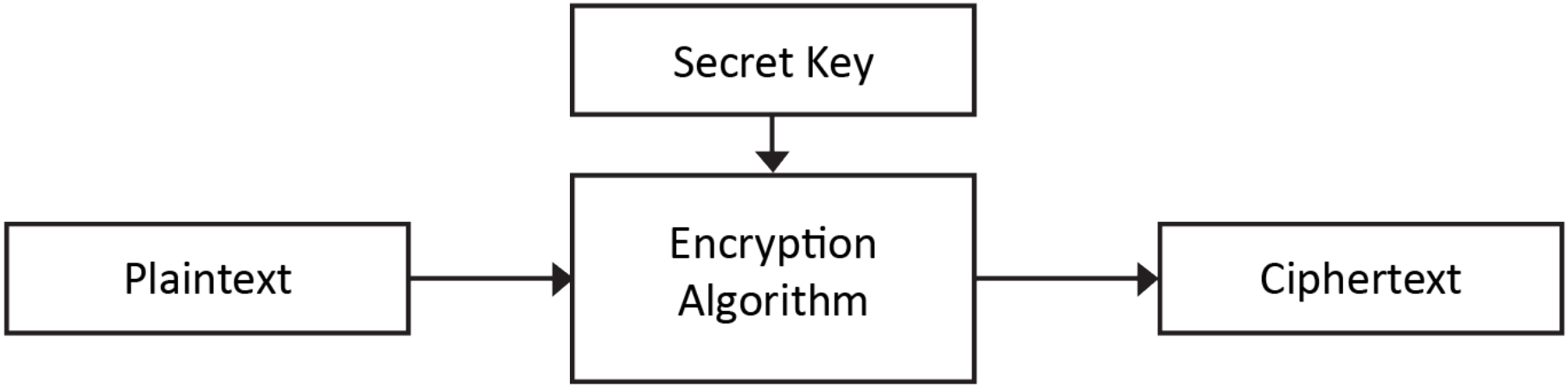
Secure and Verified Cryptographic Implementations in The Random Probing Model

Abdel Taleb[1,2]

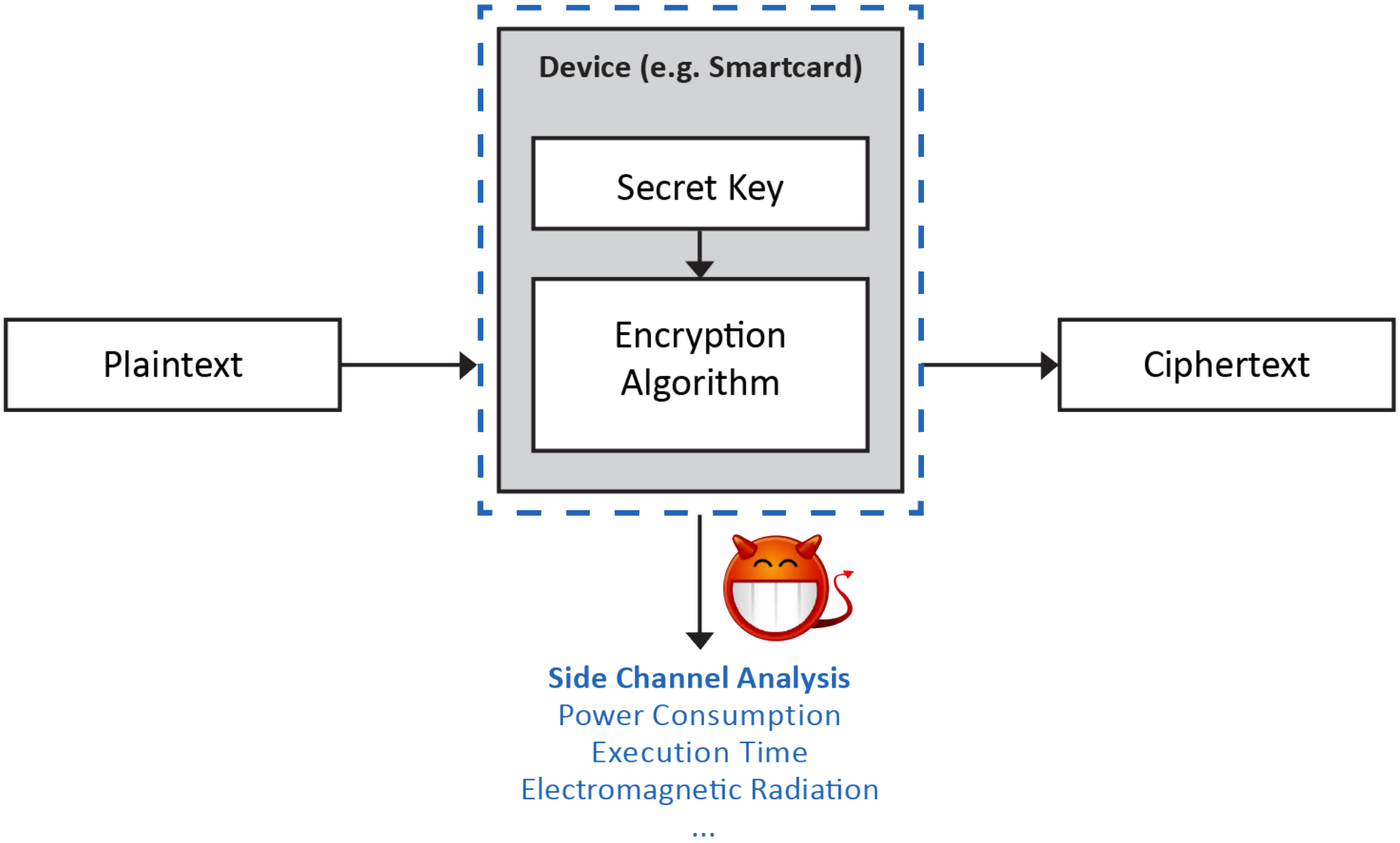
[1] CryptoExperts, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

December 3rd, 2021

Side-Channel Attacks



Side-Channel Attacks



Masking

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow (x_1, \ldots, x_n) \in \mathbb{K}^n$$

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow \underbrace{(x_1, \ldots, x_n)}_{\substack{\text{shares of } x \\ x_1+\ldots+x_n=x}} \in \mathbb{K}^n$$

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow \underbrace{(x_1, \ldots, x_n)}_{\substack{\text{shares of } x \\ x_1+\ldots+x_n=x}} \in \mathbb{K}^n$$

$(+, \times, ||)$ operations over $\mathbb{K}$

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow \underbrace{(x_1, \ldots, x_n)}_{\substack{\text{shares of } x \\ x_1+\ldots+x_n=x}} \in \mathbb{K}^n$$

$(+, \times, ||)$ operations over $\mathbb{K} \longrightarrow (G_{\text{add}}, G_{\text{mult}}, G_{\text{copy}}, \mathbf{G}_{\text{refresh}})$ n-share circuits over $\mathbb{K}$

Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow \underbrace{(x_1, \ldots, x_n)}_{\substack{\text{shares of } x \\ x_1+\ldots+x_n=x}} \in \mathbb{K}^n$$

$(+, \times, ||)$ operations over $\mathbb{K} \longrightarrow (G_{\text{add}}, G_{\text{mult}}, G_{\text{copy}}, \mathbf{G}_{\text{refresh}})$ n-share circuits over $\mathbb{K}$

Example $G_{\text{add}}(a, b) = c$ with $n = 2$

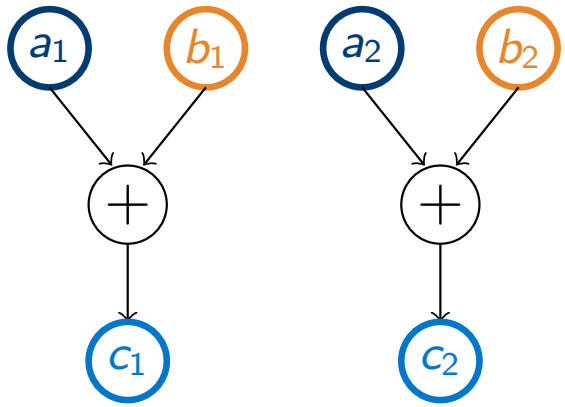
Masking

Masking countermeasure (sensitive variable x over field $\mathbb{K}$)

$$x \longrightarrow \underbrace{(x_1, \ldots, x_n)}_{\substack{\text{shares of } x \\ x_1+\ldots+x_n=x}} \in \mathbb{K}^n$$

