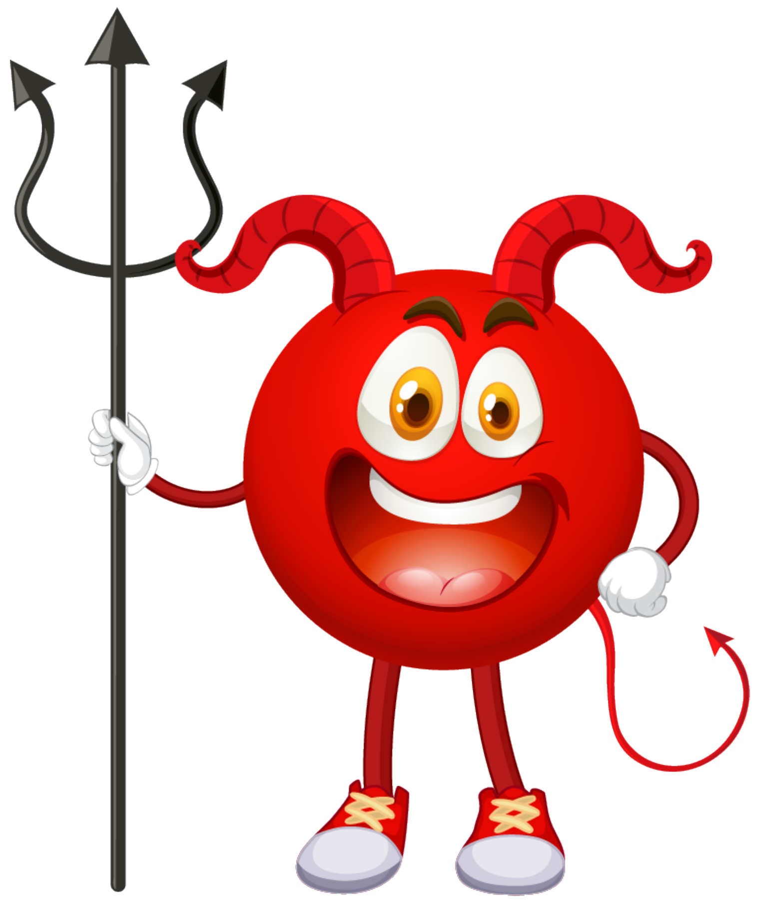
[1] CryptoExperts, France

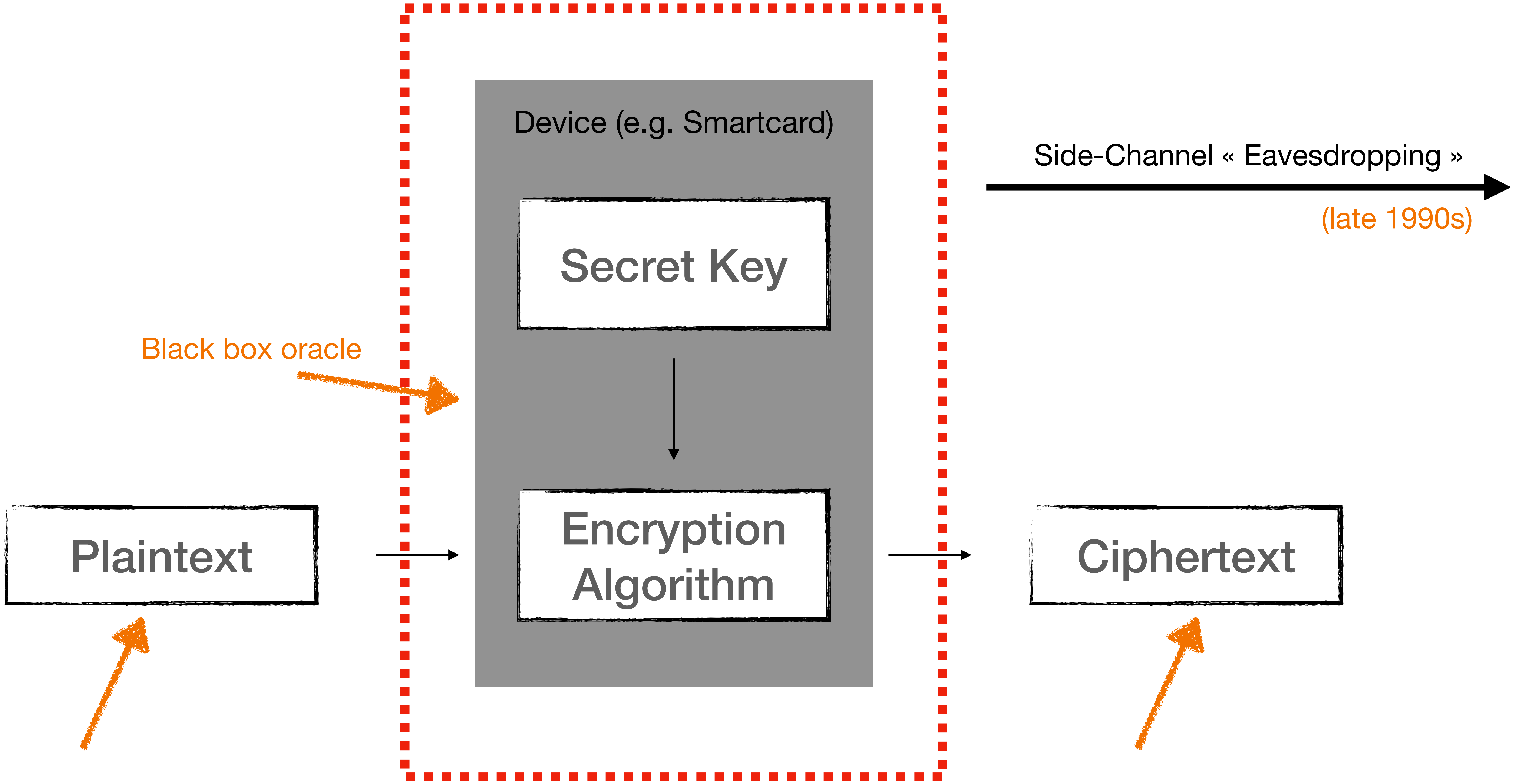
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

[3] TU Graz, Austria

CRYPTO - 21/08/2023

Side-Channel Attacks





Execution Time

Power Consumtion

Electromagnetic Radiation

Memory Cache

…

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

each observation comes with noise
Number of observations grows $\Longrightarrow$ harder to retrieve the secret

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

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

Encode

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

s.t.

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

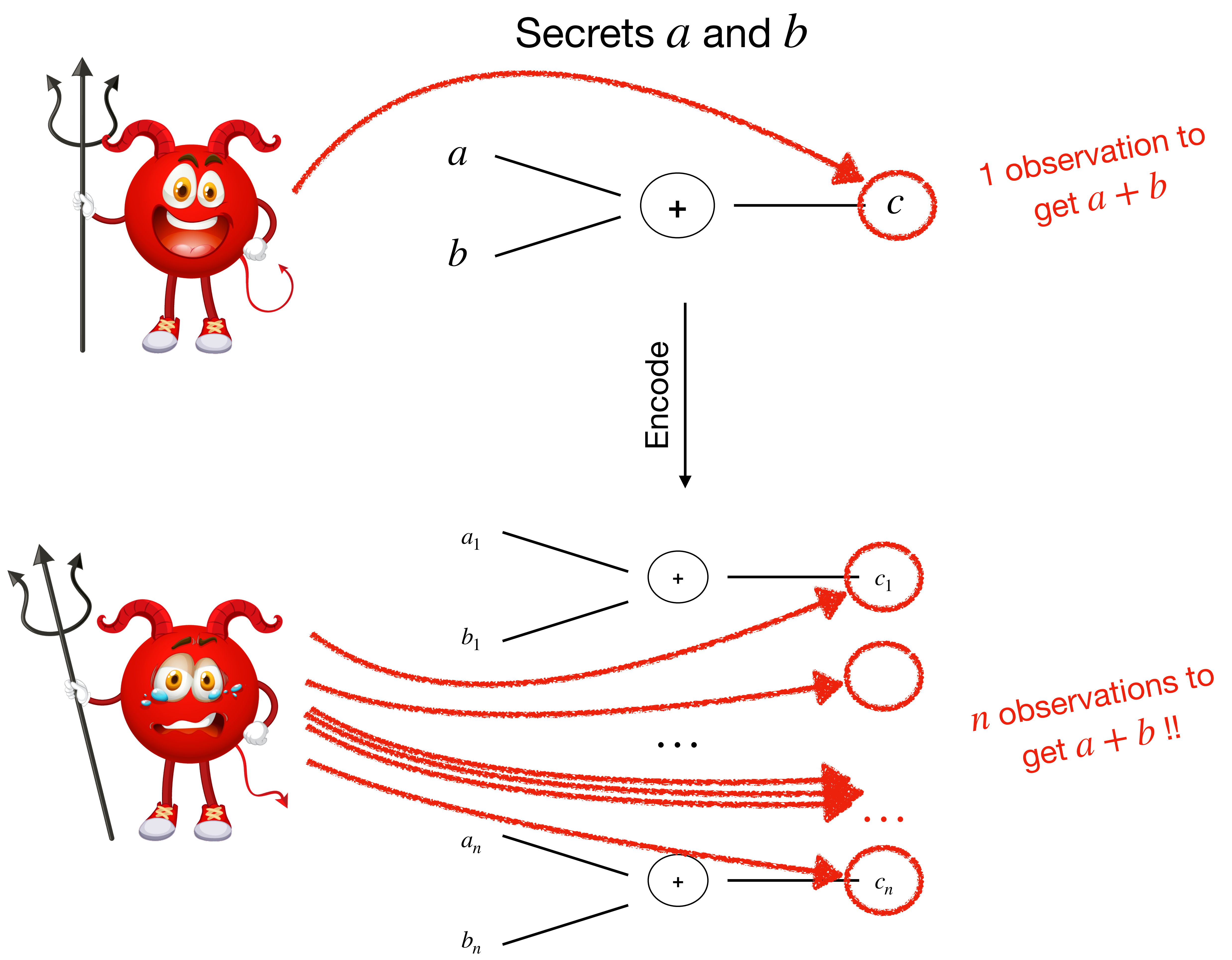
$\ldots$

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

$n-1$ random values

secret recombination

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



Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

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

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

random

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

Countermeasure

Gadgets

Operations over variables $\mathbb{F}_2$

Atomic gates

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

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

random

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

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

n-share Gadgets formed of atomic gates

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

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

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 → $\times$ → $a \times b$

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. $c_1 + \ldots + c_n = a \times b$

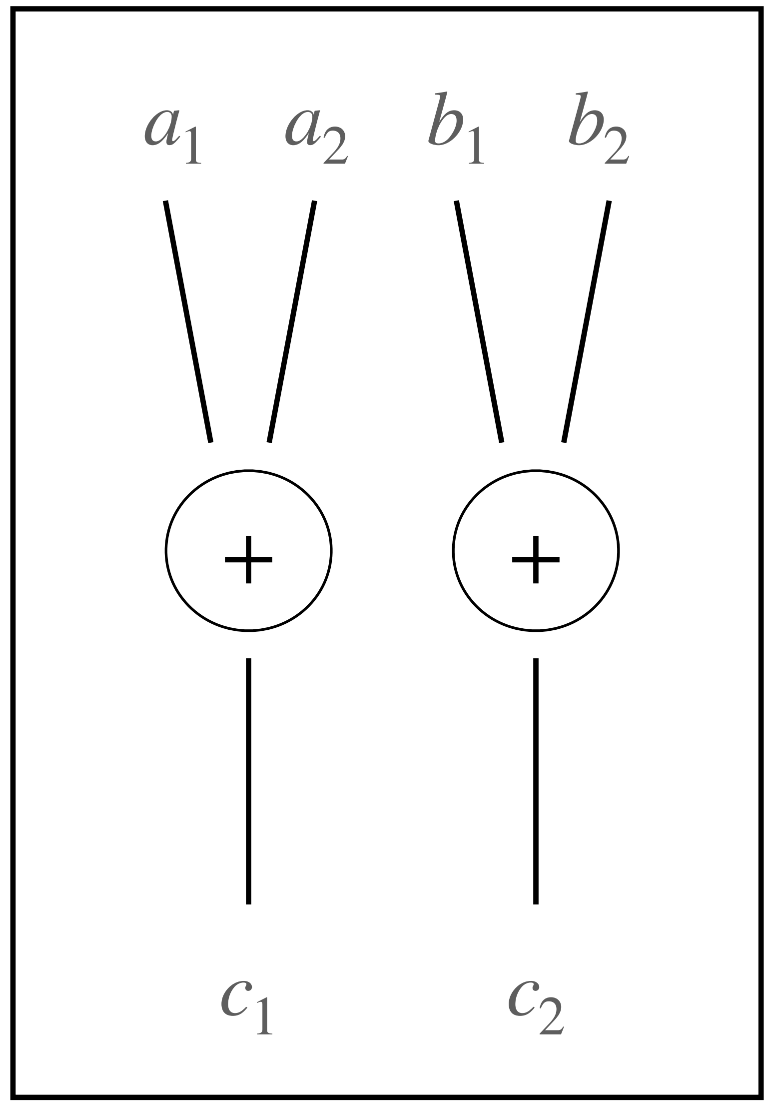
$(a_1, \ldots, a_n)$ → $G_{refresh}$ → new fresh shares $(c_1, \ldots, c_n)$ s.t. $c_1 + \ldots + c_n = a$