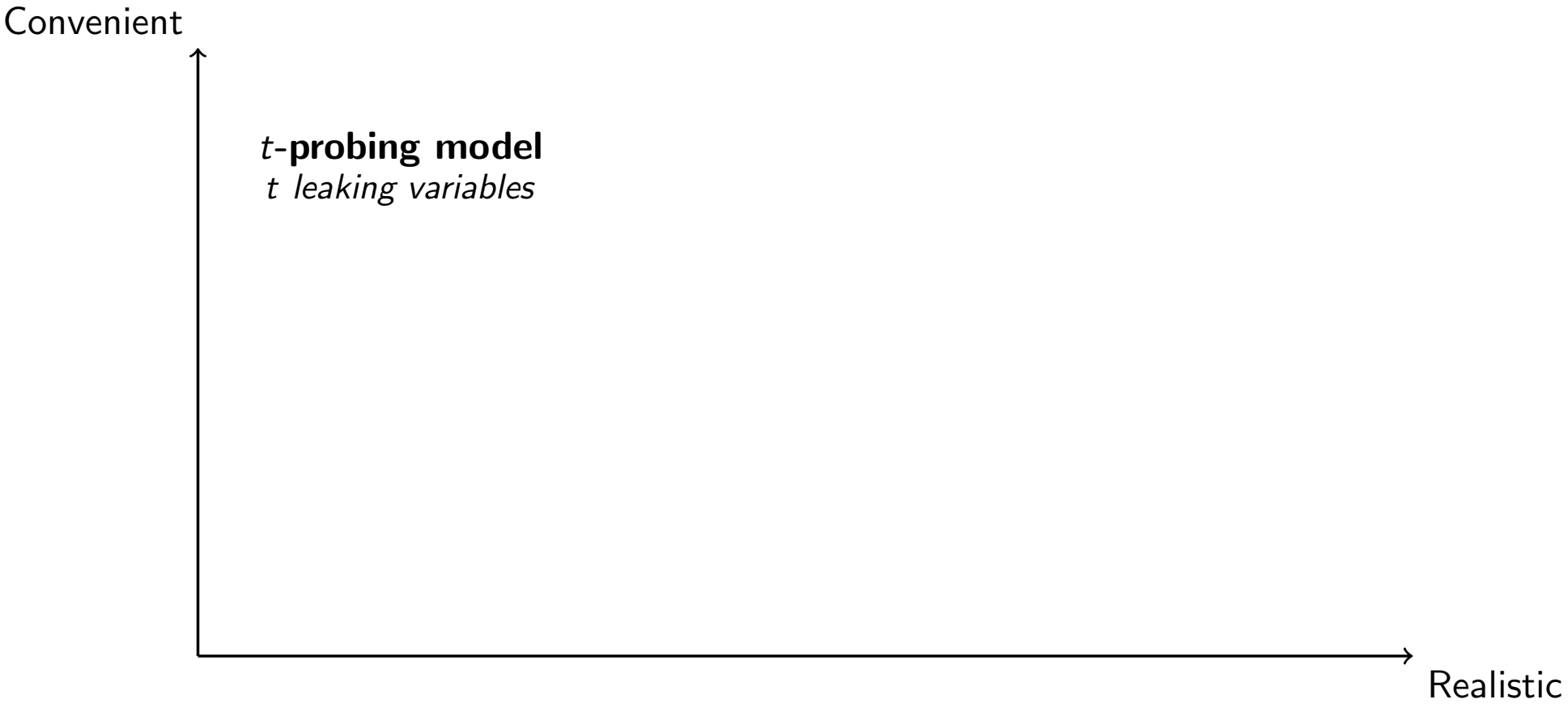
$(+, \times, ||)$ operations over $\mathbb{K} \longrightarrow (G_{\text{add}}, G_{\text{mult}}, G_{\text{copy}}, \mathbf{G}_{\text{refresh}})$ n-share circuits over $\mathbb{K}$

Example $G_{\text{add}}(a, b) = c$ with $n = 2$

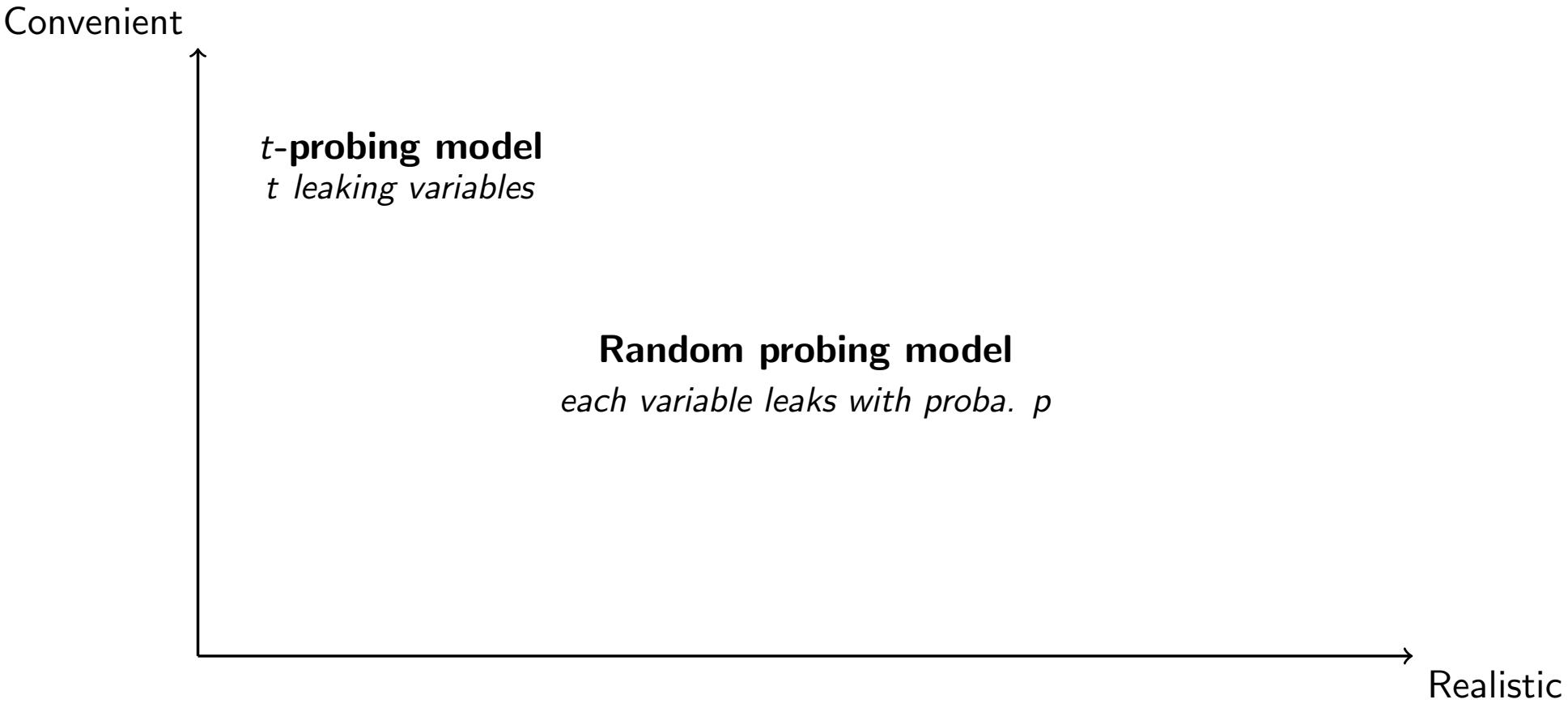


Leakage Models

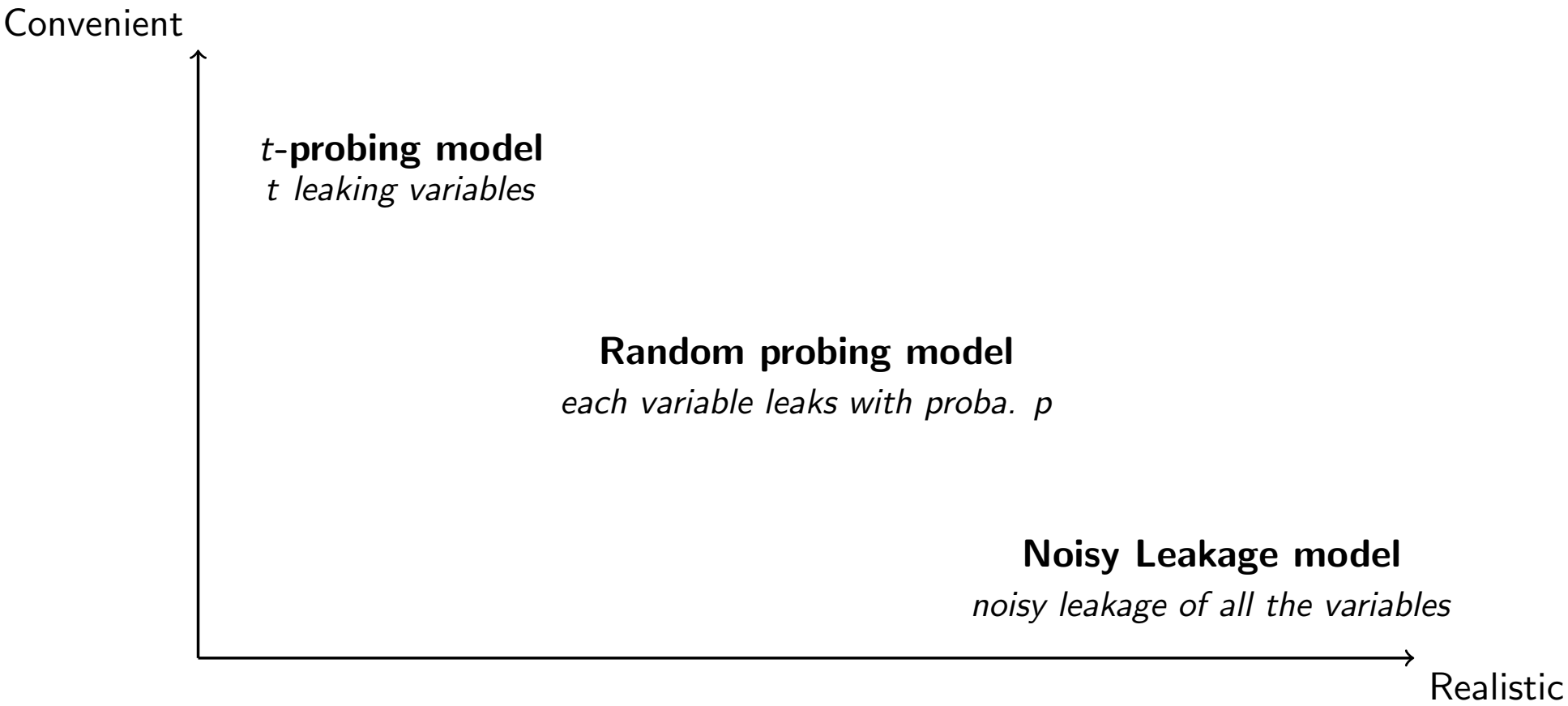
Leakage Models



Leakage Models



Leakage Models



PhD Overview

PhD Overview

- Security of masking in the **Random Probing (RP) Model**

PhD Overview

- Security of masking in the **Random Probing (RP) Model**

- RP-secure gadgets composition (RP composition)

PhD Overview

- Security of masking in the **Random Probing (RP) Model**
- RP-secure gadgets composition (RP composition)
- RP-secure security level amplification (RP expansion)

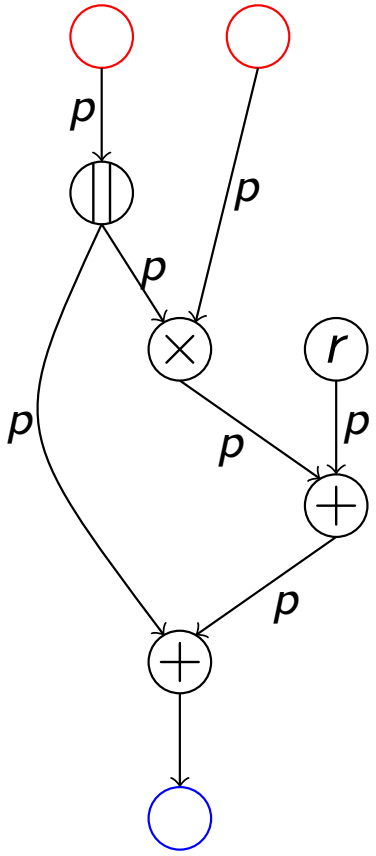
PhD Overview

- Security of masking in the **Random Probing (RP) Model**
- RP-secure gadgets composition (RP composition)
- RP-secure security level amplification (RP expansion)
- **VRAPS:** (V)erifier of (RA)ndom (P)robing (S)ecurity

PhD Overview

- Security of masking in the **Random Probing (RP) Model**
- RP-secure gadgets composition (RP composition)
- RP-secure security level amplification (RP expansion)
- **VRAPS:** (V)erifier of (RA)ndom (P)robing (S)ecurity
- **IronMask:** Versatile Verifier of Masking Security

RP Security

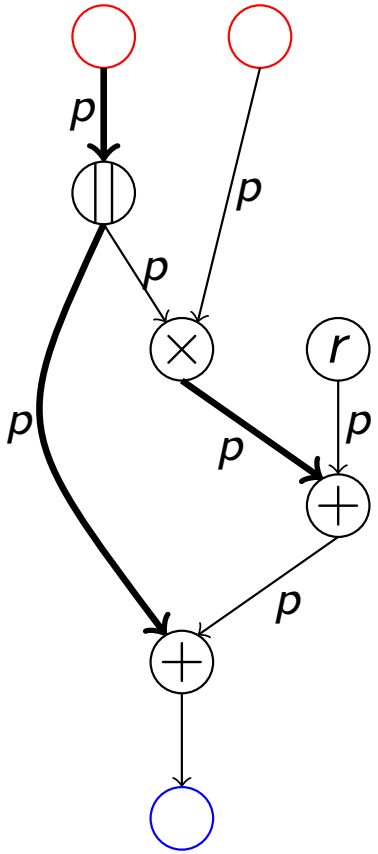


Add Mult.

Copy Random

(p, ε)-RP Security

RP Security



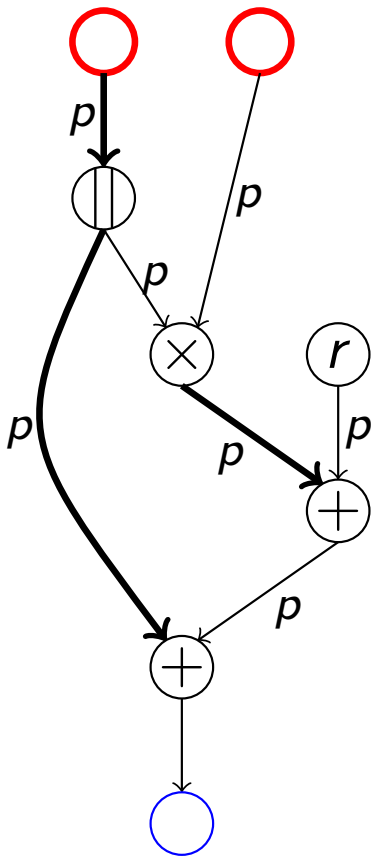
⊕ Add ⊗ Mult.

Copy r Random

(p, ε)-RP Security

W set of wires

RP Security



⊕ Add ⊗ Mult.

Copy r Random

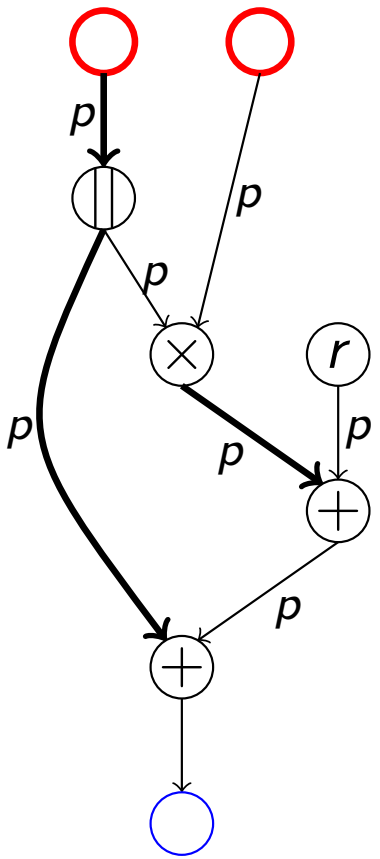
(p, ε)-RP Security

W set of wires

↓

Independent from secret inputs ?

RP Security



⊕ Add ⊗ Mult.

Copy r Random

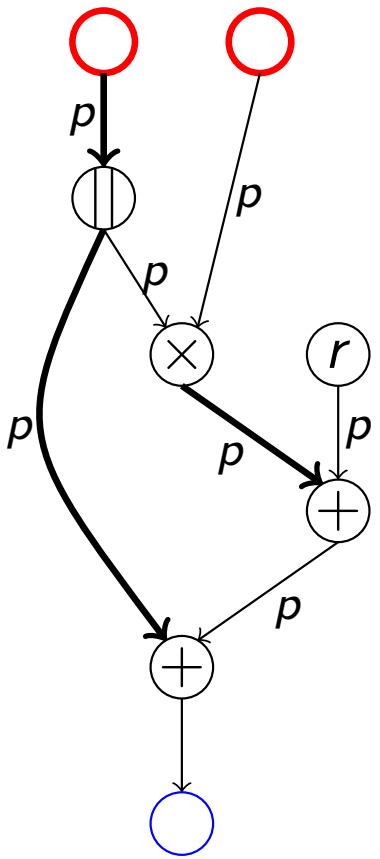
(p, ε)-RP Security

W set of wires

Independent from secret inputs ?

yes no

RP Security



⊕ Add ⊗ Mult.

Copy r Random

(p, ε)-RP Security

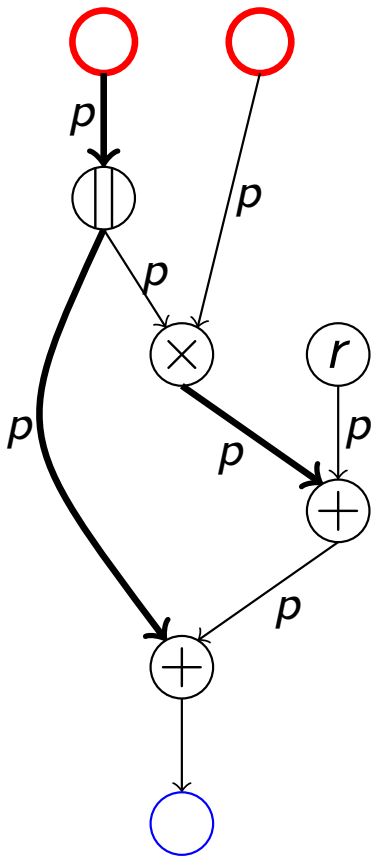
W set of wires

Independent from secret inputs ?

yes no

Simulation Success

RP Security



⊕ Add ⊗ Mult.