Probing Model

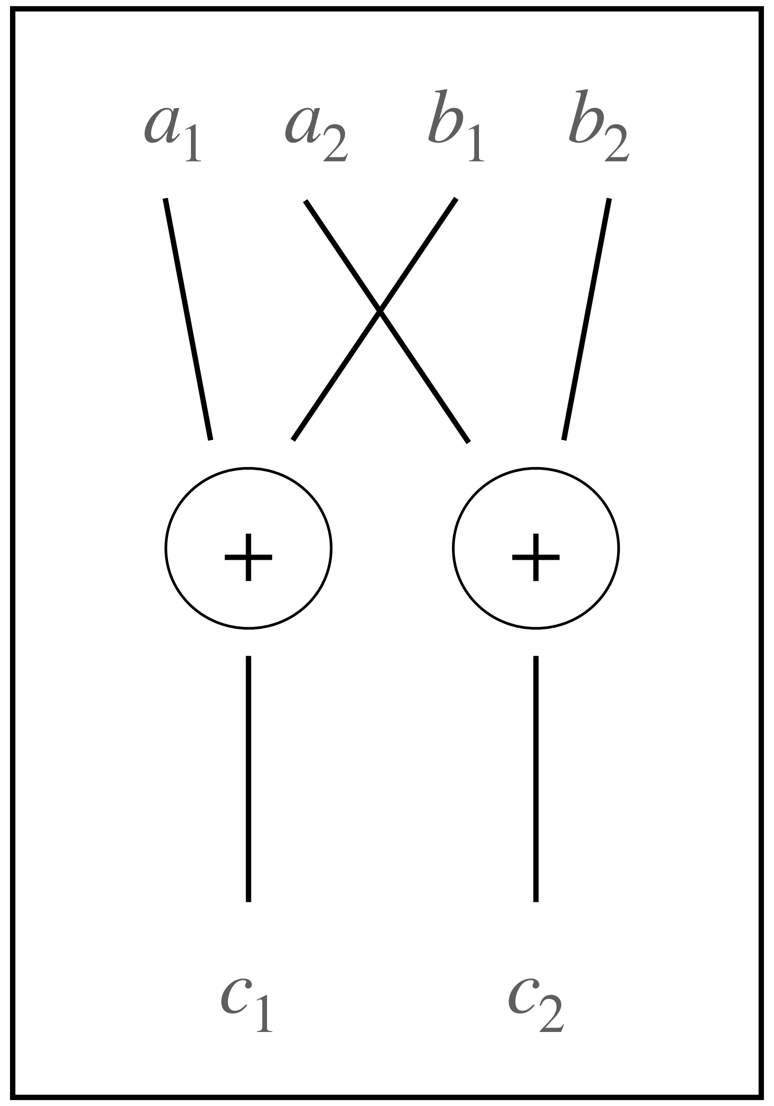
Security *Ishai, Sahai and Wagner [CRYPTO'03]*

t-probing security ($t < n$): any set of at most t variables is independent of the secrets

G_+^1



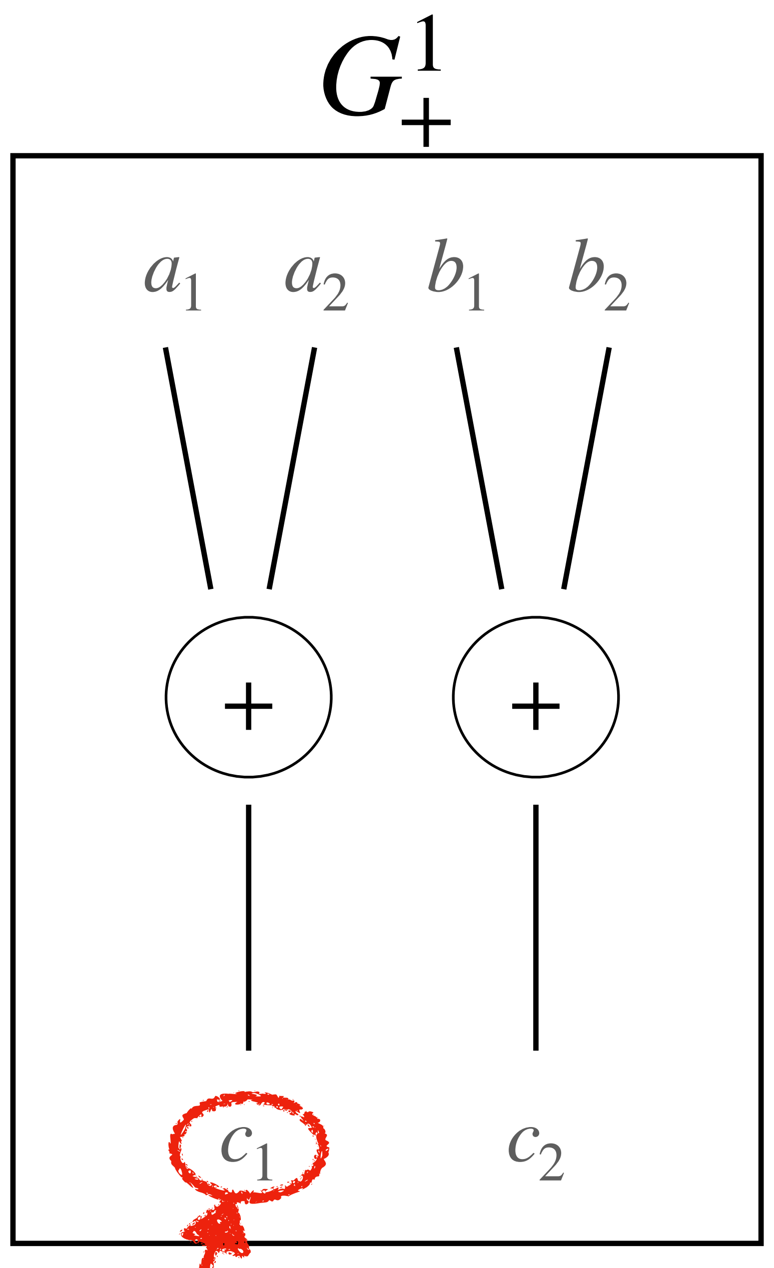
G_+^2

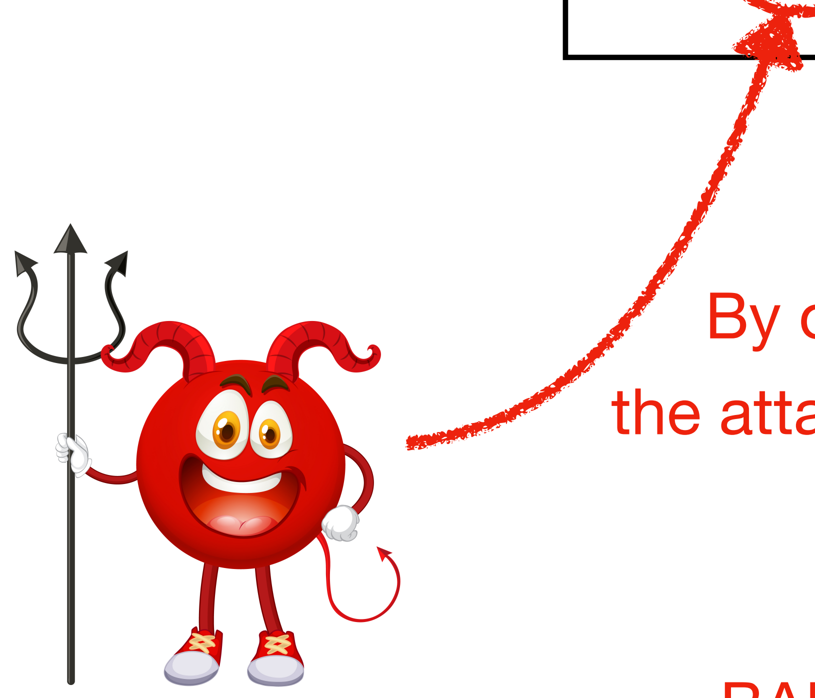


Probing Model

Security *Ishai, Sahai and Wagner [CRYPTO'03]*

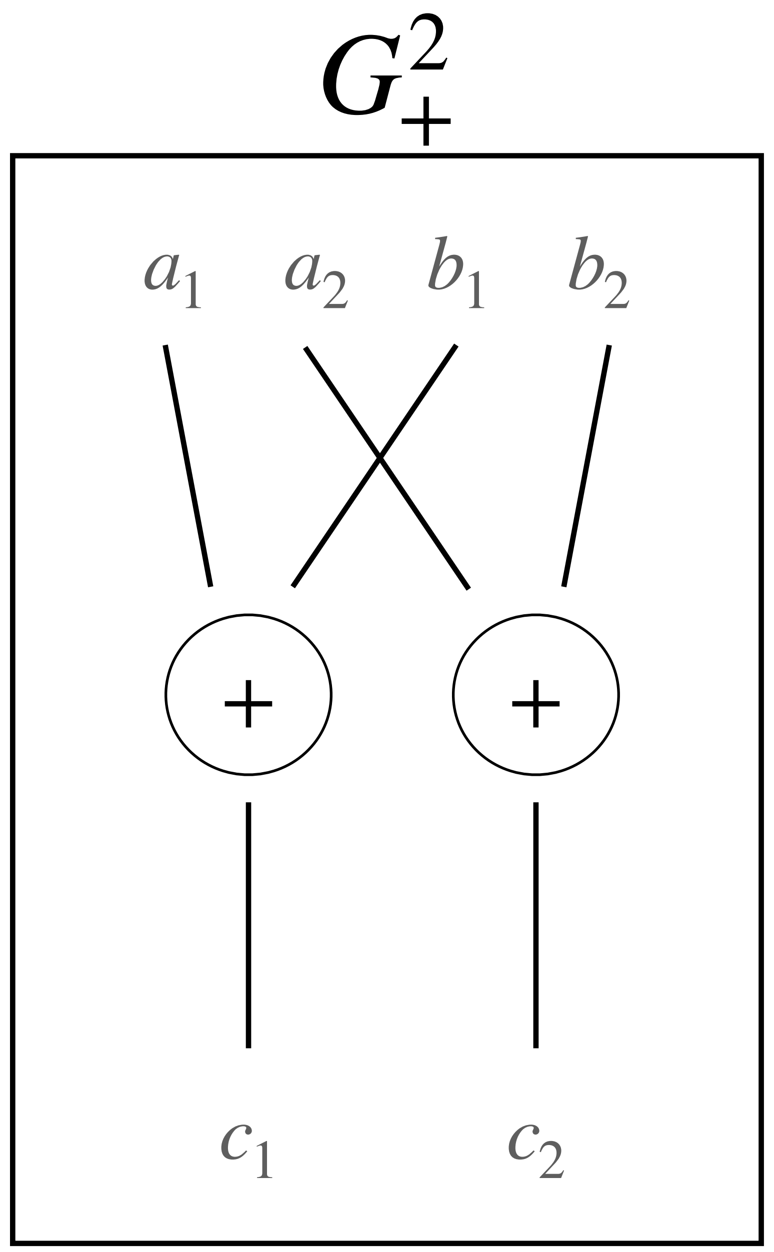
t-probing security ($t < n$): any set of at most t variables is independent of the secrets