Copy r Random

(p, ε)-RP Security

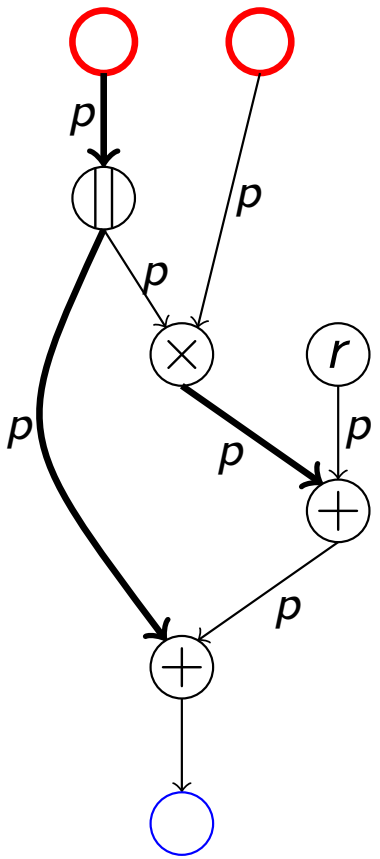
W set of wires

Independent from secret inputs ?

yes no

Simulation Success *Simulation Failure*

RP Security



⊕ Add ⊗ Mult.

Copy r Random

(p, ε)-RP Security

W set of wires

Independent from secret inputs ?

yes

no

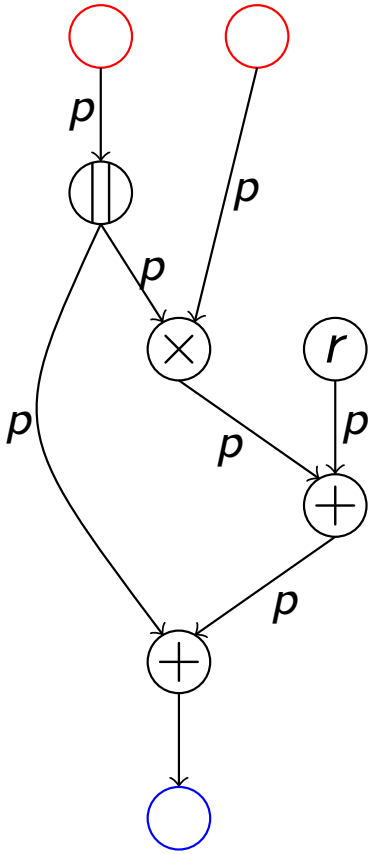
Simulation Success

Simulation Failure

Failure Probability ε

RP Security

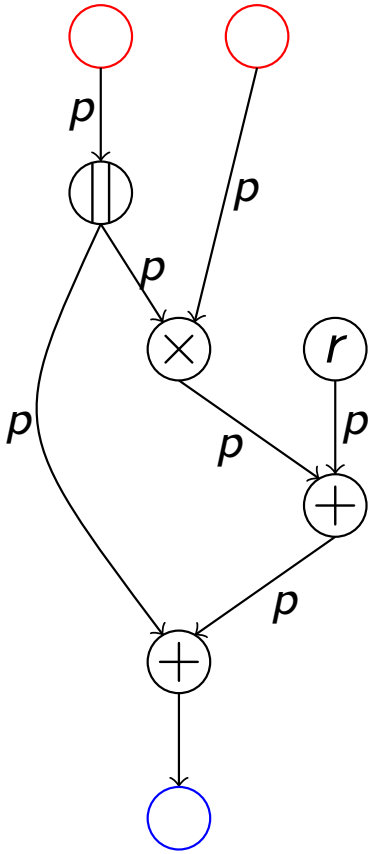
Formal Verification : Method



s: number of wires

RP Security

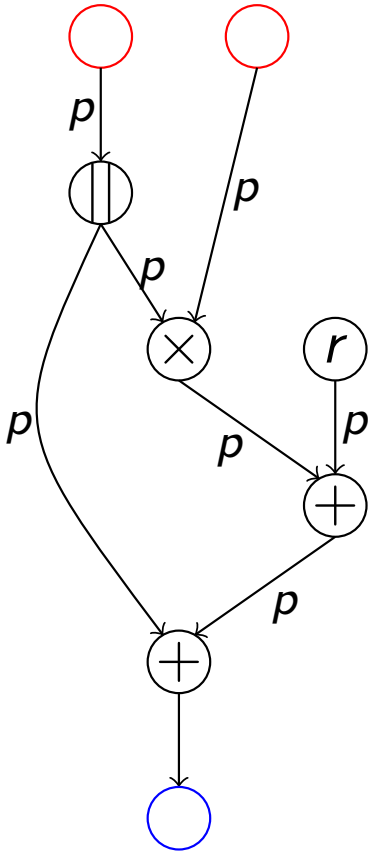
Formal Verification : Method



W set of wires

$$Pr(W) = p^{|W|}(1-p)^{s-|W|}$$

s: number of wires



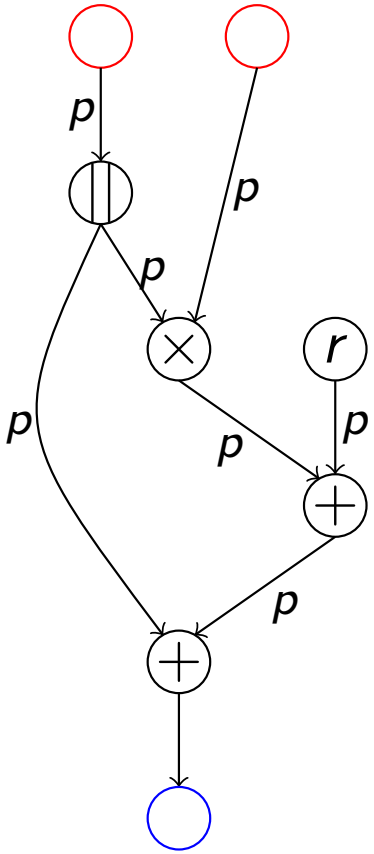
W set of wires

$$Pr(W) = p^{|W|}(1-p)^{s-|W|}$$

Failure probability ϵ

$$\epsilon = f(p) = \sum_{\substack{W \\ \text{Failure on W}}} p^{|W|}(1-p)^{s-|W|}$$

s: number of wires



s: number of wires

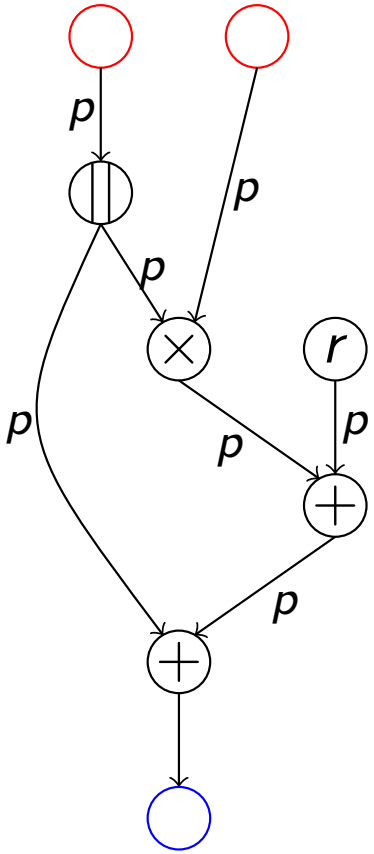
W set of wires

$$Pr(W) = p^{|W|}(1-p)^{s-|W|}$$

Failure probability ϵ

$$\epsilon = f(p) = \sum_{\substack{W \\ \text{Failure on W}}} p^{|W|}(1-p)^{s-|W|}$$

$\mathbf{c_i}$: number of W of size i with *Simulation Failure*



s: number of wires

W set of wires

$$Pr(W) = p^{|W|}(1-p)^{s-|W|}$$

Failure probability ϵ

$$\epsilon = f(p) = \sum_{\substack{W \\ \text{Failure on W}}} p^{|W|}(1-p)^{s-|W|}$$

$\mathbf{c_i}$: number of W of size i with *Simulation Failure*

$$\epsilon = \sum_{i=1}^{s} c_i p^i (1-p)^{s-i}$$

RP Security

Formal Verification : Algorithm (VRAPS Tool)

Input: circuit with s wires
Output: coefficients $c_1, \ldots, c_s$

1: $c \leftarrow (0, \ldots, 0)$

7: **return** c

RP Security

Formal Verification : Algorithm (VRAPS Tool)

Input: circuit with s wires
Output: coefficients $c_1, \ldots, c_s$

1: $c \leftarrow (0, \ldots, 0)$
2: **for** $i = 1$ to s **do**

6: **end for**
7: **return** c

RP Security

Formal Verification : Algorithm (VRAPS Tool)

Input: circuit with s wires
Output: coefficients $c_1, \ldots, c_s$

1: $c \leftarrow (0, \ldots, 0)$
2: **for** $i = 1$ to s **do**
3: $\quad L \leftarrow \{\text{all } W \text{ of size } i\}$

6: **end for**
7: **return** c

RP Security

Formal Verification : Algorithm (VRAPS Tool)

Input: circuit with s wires
Output: coefficients $c_1, \ldots, c_s$

1: $c \leftarrow (0, \ldots, 0)$
2: **for** $i = 1$ to s **do**
3: $\quad L \leftarrow \{\text{all } W \text{ of size } i\}$
4: $\quad$ Apply rules inspired from **maskVerif** on L *[Barthe et al. - EuroCrypt 2015]*

6: **end for**
7: **return** c

RP Security

Formal Verification : Algorithm (VRAPS Tool)

Input: circuit with s wires
Output: coefficients $c_1, \ldots, c_s$

1: $c \leftarrow (0, \ldots, 0)$
2: **for** $i = 1$ to s **do**
3: $\quad L \leftarrow \{\text{all } W \text{ of size } i\}$
4: $\quad$ Apply rules inspired from **maskVerif** on L *[Barthe et al. - EuroCrypt 2015]*
5: $\quad c_i \leftarrow$ Nb. of failures in L
6: **end for**
7: **return** c

RP Composition

Goal: Achieve global random probing security

RP Composition

Goal: Achieve global random probing security

(t, p, ϵ)-**Random probing composable** n-share gadgets

G_{add} G_{copy}

G_{mult}

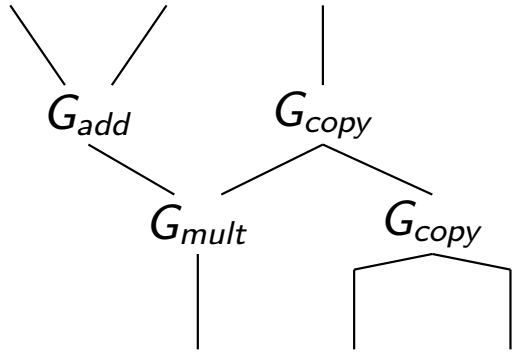
RP Composition

Goal: Achieve global random probing security

(t, p, ϵ)-**Random probing composable** n-share gadgets $\Longrightarrow$ $(p, |C|.\epsilon)$-**Random probing secure** circuit C

G_{add} G_{copy}

G_{mult}



RP Composition

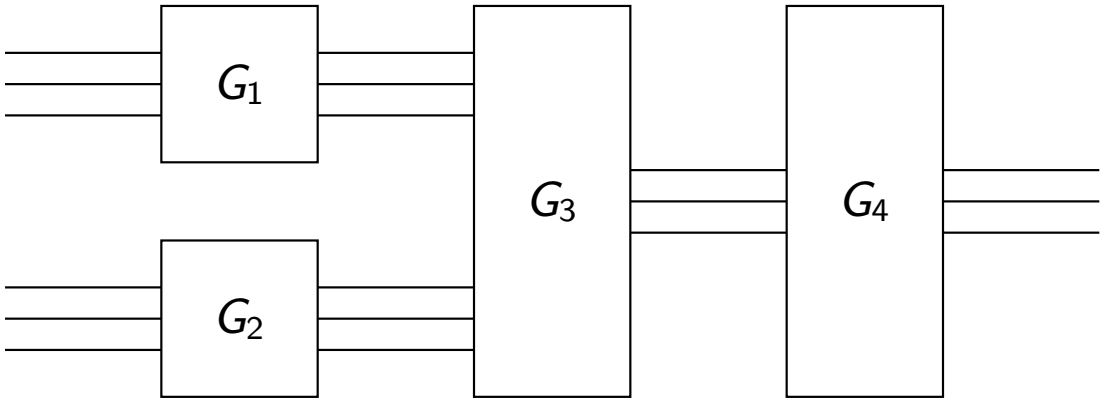
Definition

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

RP Composition

Definition

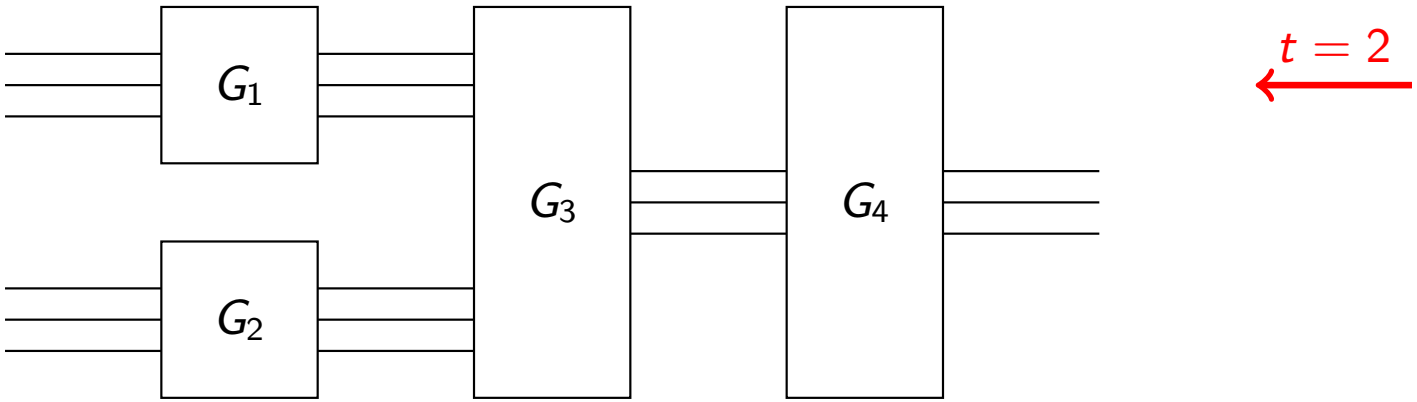
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

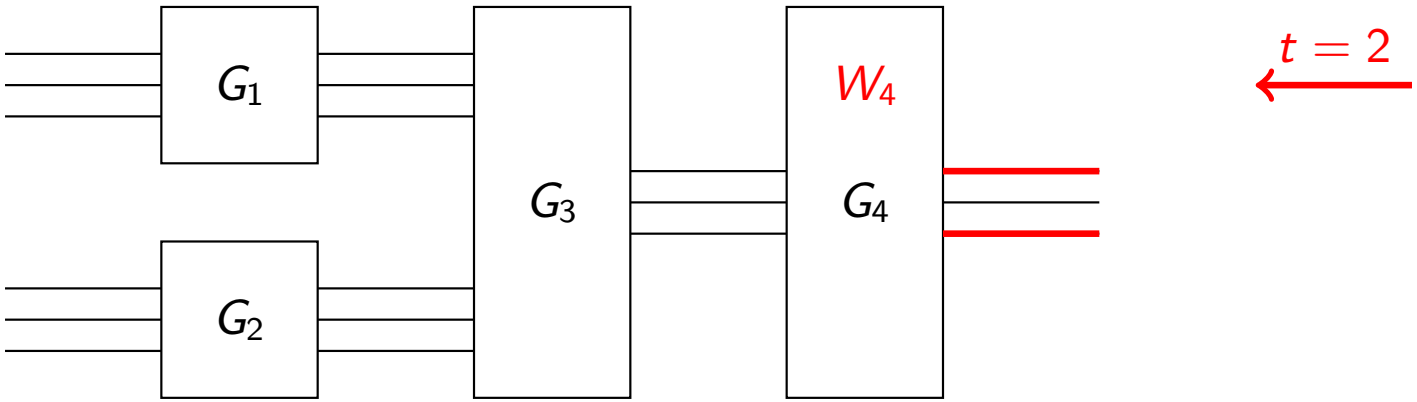
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

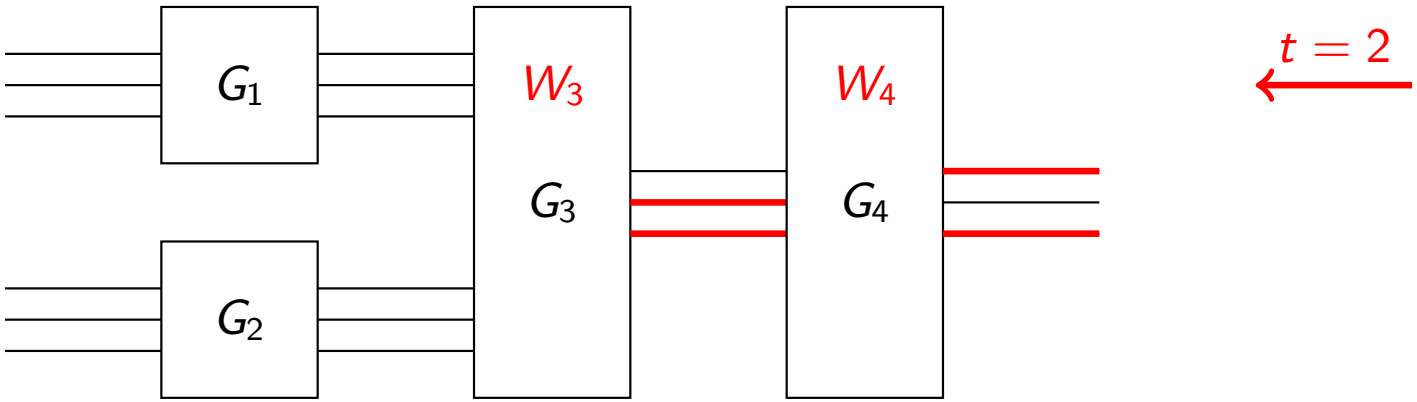
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