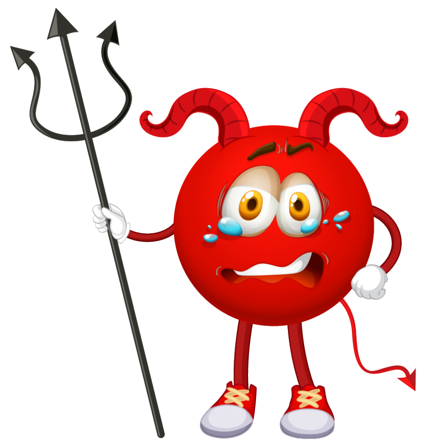




By observing c_1, the attacker retrieves a

BAD EXAMPLE



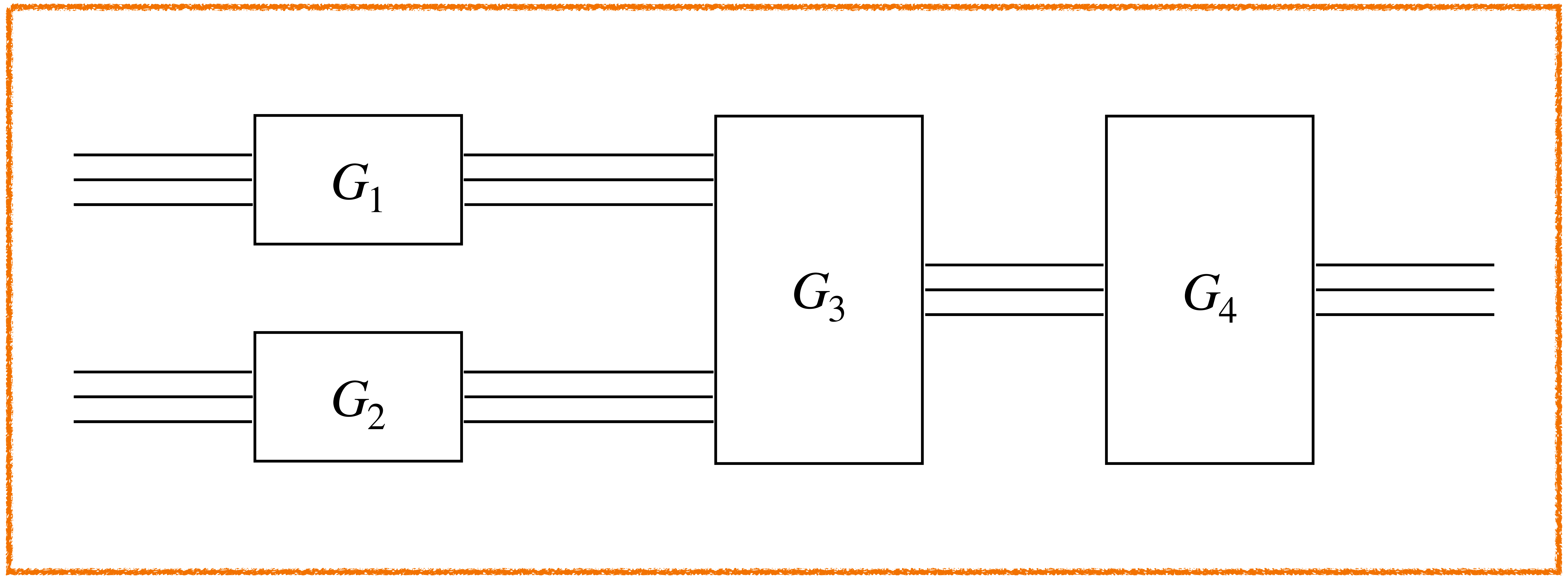


No single observation can retrieve a or b

GOOD EXAMPLE

Probing Model

Composition

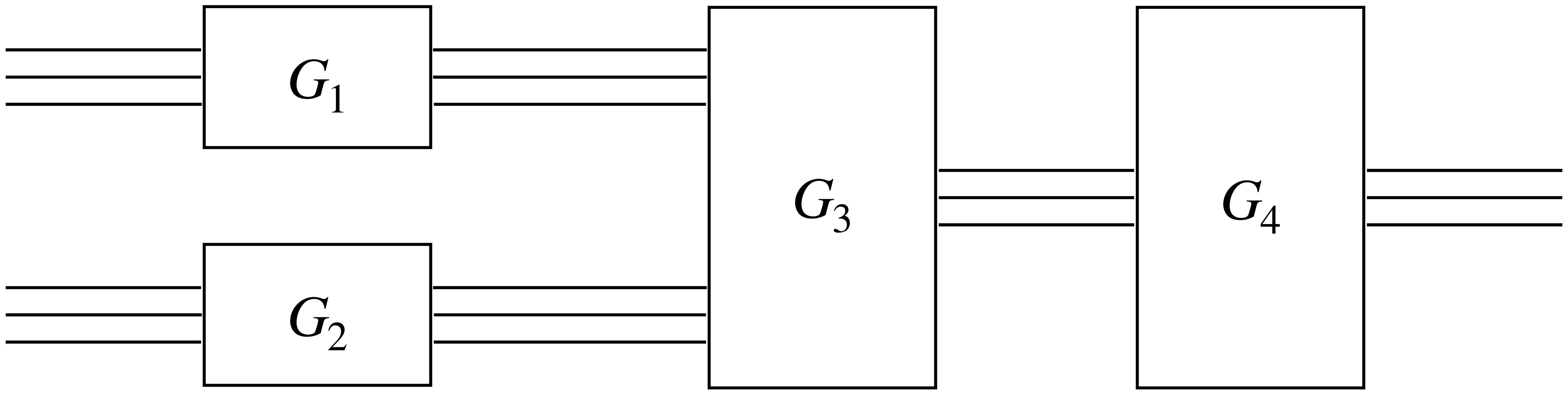


2-probing secure?
($n = 3$ shares)

Probing Model

Composition: Non-interference (NI) *Barthe et al. [CCS'16]*

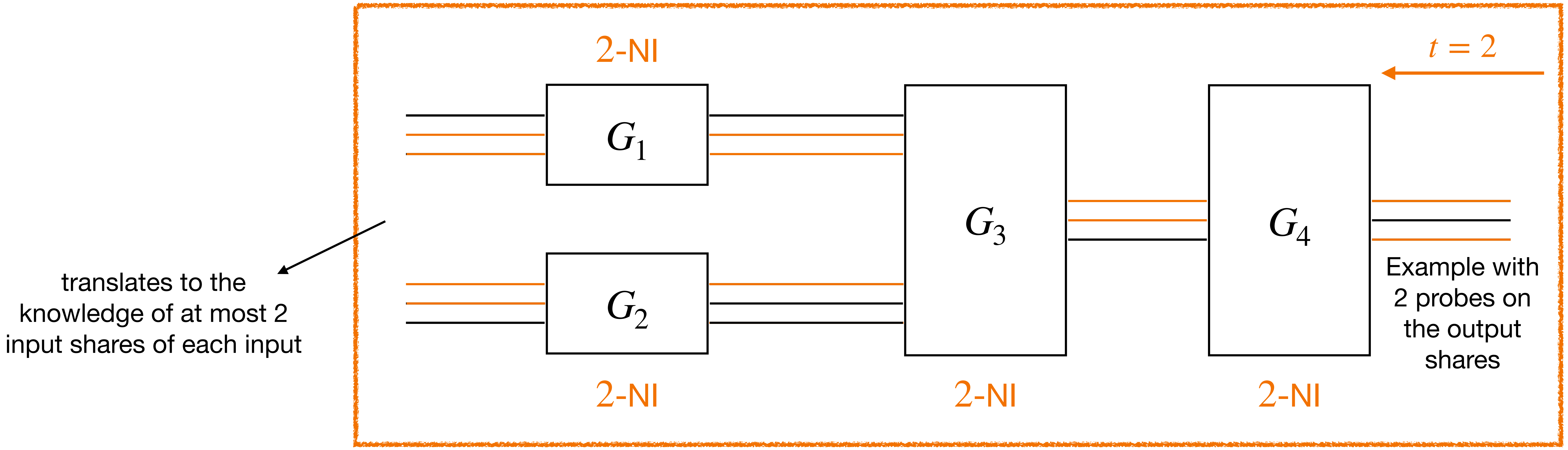
t-**NI:** the distribution of any set of at most t variables can be simulated with the knowledge of at most t input shares of each input



Probing Model

Composition: Non-interference (NI) *Barthe et al. [CCS'16]*

t**-NI:** the distribution of any set of at most t variables can be simulated with the knowledge of at most t input shares of each input