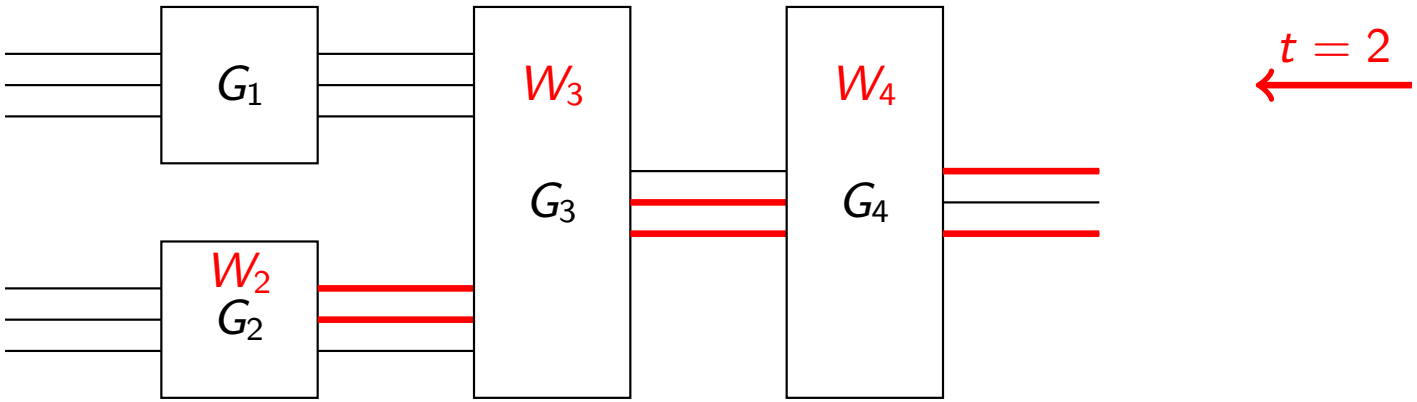
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

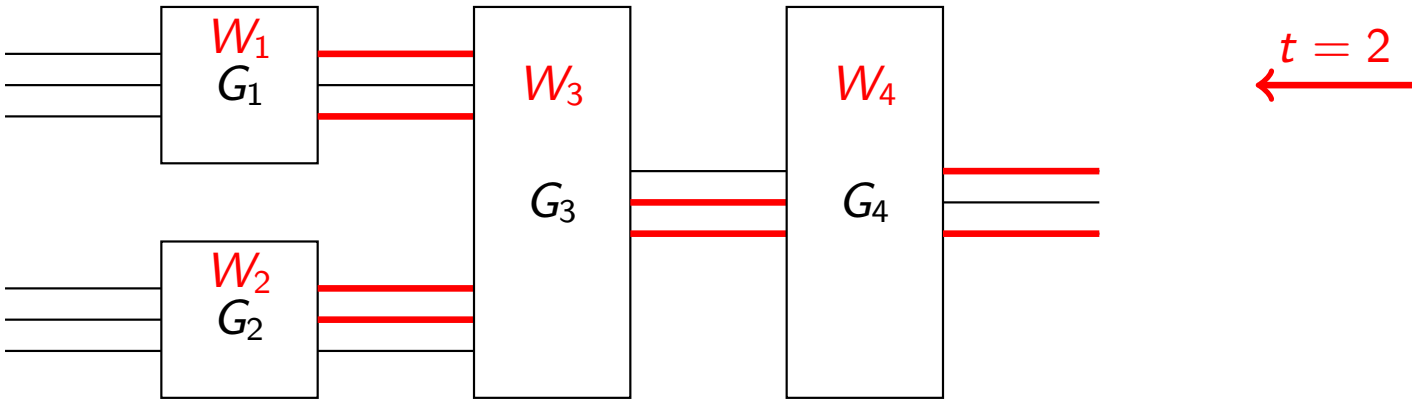
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

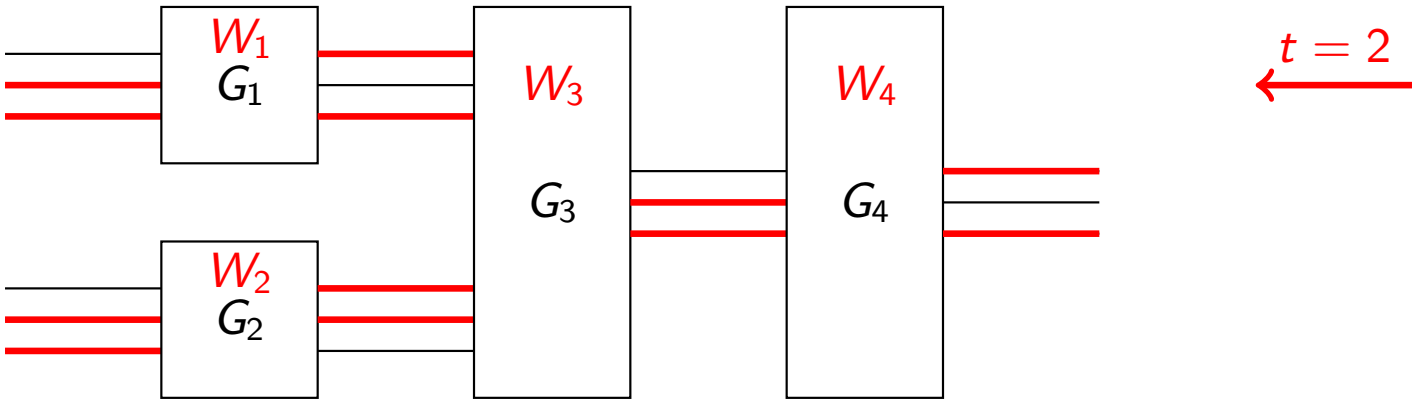
Composition of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Composition

Definition

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



RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$

RP Expansion

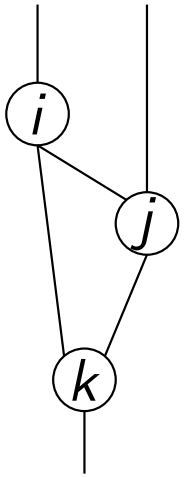
Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$

RP Expansion

Illustration

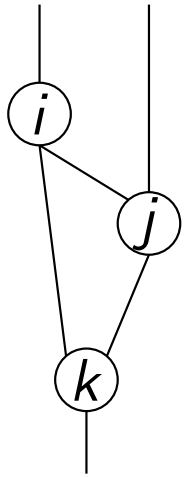
Using n-share gadgets $G_1, \ldots, G_\beta$



RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$

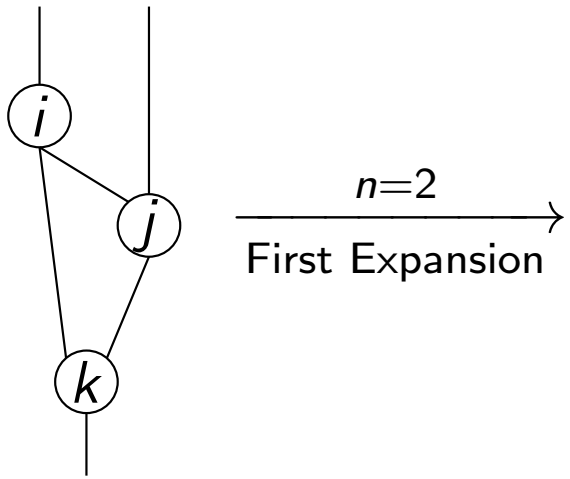


Leakage probability
p

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$

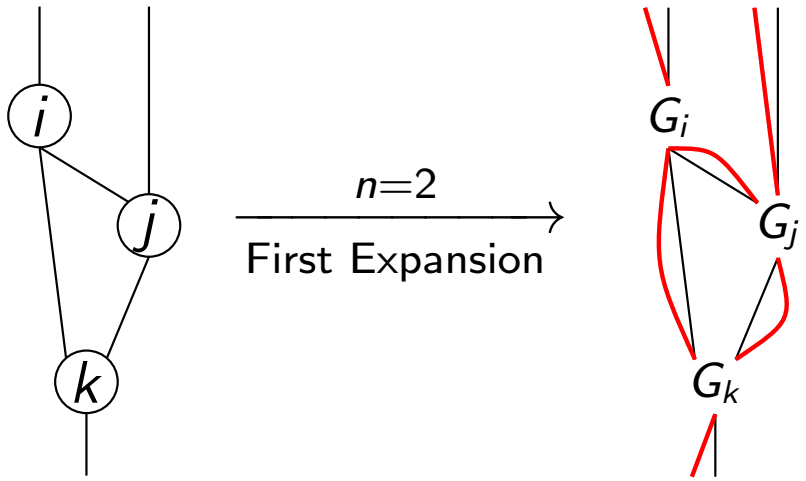


Leakage probability
p

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$

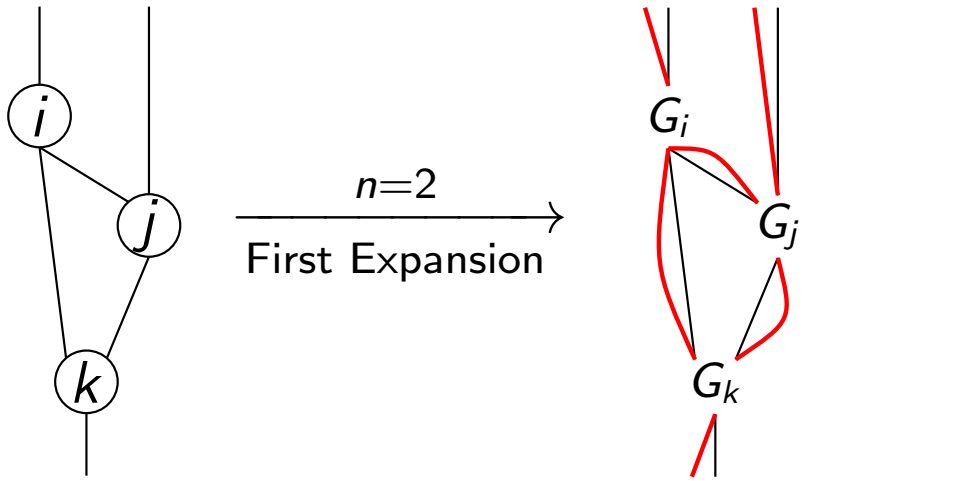


Leakage probability p

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



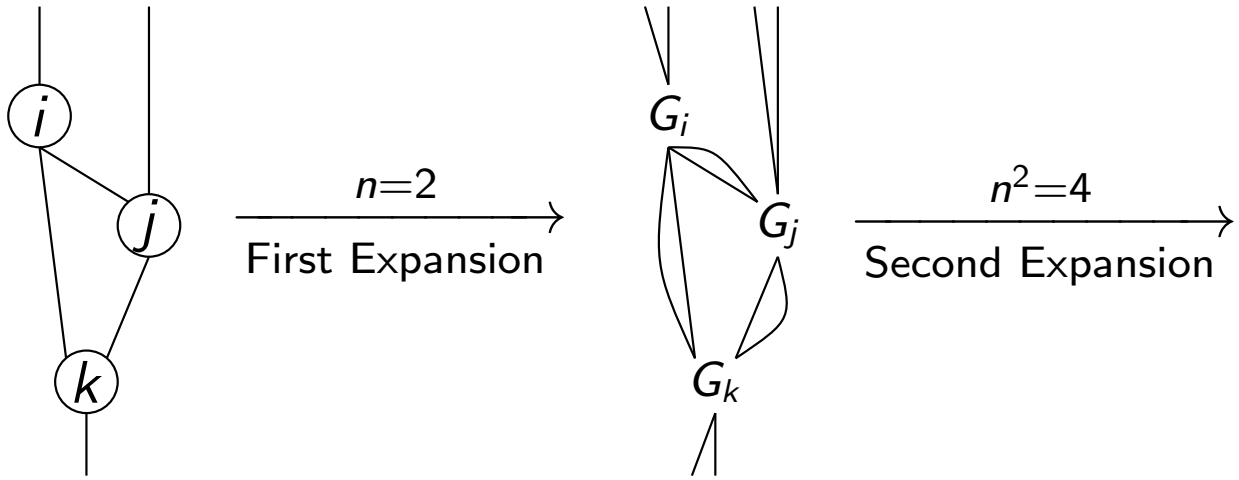
Leakage probability
p

Simulation Failure
ε

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



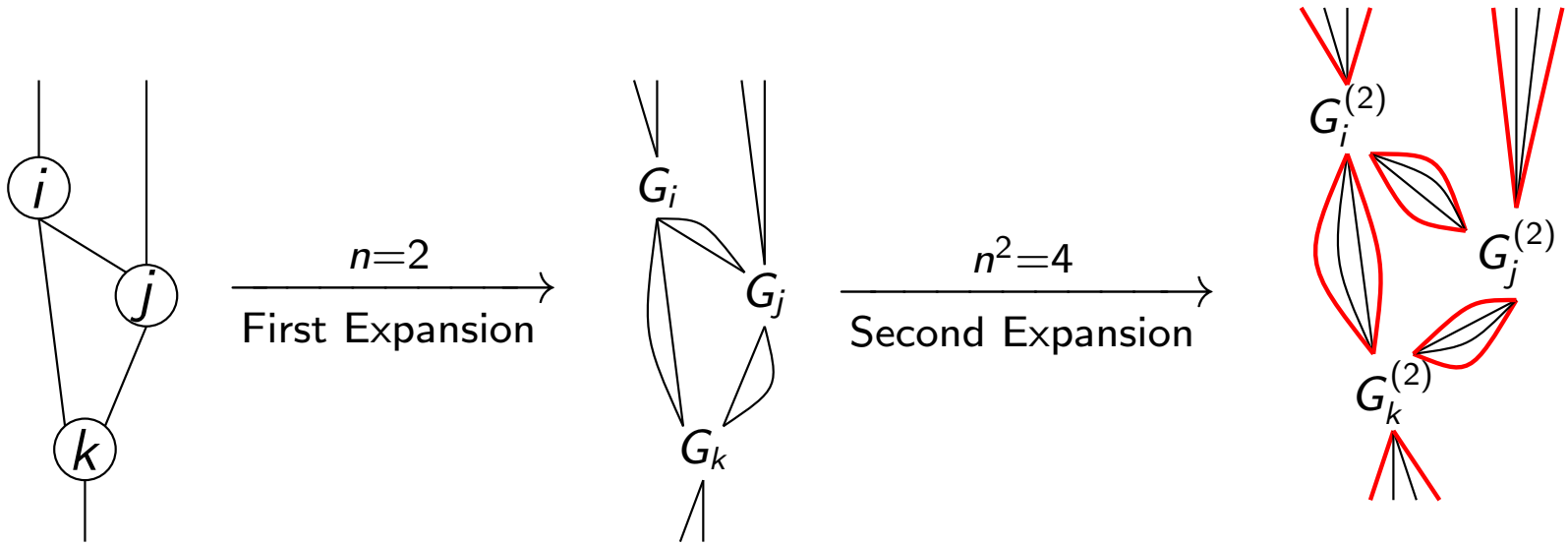
Leakage probability
p

Simulation Failure
ε

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



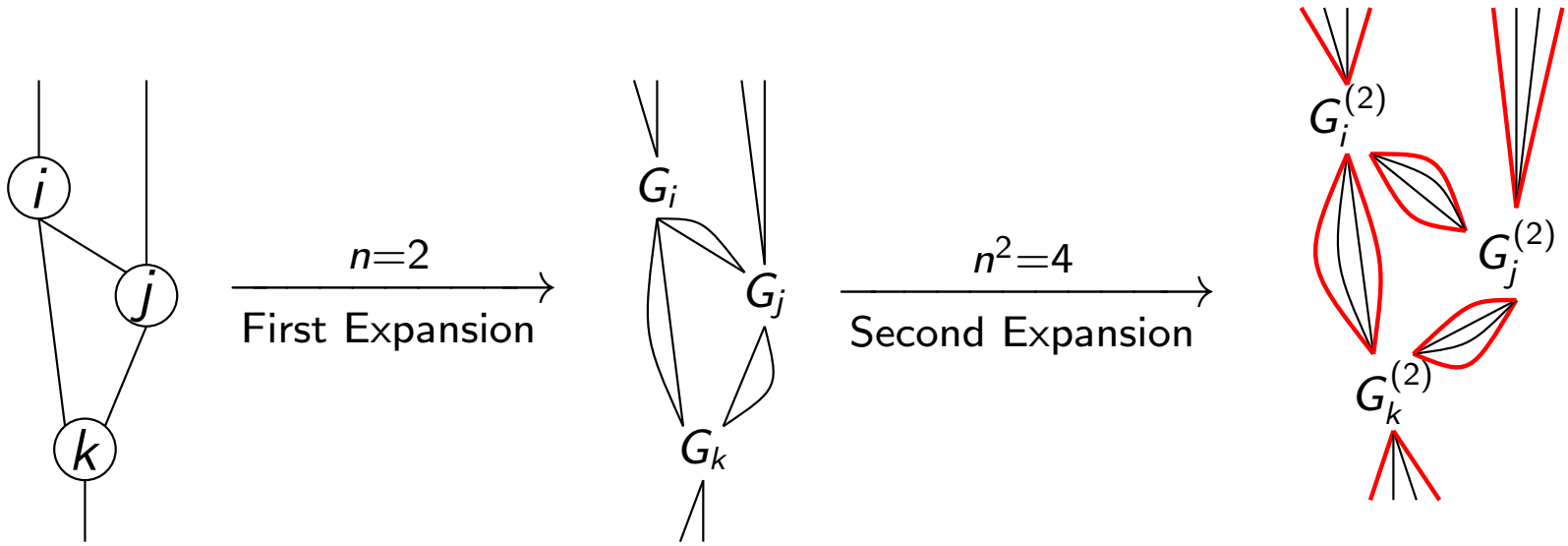
Leakage probability
p

Simulation Failure
ε

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



Leakage probability
p

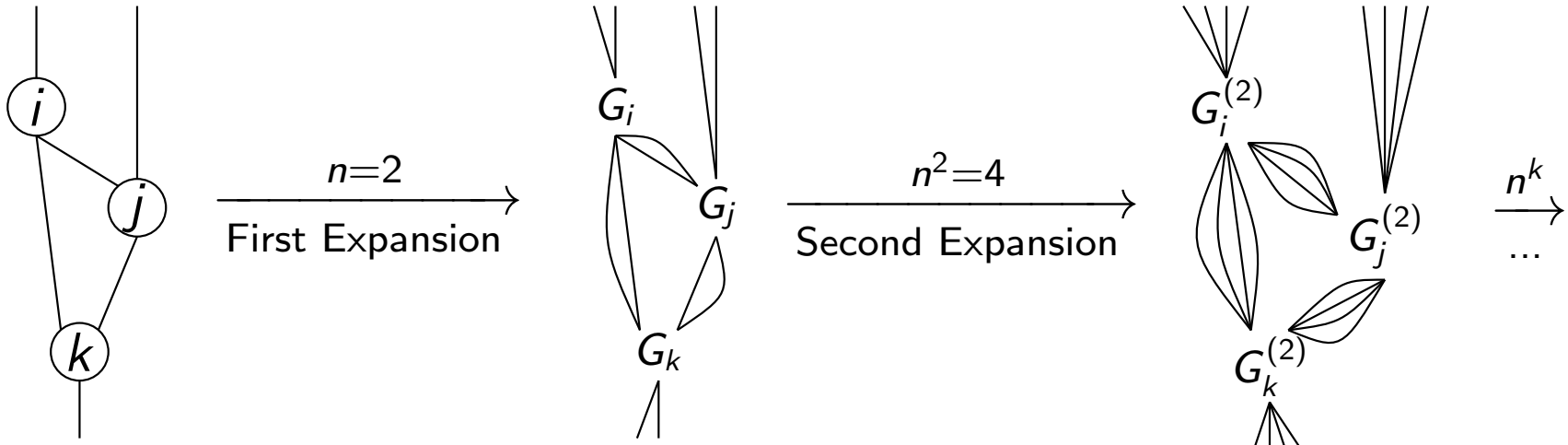
Simulation Failure
ε

ε^2

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



Leakage probability
p

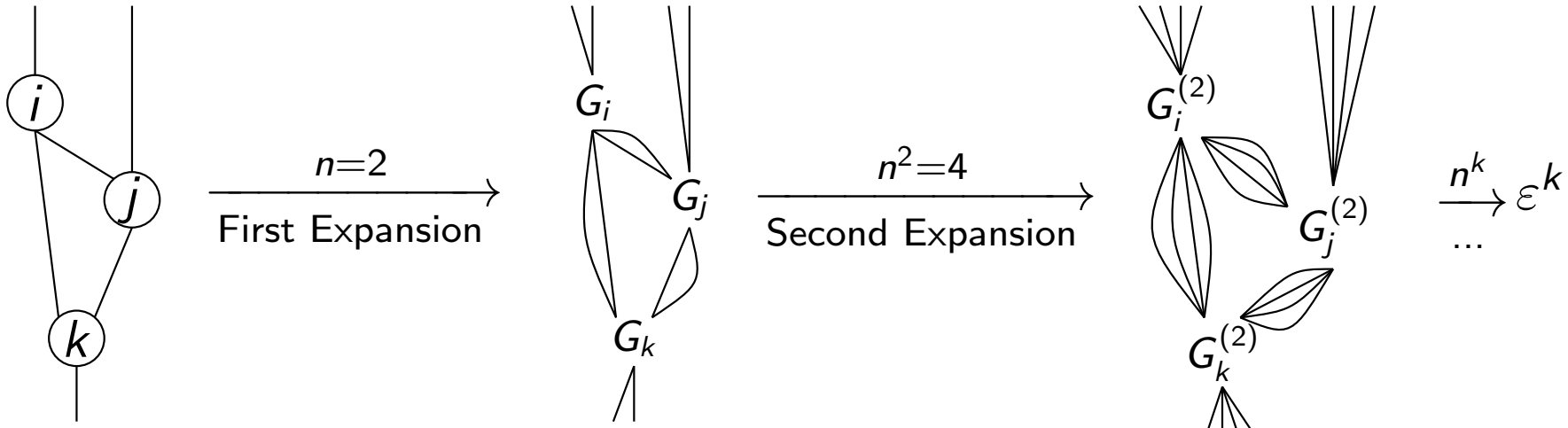
Simulation Failure
ε

ε^2

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



Leakage probability
p

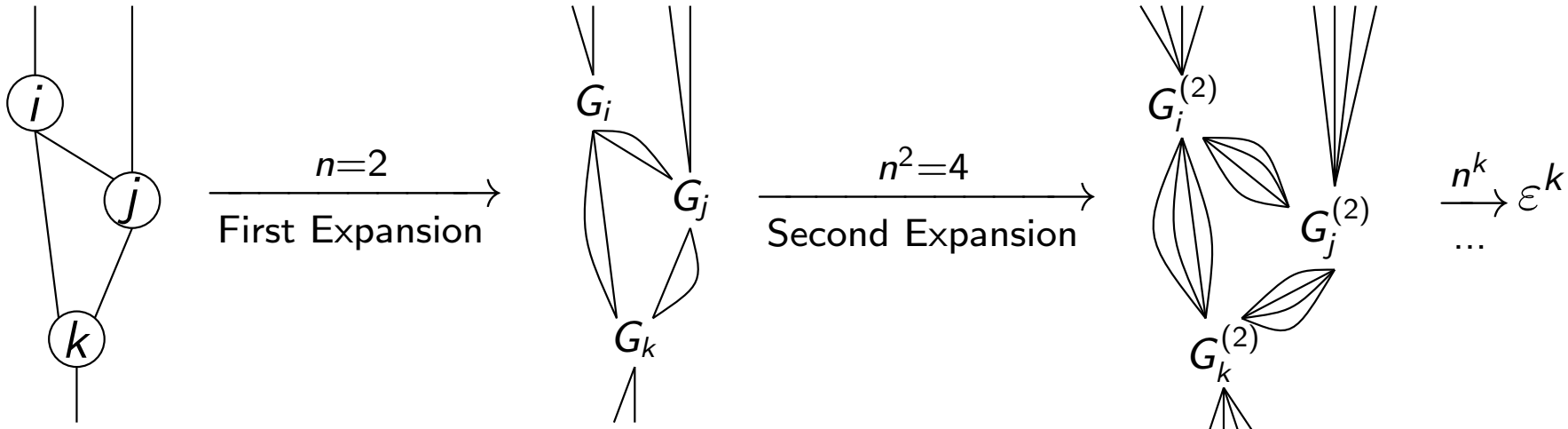
Simulation Failure
ε

ε^2

RP Expansion

Illustration

Using n-share gadgets $G_1, \ldots, G_\beta$



Leakage probability
p

Simulation Failure
ε

ε^2

Condition : $\varepsilon < p$ (tolerated leakage rate)

RP Expansion

Definition