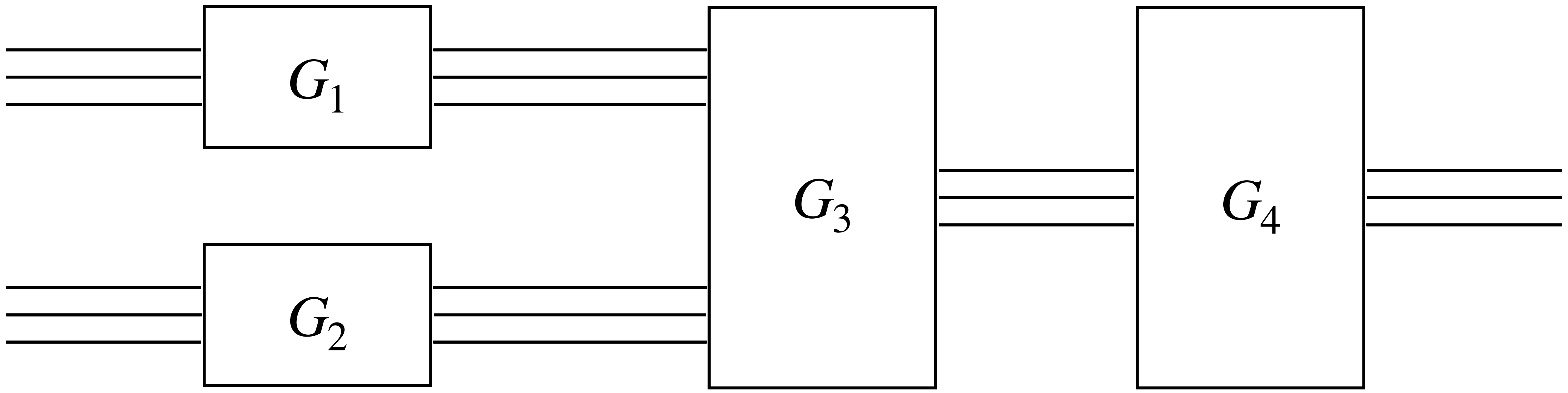


$\Longrightarrow$ 2-probing secure

Probing Model

Composition: Strong Non-interference (SNI) *Barthe et al. [CCS'16]*

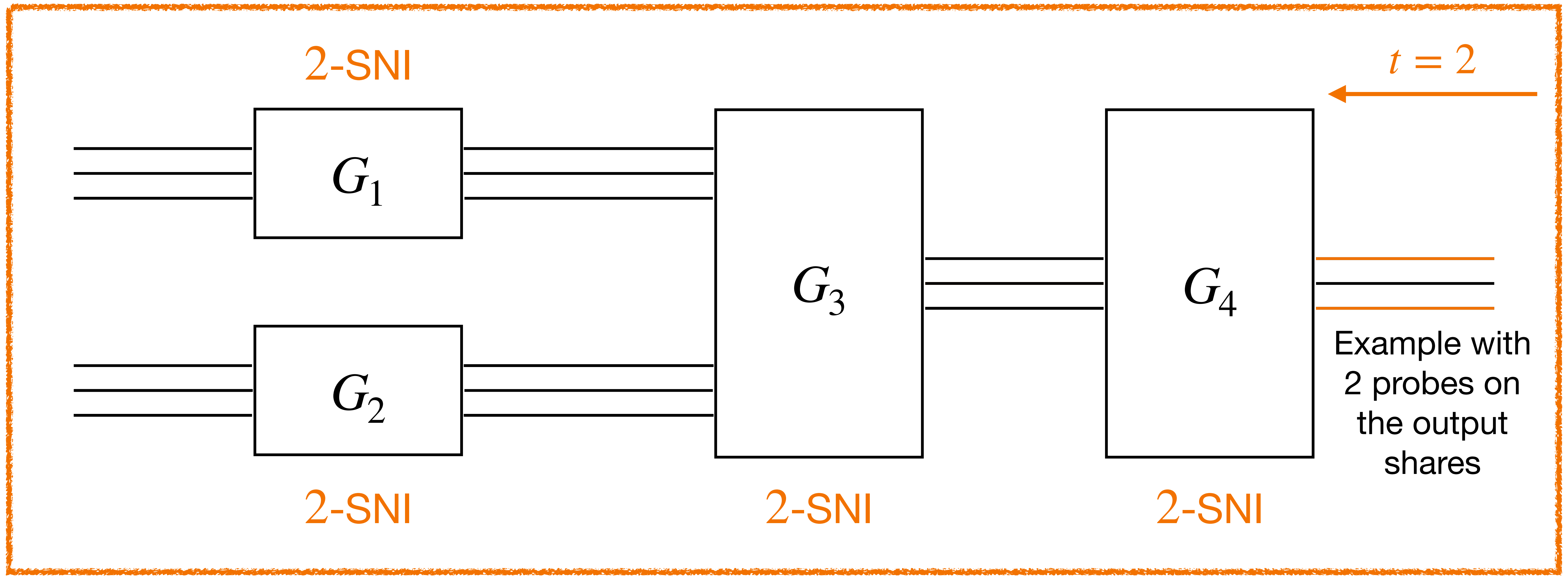
t-SNI: the distribution of any set of at most t_1 intermediate variables and t_2 output variables such that $t_1 + t_2 \leq t$, can be simulated with the knowledge of at most t_1 input shares of each input



Probing Model

Composition: Strong Non-interference (SNI) *Barthe et al. [CCS'16]*

***t*-SNI:** the distribution of any set of at most t_1 intermediate variables and t_2 output variables such that $t_1 + t_2 \leq t$, can be simulated with the knowledge of at most t_1 input shares of each input

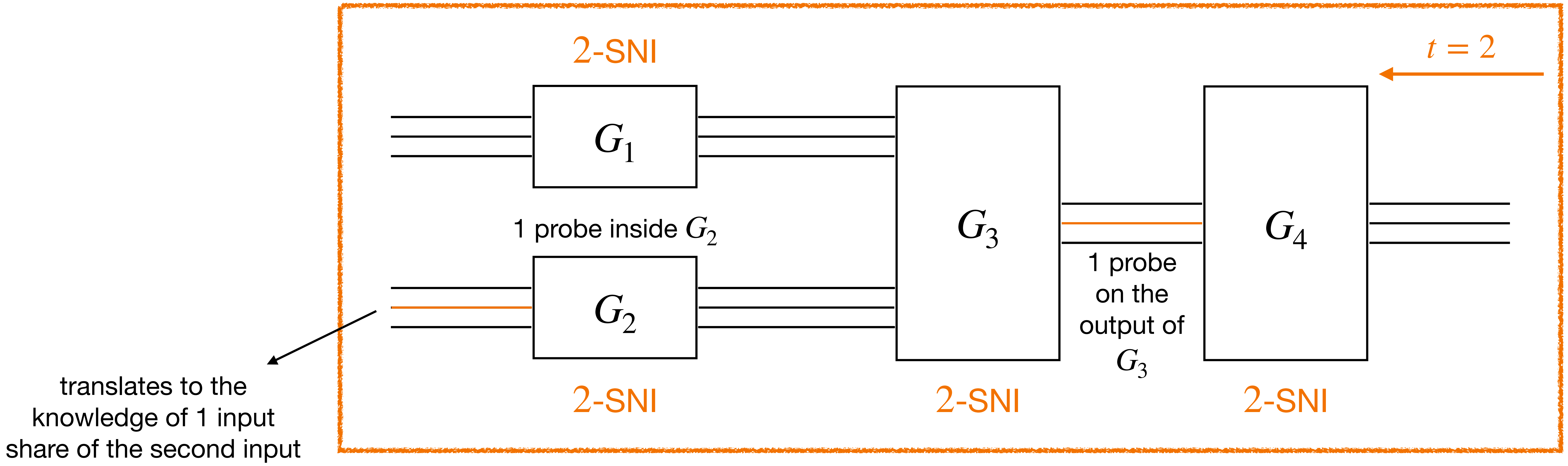


$\Longrightarrow$ 2-probing secure

Probing Model

Composition: Strong Non-interference (SNI) *Barthe et al. [CCS'16]*

t-SNI: the distribution of any set of at most t_1 intermediate variables and t_2 output variables such that $t_1 + t_2 \leq t$, can be simulated with the knowledge of at most t_1 input shares of each input



$\Longrightarrow$ 2-probing secure

Probing Model

Stronger Region Probing Security

Split the circuit into regions

Each region is t-probing secure $\Longrightarrow$ whole circuit is t-region probing secure

Better reduction to more realistic leakage models

Motivation of this Work

Tight Private Circuits *Belaïd, Goudarzi and Rivain [ASIACRYPT'18]*

Secure composition in the probing model by inserting refresh gadgets

Only inserts refresh gadgets when needed (tight composition)

Uses SNI multiplication and refresh gadgets (authors use ISW scheme)

Motivation of this Work

Tight Private Circuits *Belaïd, Goudarzi and Rivain [ASIACRYPT'18]*

Secure composition in the probing model by inserting refresh gadgets

Only inserts refresh gadgets when needed (tight composition)

Uses SNI multiplication and refresh gadgets (authors use ISW scheme)

↓

Not sufficient!

(more details later)

Contributions

free t-SNI

Coron and Spignoli [CRYPTO'21]

secure wire shuffling in the probing model

t-IOS (Input Output Separation)

Goudarzi et al. [TCHES'21]

composition in the region probing model

Contributions

free t-SNI

Coron and Spignoli [CRYPTO'21]

secure wire shuffling in the probing model

t-IOS (Input Output Separation)

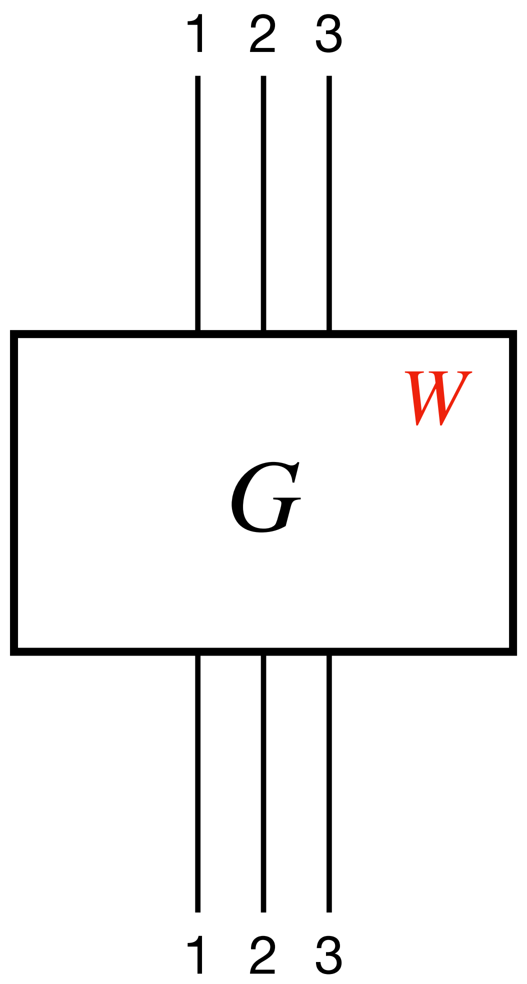
Goudarzi et al. [TCHES'21]

composition in the region probing model

- Unify and extend free t-SNI and t-IOS
- Propose efficient automatic verification for both properties and include it in IronMask ***(Belaïd et al. [S&P'22])***
- Propose gadgets that satisfy both notions
- Generalize Tight Private Circuits (TPC) and show that it requires free t-SNI multiplication and refresh gadgets
- Provide more efficient composition in the region probing model

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

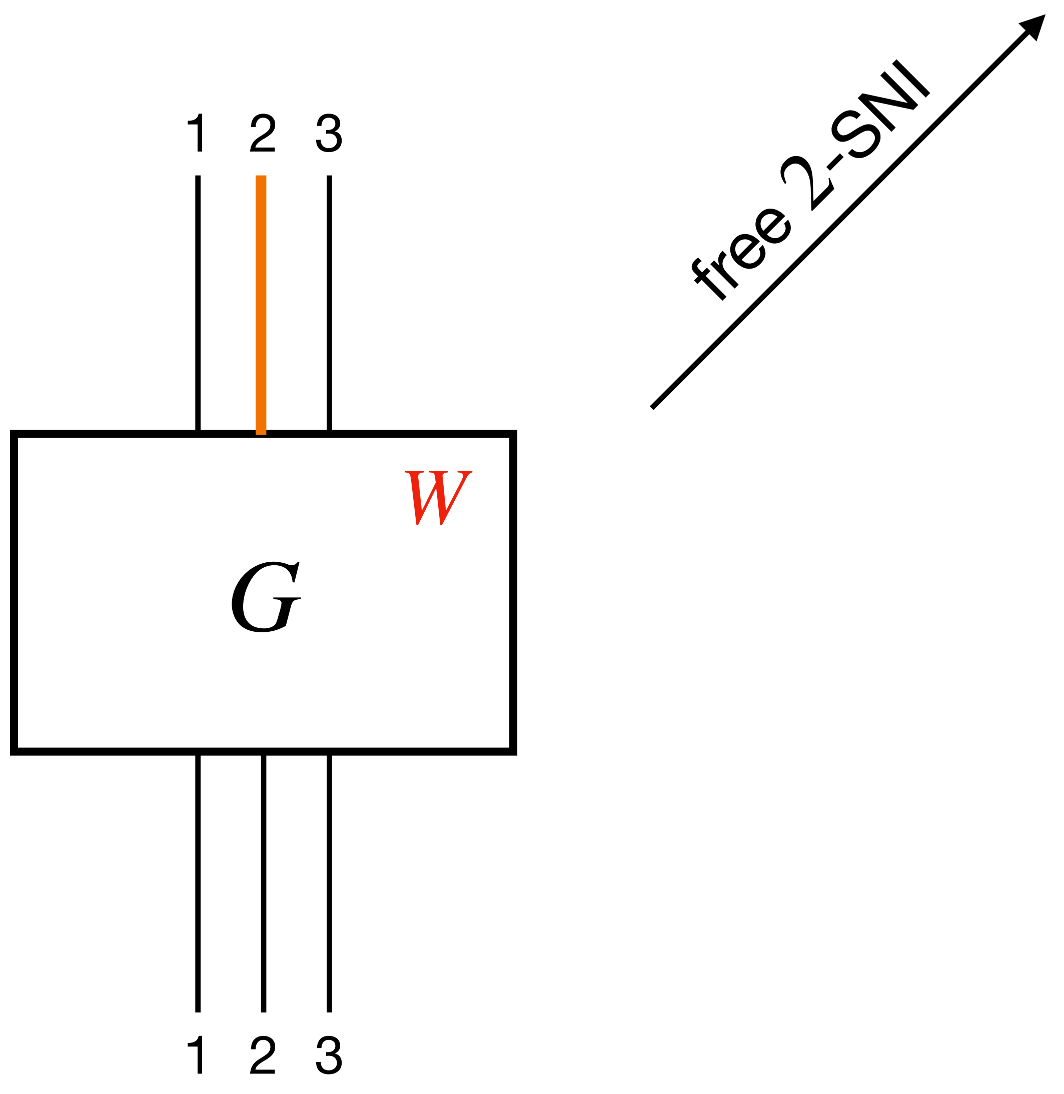
$|W| \leq 2$

Stronger Composition Notions

Free-SNI & IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

example: $I = \{2\}$



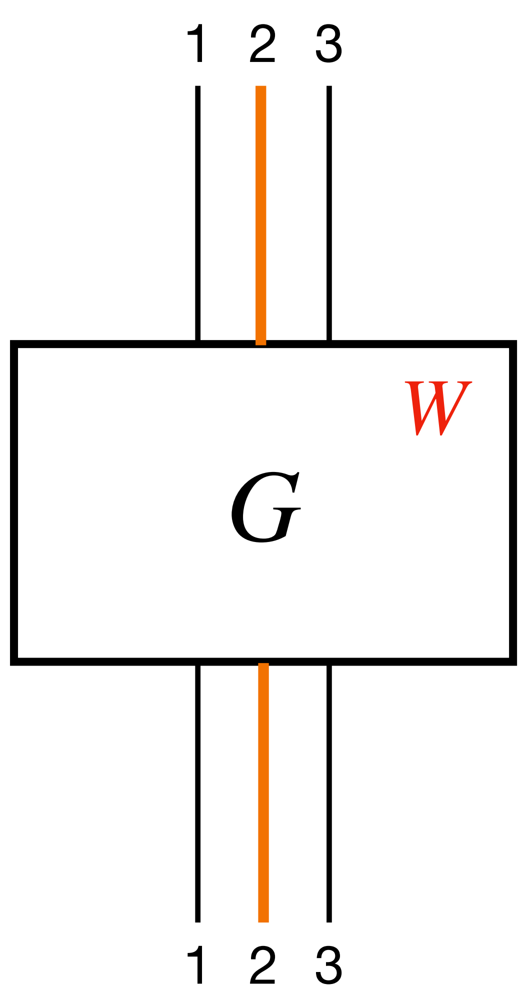
3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI → $\exists\ I$, set of input shares s.t. $|I| \leq |W|$

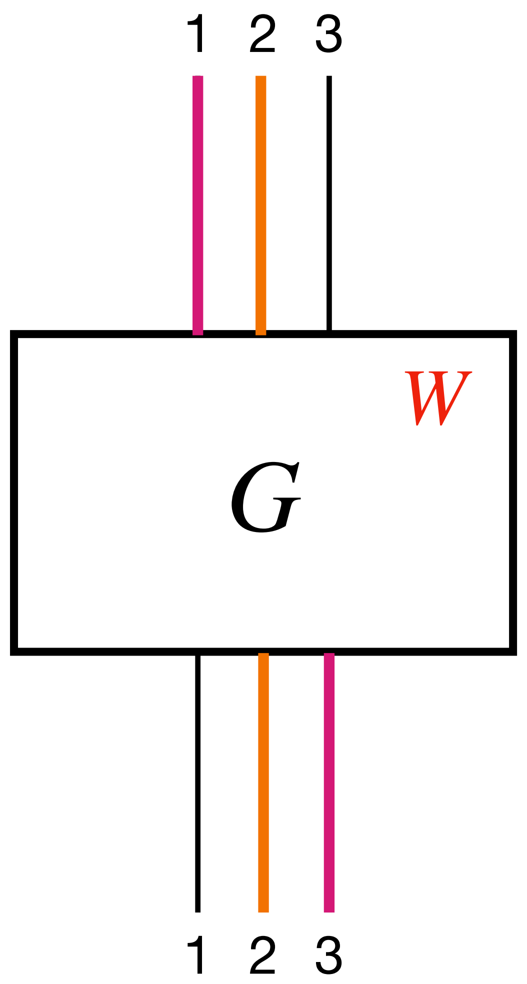
example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1{,}3\} \backslash J$ are mutually independent from the simulation and uniform

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI →

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1{,}3\} \backslash J$ are mutually independent from the simulation and uniform

2-IOS →

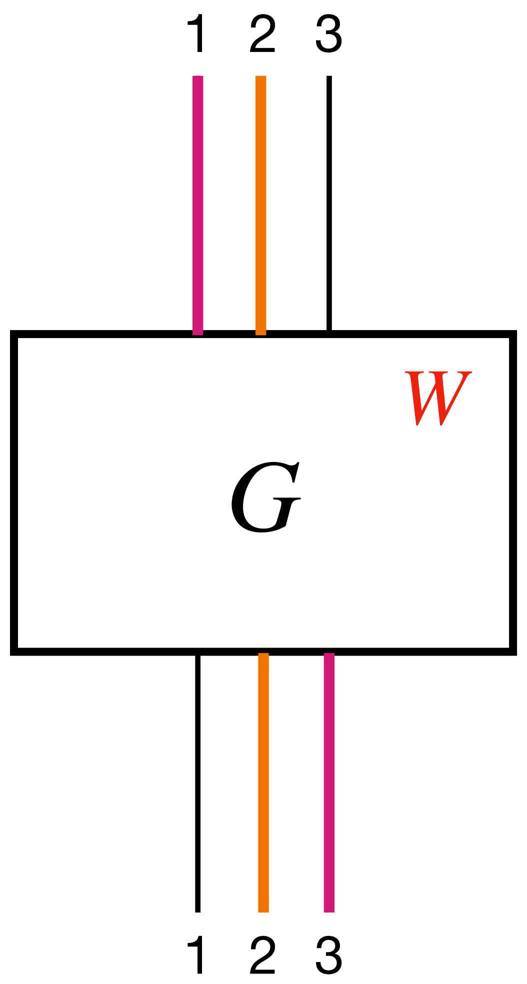
$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

example: $I = \{1\}, J = \{3\}$

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI:

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1,3\} \backslash J$ are mutually independent from the simulation and uniform

2-IOS:

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

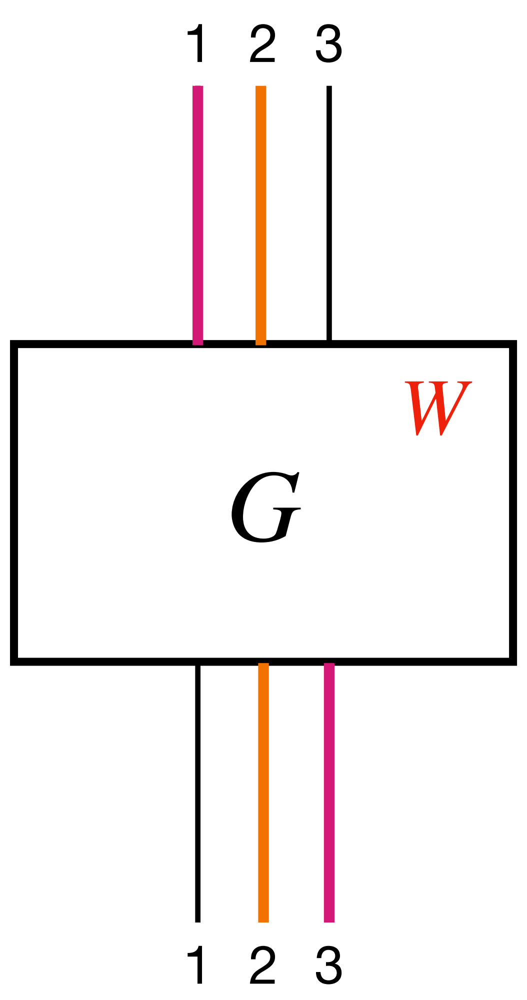
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

example: $I = \{1\}, J = \{3\}$

→ perfect simulation of W using input shares in I and output shares in J

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI:

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1{,}3\} \backslash J$ are mutually independent from the simulation and uniform

implies uniformity of the output sharing (when $W = \emptyset$)

2-IOS:

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

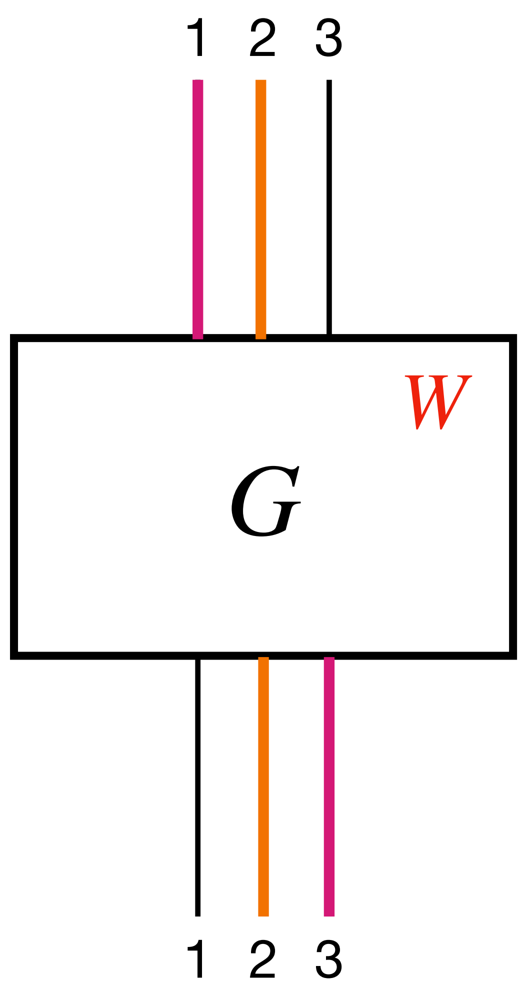
example: $I = \{1\}, J = \{3\}$

→ perfect simulation of W using input shares in I and output shares in J

requires uniformity of the output sharing

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI →

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1{,}3\} \backslash J$ are mutually independent from the simulation and uniform

implies uniformity of the output sharing (when $W = \emptyset$)

input sharing is fixed for the simulation

2-IOS →

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$
example: $I = \{1\}, J = \{3\}$

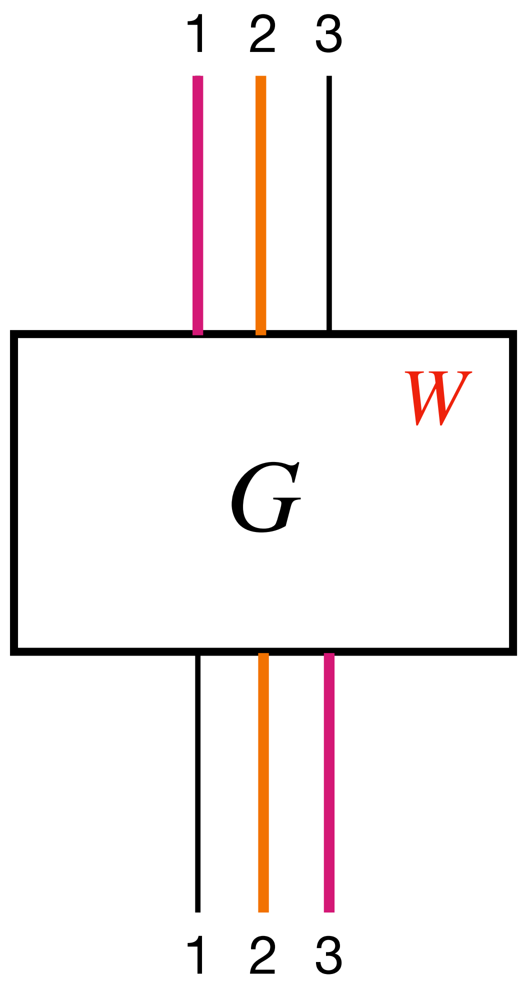
→ perfect simulation of W using input shares in I and output shares in J

requires uniformity of the output sharing

input and output sharings are fixed for the simulation

Stronger Composition Notions

Free-SNI & IOS



3-share 1-input 1-output gadget

W: set of probes on G

$|W| \leq 2$

free 2-SNI

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

example: $I = \{2\}$

→ perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1,3\} \backslash J$ are mutually independent from the simulation and uniform

implies uniformity of the output sharing (when $W = \emptyset$)

input sharing is fixed for the simulation

We generalize both to 2-input gadgets

2-IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

example: $I = \{1\}, J = \{3\}$

→ perfect simulation of W using input shares in I and output shares in J

requires uniformity of the output sharing

input and output sharings are fixed for the simulation

Stronger Composition Notions

Free-SNI & IOS

$W, |W| \leq t$

free t-SNI

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1,\ldots,n\} \backslash J$ are mutually independent from the simulation and uniform

t-IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

perfect simulation of W using input shares in I and output shares in J

Stronger Composition Notions

Free-SNI & IOS

$W, |W| \leq t$

free t-SNI

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1,\ldots,n\}\backslash J$ are mutually independent from the simulation and uniform

$\Longleftrightarrow$

balanced t-IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$J = I$ set of output shares

perfect simulation of W using input shares in I and output shares in J

Stronger Composition Notions

Free-SNI & IOS

$W, |W| \leq t$

free t-SNI

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$

perfect simulation of W and output shares in $J = I$, using input shares in I

output shares in **any strict subset of** $\{1, \ldots, n\} \backslash J$ are mutually independent from the simulation and uniform

t-IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

perfect simulation of W using input shares in I and output shares in J

Stronger Composition Notions

Free-SNI & IOS

$W, |W| \leq t$

Unbalanced free t-SNI

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

perfect simulation of W and output shares in J, using input shares in I

output shares in **any strict subset of** $\{1,\ldots,n\}\backslash J$ are mutually independent from the simulation and uniform

$\Longleftrightarrow$

t-IOS

$\exists\ I$, set of input shares s.t. $|I| \leq |W|$
$\exists\ J$, set of output shares s.t. $|J| \leq |W|$

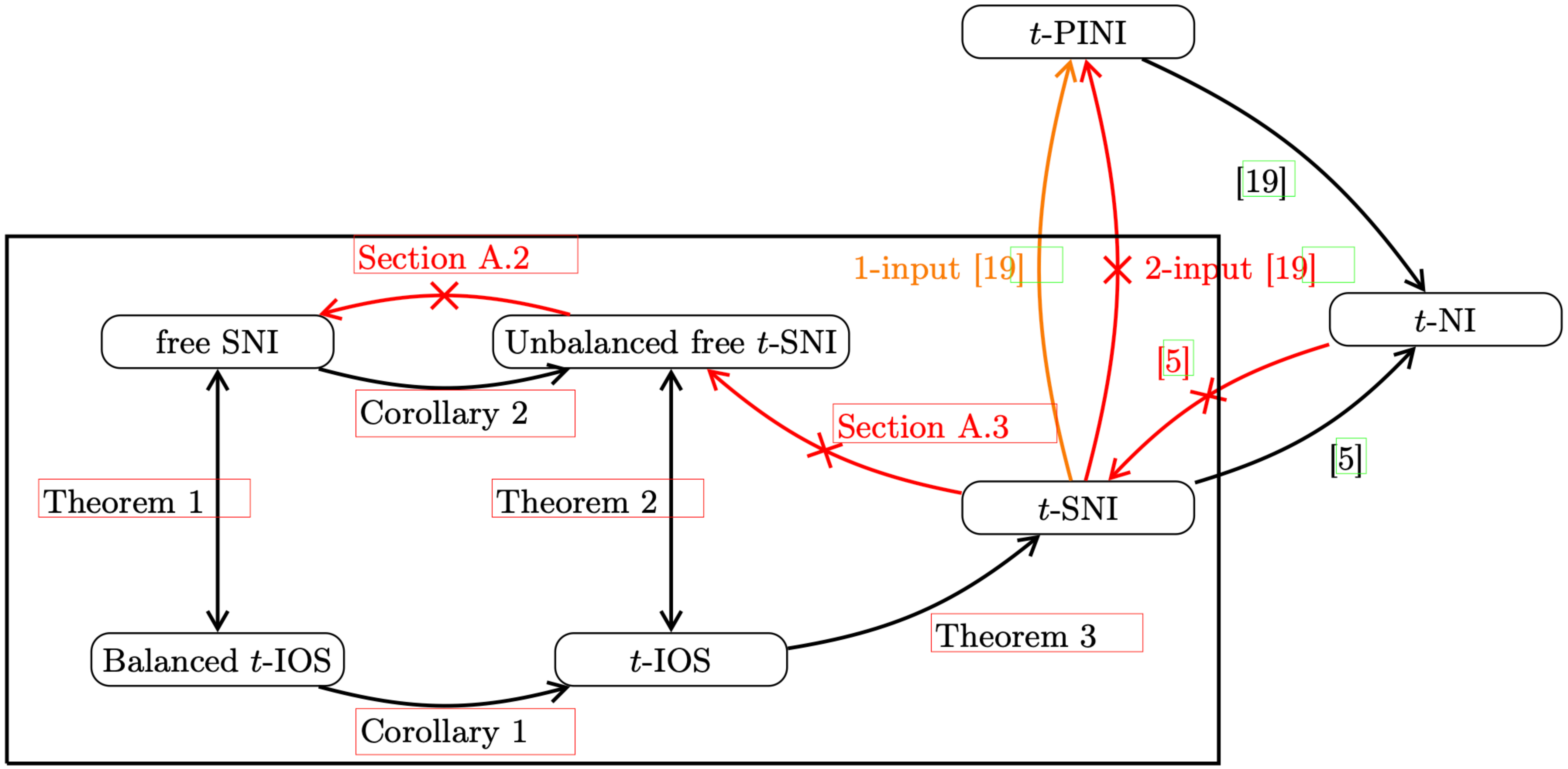
perfect simulation of W using input shares in I and output shares in J

Stronger Composition Notions

Free-SNI & IOS

Stronger Composition Notions

Free-SNI & IOS



Automatic Verification

IronMask *Belaïd et al. [S&P'22]*

- Verification tool for probing and random probing properties
- Algebraic characterization for probe expression

Automatic Verification

IronMask *Belaïd et al. [S&P'22]*

- Verification tool for probing and random probing properties
- Algebraic characterization for probe expression

Set of Probes

Automatic Verification

IronMask *Belaïd et al. [S&P'22]*

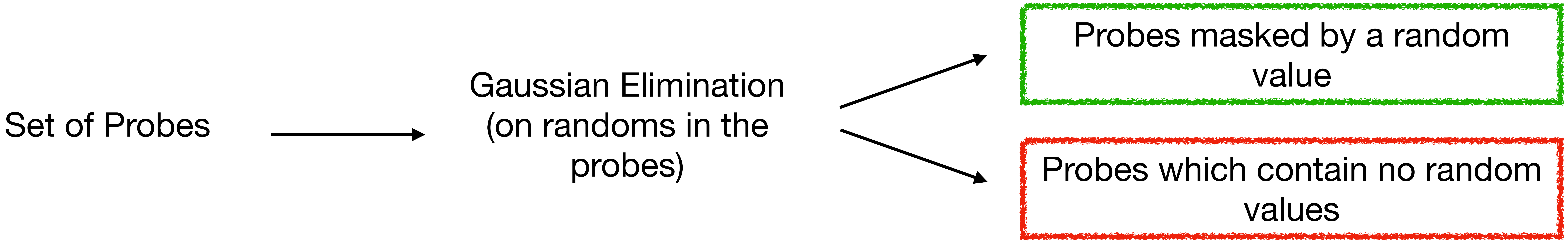
- Verification tool for probing and random probing properties
- Algebraic characterization for probe expression

Set of Probes → Gaussian Elimination (on randoms in the probes)

Automatic Verification

IronMask *Belaïd et al. [S&P'22]*

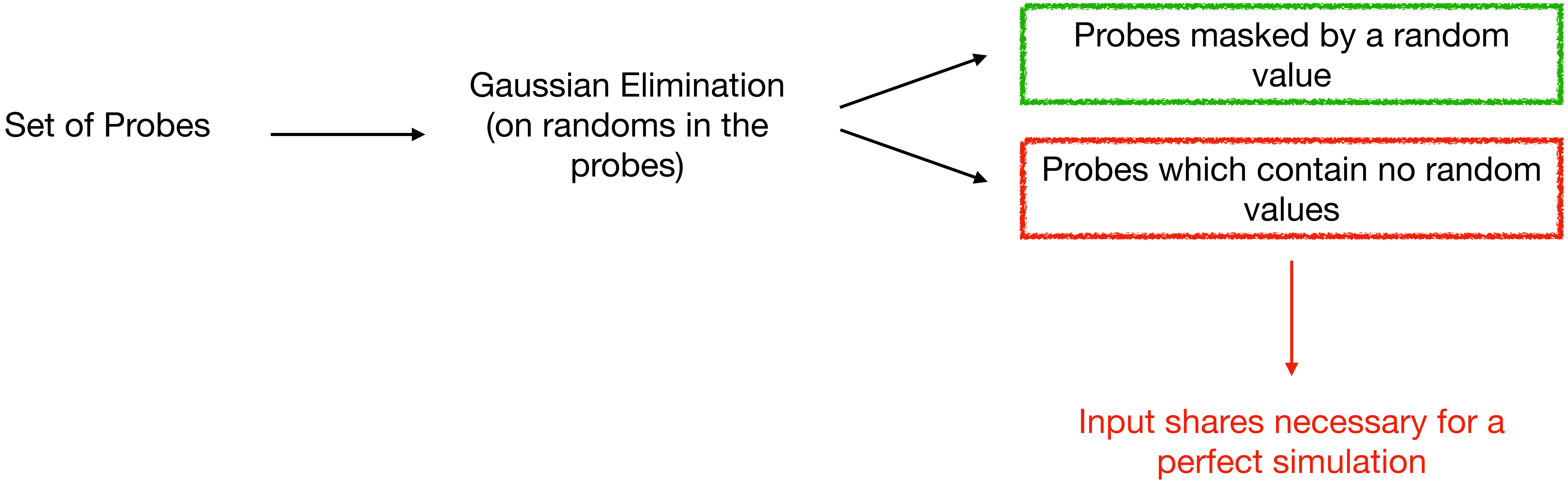
- Verification tool for probing and random probing properties
- Algebraic characterization for probe expression



Automatic Verification

IronMask *Belaïd et al. [S&P'22]*

- Verification tool for probing and random probing properties
- Algebraic characterization for probe expression



Automatic Verification

Free-SNI & IOS

Verification of Free-SNI and IOS (or balanced Free-SNI)

Automatic Verification

Free-SNI & IOS

Verification of Free-SNI and IOS (or balanced Free-SNI)

Set of Probes

Automatic Verification

Free-SNI & IOS

Verification of Free-SNI and IOS (or balanced Free-SNI)

Set of Probes

Input shares
necessary for
a simulation

Automatic Verification

Free-SNI & IOS

Verification of Free-SNI and IOS (or balanced Free-SNI)

Set of Probes

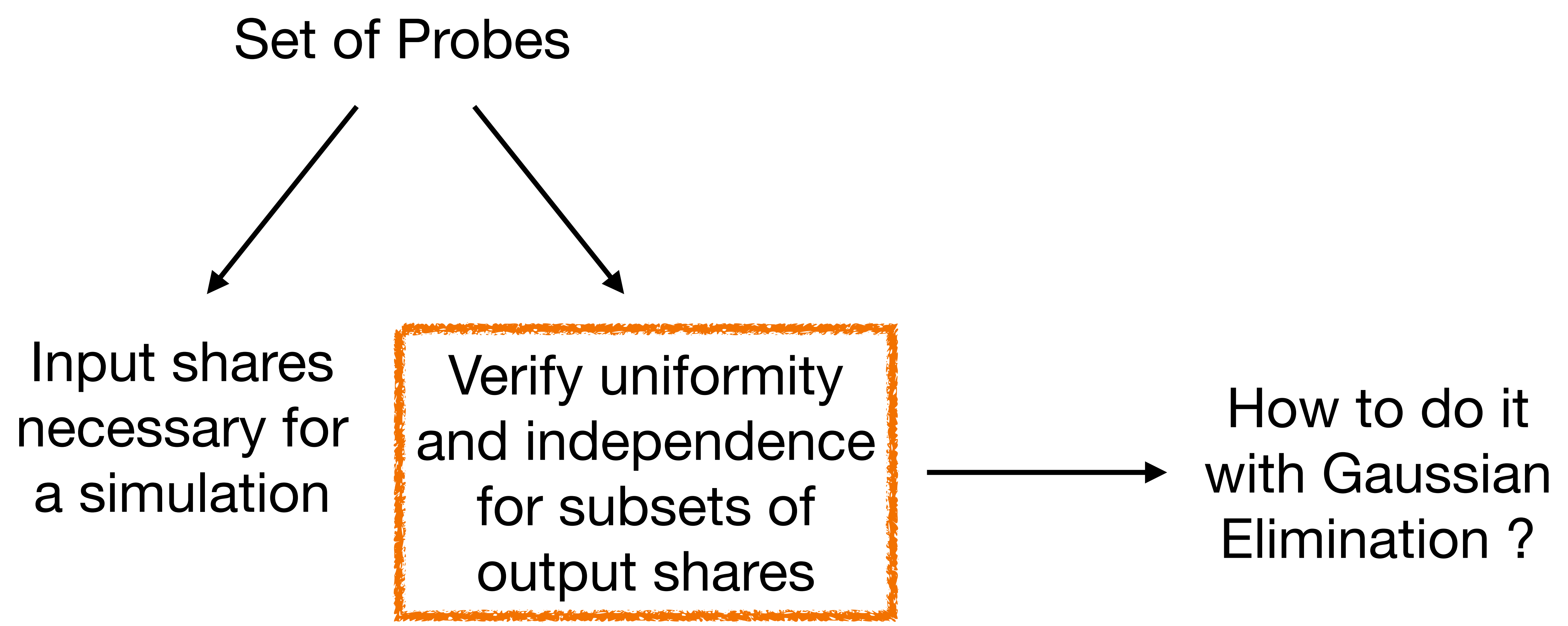
Input shares necessary for a simulation

Verify uniformity and independence for subsets of output shares

Automatic Verification

Free-SNI & IOS

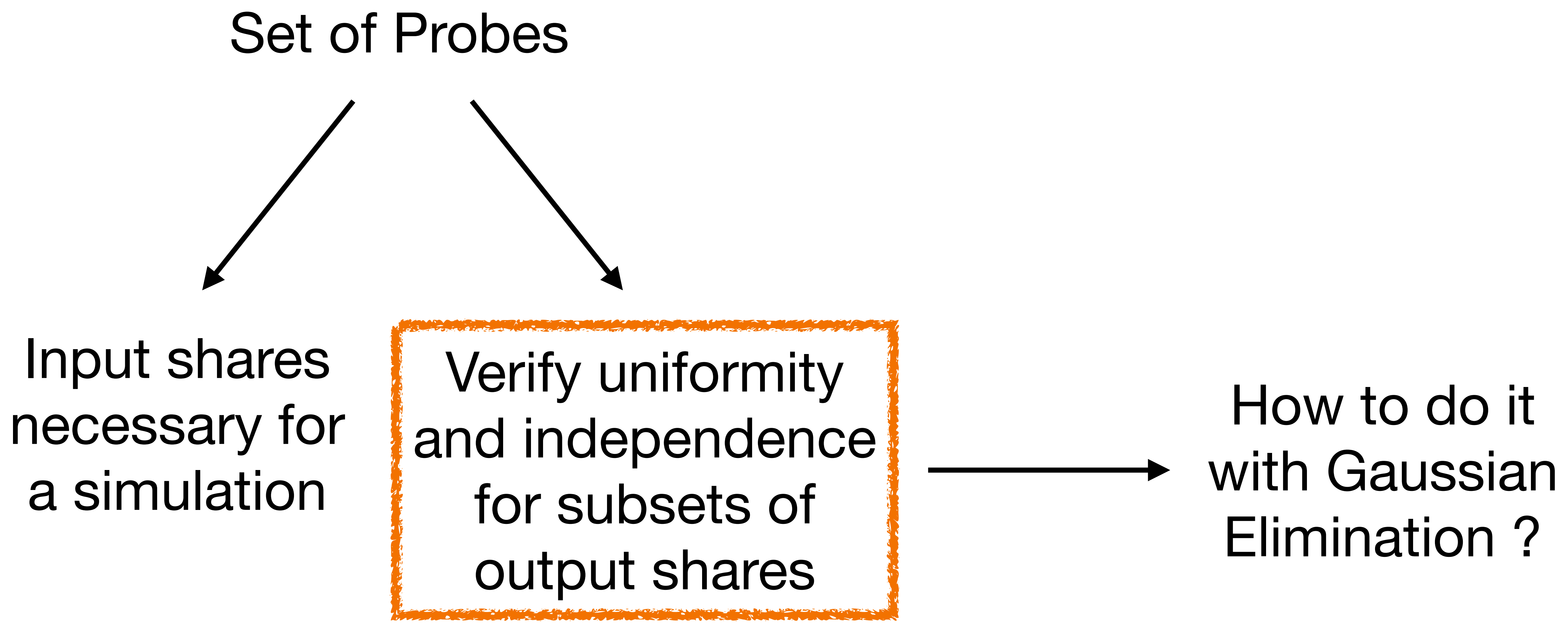
Verification of Free-SNI and IOS (or balanced Free-SNI)



Automatic Verification

Free-SNI & IOS

Verification of Free-SNI and IOS (or balanced Free-SNI)



We show that one Gaussian Elimination is sufficient to find the set of input shares for the simulation and ensure the independence of the necessary subsets of output shares

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$	$a_1 \times b_2$	$a_1 \times b_3$
$a_2 \times b_1$	$a_2 \times b_2$	$a_2 \times b_3$
$a_3 \times b_1$	$a_3 \times b_2$	$a_3 \times b_3$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$	$a_1 \times b_2 + r_{1,2}$	$a_1 \times b_3 + r_{1,3}$
$a_2 \times b_1$	$a_2 \times b_2$	$a_2 \times b_3 + r_{2,3}$
$a_3 \times b_1$	$a_3 \times b_2$	$a_3 \times b_3$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$	$a_1 \times b_2$ $+r_{1,2}$ $+a_2 \times b_1$	$a_1 \times b_3$ $+r_{1,3}$
	$a_2 \times b_2$	$a_2 \times b_3$ $+r_{2,3}$
$a_3 \times b_1$	$a_3 \times b_2$	$a_3 \times b_3$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$

$a_1 \times b_2 + r_{1,2} + a_2 \times b_1$

$a_2 \times b_2$

$a_1 \times b_3 + r_{1,3} + a_3 \times b_1$

$a_2 \times b_3 + r_{2,3} + a_3 \times b_2$

$a_3 \times b_3$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$$a_1 \times b_1 \qquad a_1 \times b_2 + r_{1,2} + a_2 \times b_1 \qquad a_1 \times b_3 + r_{1,3} + a_3 \times b_1$$

$$a_2 \times b_2 \qquad a_2 \times b_3 + r_{2,3} + a_3 \times b_2$$

$$a_3 \times b_3$$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$	$+r_{1,2}$	$a_1 \times b_3 + r_{1,3} + a_3 \times b_1$
$a_2 \times b_2$	$a_1 \times b_2 + r_{1,2} + a_2 \times b_1$	$a_2 \times b_3 + r_{2,3} + a_3 \times b_2$
$a_3 \times b_3$		

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$a_1 \times b_1$	$+r_{1,2}$	$+r_{1,3}$
$a_2 \times b_2$	$a_1 \times b_2 \; +r_{1,2} \; +a_2 \times b_1$	$a_2 \times b_3 +r_{2,3} +a_3 \times b_2$
$a_3 \times b_3$	$a_1 \times b_3 +r_{1,3} +a_3 \times b_1$	

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$$c_1 \leftarrow a_1 \times b_1 + r_{1,2} + r_{1,3}$$

$$c_2 \leftarrow a_2 \times b_2 + a_1 \times b_2 + r_{1,2} + a_2 \times b_1 + r_{2,3}$$

$$c_3 \leftarrow a_3 \times b_3 + a_1 \times b_3 + r_{1,3} + a_3 \times b_1 + a_2 \times b_3 + r_{2,3} + a_3 \times b_2$$

$c_1 + \ldots + c_n = a \times b$ (over $\mathbb{F}_2$)

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Example: 3-share ISW multiplication

$$c_1 \leftarrow a_1 \times b_1 \quad +r_{1,2} \quad +r_{1,3}$$

$$c_2 \leftarrow a_2 \times b_2 + a_1 \times b_2 + r_{1,2} + a_2 \times b_1 + r_{2,3}$$

$$c_3 \leftarrow a_3 \times b_3 + a_1 \times b_3 + r_{1,3} + a_3 \times b_1 + a_2 \times b_3 + r_{2,3} + a_3 \times b_2$$

$c_1 + \ldots + c_n = a \times b$ (over $\mathbb{F}_2$)

Randomness Complexity $\mathcal{O}(n^2)$
Gates Complexity $\mathcal{O}(n^2)$

Constructions Satisfying Free SNI & IOS

ISW Scheme *Ishai, Sahai and Wagner [CRYPTO'03]*

Known

n-share ISW multiplication is $(n-1)$-SNI

Our work

n-share ISW multiplication is only free $(n-2)$-SNI

n-share ISW refresh (by fixing $b_1, \ldots, b_n = 1,0,\ldots,0$) is free $(n-1)$-SNI

Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*

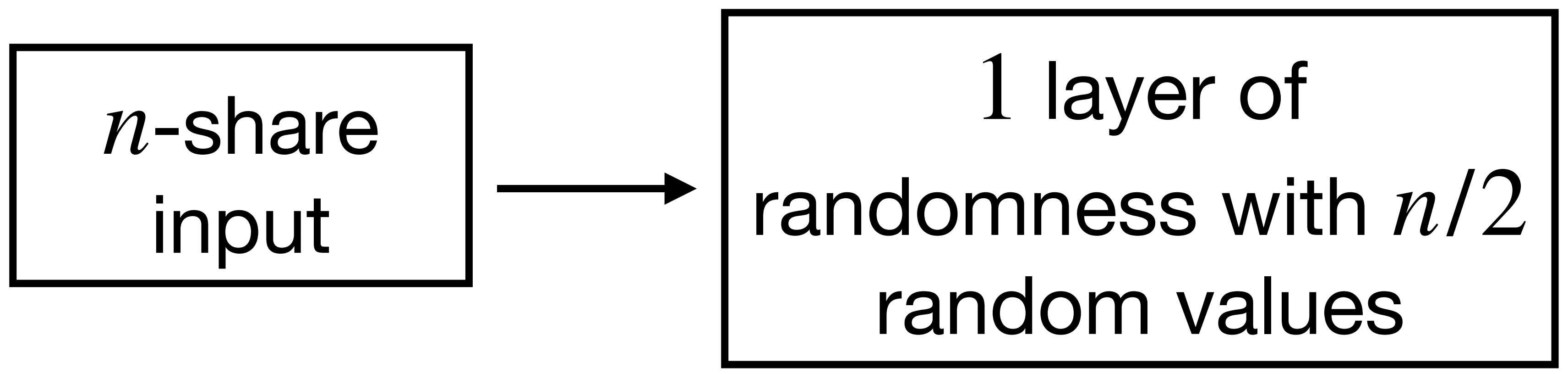
Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*

n-share input

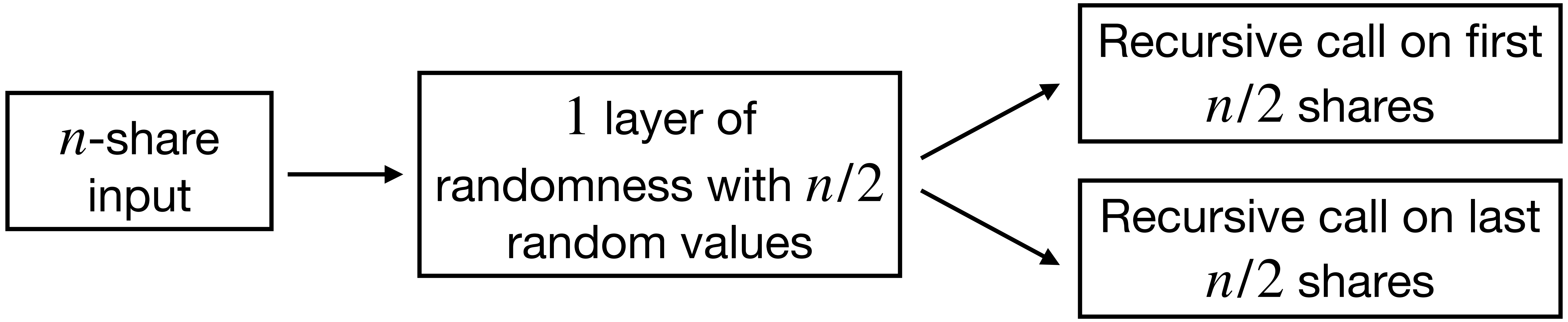
Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*



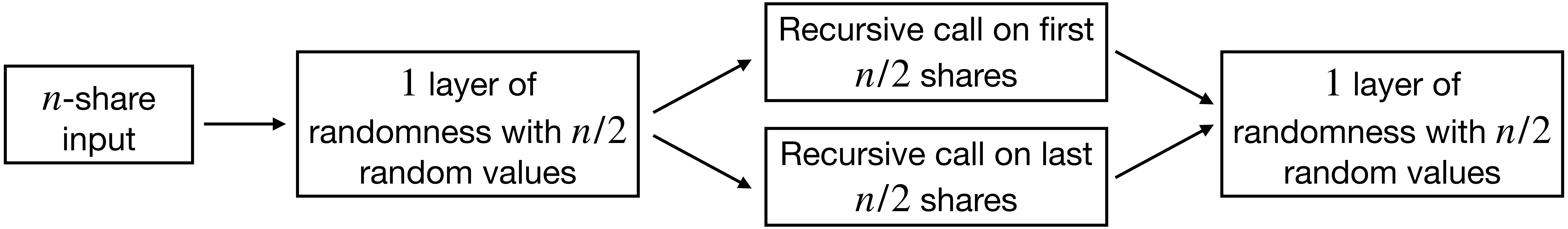
Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*



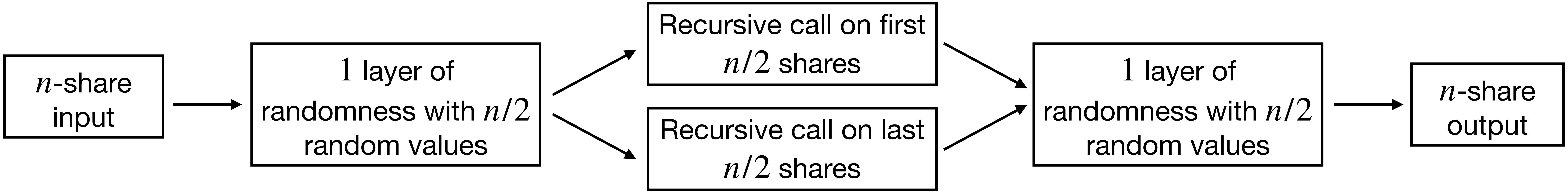
Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*



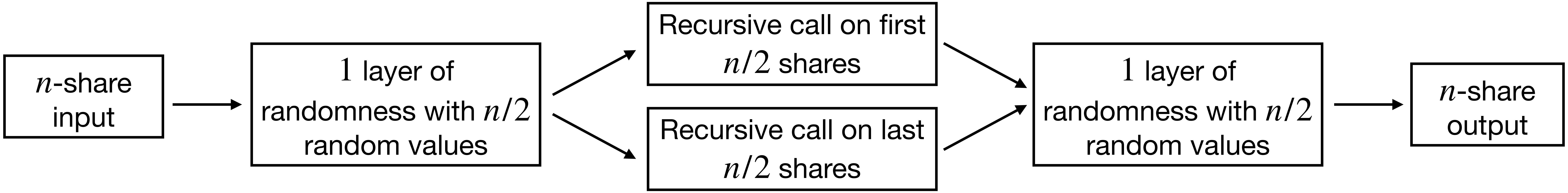
Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*



Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*



Randomness Complexity $\mathcal{O}(n \log n)$

Constructions Satisfying Free SNI & IOS

$\mathcal{O}(n \log n)$ Refresh Gadget *Battistello et al. [TCHES'03]*

Known

n-share $\mathcal{O}(n \log n)$ refresh is $(n-1)$-SNI

Our work

n-share $\mathcal{O}(n \log n)$ refresh is free $(n-1)$-SNI

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Authors use and prove that any $n-1$ shares of the output sharing of a $(n-1)$-SNI gadget are uniform and independent of the input sharing

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Authors use and prove that any $n-1$ shares of the output sharing of a $(n-1)$-SNI gadget are uniform and independent of the input sharing

↓

Not necessarily true when we have probes inside the gadget

Breaks the correctness of the strategy

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

$$c_1 \leftarrow a_1 \times b_1 + r_{1,2} + r_{1,3} + r_{1,4}$$

$$c_2 \leftarrow a_2 \times b_2 + (a_1 \times b_2 + r_{1,2} + a_2 \times b_1) + r_{2,3} + r_{2,4}$$

$$c_3 \leftarrow a_3 \times b_3 + (a_1 \times b_3 + r_{1,3} + a_3 \times b_1) + (a_2 \times b_3 + r_{2,3} + a_3 \times b_2) + r_{3,4}$$

$$c_4 \leftarrow a_4 \times b_4 + (a_1 \times b_4 + r_{1,4} + a_4 \times b_1) + (a_2 \times b_4 + r_{2,4} + a_4 \times b_2) + (a_3 \times b_4 + r_{3,4} + a_4 \times b_3)$$

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

$$c_1 \leftarrow a_1 \times b_1 + r_{1,2} + r_{1,3} + r_{1,4}$$

$$c_2 \leftarrow a_2 \times b_2 + (a_1 \times b_2 + r_{1,2} + a_2 \times b_1) + r_{2,3} + r_{2,4}$$

$$c_3 \leftarrow \boxed{a_3 \times b_3 + (a_1 \times b_3 + r_{1,3} + a_3 \times b_1) + (a_2 \times b_3 + r_{2,3} + a_3 \times b_2)} + r_{3,4}$$

$$c_4 \leftarrow a_4 \times b_4 + (a_1 \times b_4 + r_{1,4} + a_4 \times b_1) + (a_2 \times b_4 + r_{2,4} + a_4 \times b_2) + \boxed{(a_3 \times b_4 + r_{3,4} + a_4 \times b_3)}$$

c_3 is not uniform independent conditioned the probes

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Using free $(n-2)$-SNI multiplication and refresh fixes the flaw in the TPC proof (uniformity of a subset of the output shares, conditioned on the probes)

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Using free $(n-2)$-SNI multiplication and refresh fixes the flaw in the TPC proof (uniformity of a subset of the output shares, conditioned on the probes)

↓

The results in TPC are still correct, because the authors use ISW, which is free $(n-2)$-SNI

Tight Private Circuits

The Return

- Secure tight composition in the probing model by inserting refresh gadgets only when needed
- Uses $(n-1)$-SNI multiplication and refresh gadgets

Using free $(n-2)$-SNI multiplication and refresh fixes the flaw in the TPC proof (uniformity of a subset of the output shares, conditioned on the probes)

↓

The results in TPC are still correct, because the authors use ISW, which is free $(n-2)$-SNI

Our results generalize TPC to any free $(n-2)$-SNI gadgets, like the $\mathcal{O}(n \log n)$ refresh gadget instead of the ISW refresh gadget (improved efficiency)

Composition in the Region Probing Model

Composition in the Region Probing Model

Framework by ***Goudarzi et al. [TCHES'21]*** provides region probing security by inserting **IOS refresh gadgets** between probing secure regions

Composition in the Region Probing Model

Framework by ***Goudarzi et al. [TCHES'21]*** provides region probing security by inserting **IOS refresh gadgets** between probing secure regions

↓

We adapt the generalization of TPC to region probing security

Composition in the Region Probing Model

Framework by ***Goudarzi et al. [TCHES'21]*** provides region probing security by inserting **IOS refresh gadgets** between probing secure regions

↓

We adapt the generalization of TPC to region probing security

- Use any IOS gadgets (not only refresh)
- Reduced number of IOS refresh gadgets to insert
- Increased efficiency and generalization to more IOS gadgets from the literature

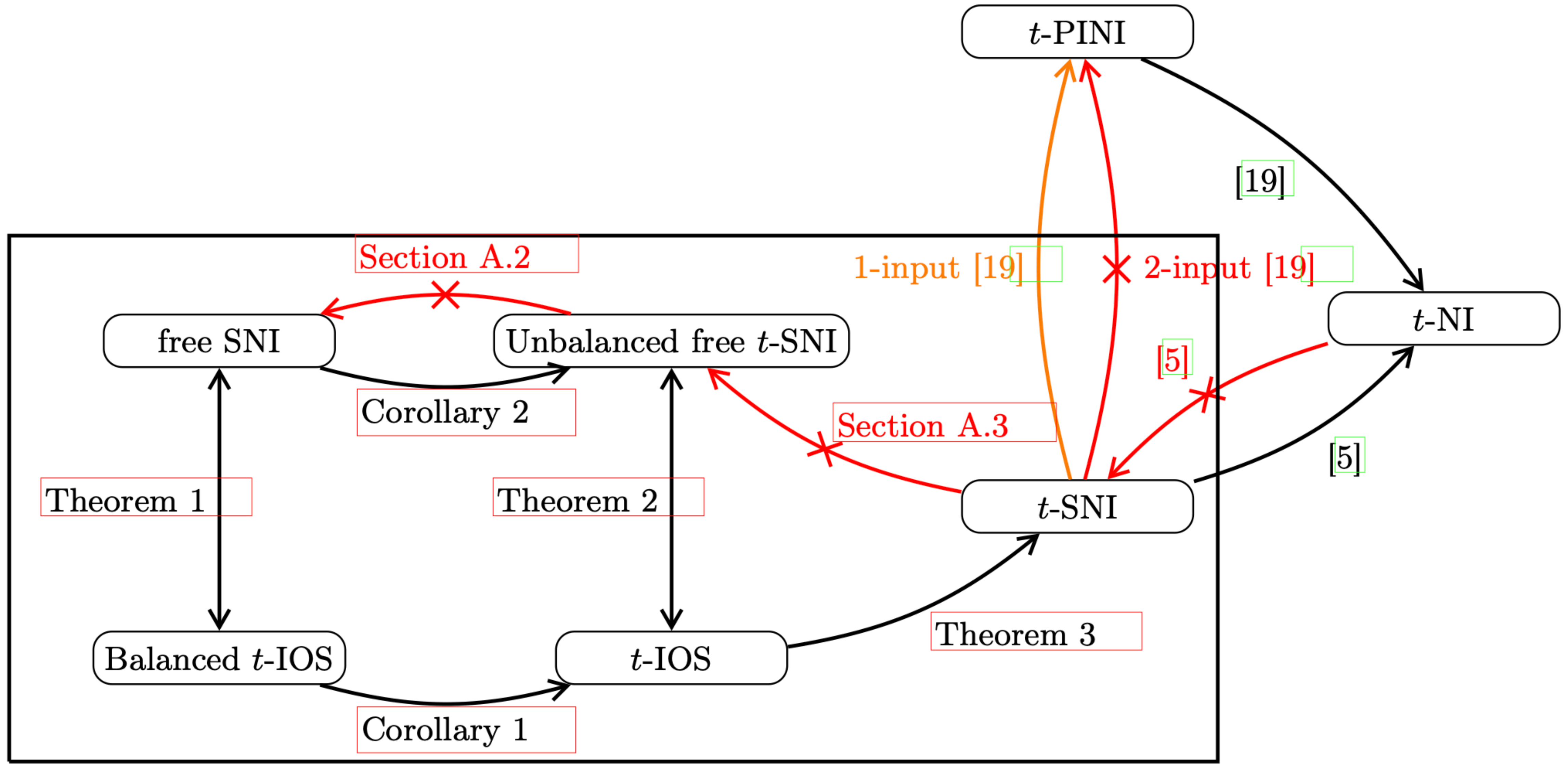
Conclusion

Conclusion

- Equivalence of Free-SNI and IOS, notions introduced in different contexts and for different purposes
- Both can be efficiently verified like other probing notions (SNI, NI, PINI, …) using IronMask
- Well-known gadgets from the literature already satisfy these stronger notions
- Both notions lead to more efficient composition in the probing and region probing models

Conclusion

- Equivalence of Free-SNI and IOS, notions introduced in different contexts and for different purposes
- Both can be efficiently verified like other probing notions (SNI, NI, PINI, …) using IronMask
- Well-known gadgets from the literature already satisfy these stronger notions
- Both notions lead to more efficient composition in the probing and region probing models



Thank you ! Any questions ?