$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

RP Expansion

Definition

$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

- (p, ε)-RP security of G (RPE $>>$ RP)

RP Expansion

Definition

$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

- (p, ε)-RP security of G (RPE $>>$ RP)
- $(\mathbf{t}, p, \varepsilon)$-RP composability of G (RPE $>>$ RPC)

RP Expansion

Definition

$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

- (p, ε)-RP security of G (RPE $>>$ RP)
- $(\mathbf{t}, p, \varepsilon)$-RP composability of G (RPE $>>$ RPC)
- Independent failure probability on each input sharing

RP Expansion

Definition

$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

- (p, ε)-RP security of G (RPE $>>$ RP)
- $(\mathbf{t}, p, \varepsilon)$-RP composability of G (RPE $>>$ RPC)
- Independent failure probability on each input sharing
- $G_1, \ldots, G_\beta$ are (t, p, ε)-RPE $\implies$ compiled circuit C is $(p, 2.|C|.\varepsilon^k)$-RP Secure

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$$

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+, ||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

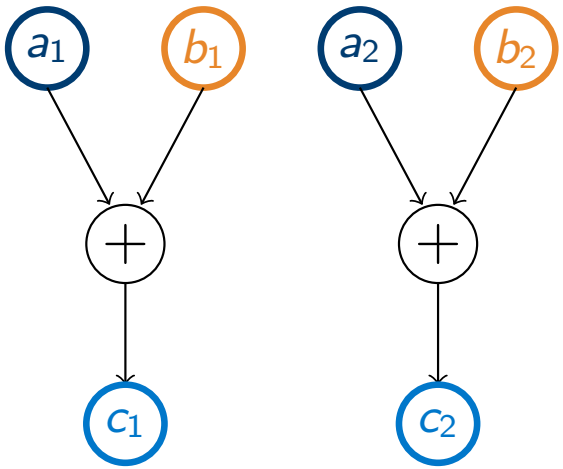
$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{log(\mathrm{N_{max}})}{log(\mathrm{d})}$$

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$$

d: amplification order (*i.e.* smallest failure set of internal wires)

Example $t = 1$, $n = 2$



RP Expansion

Parameters

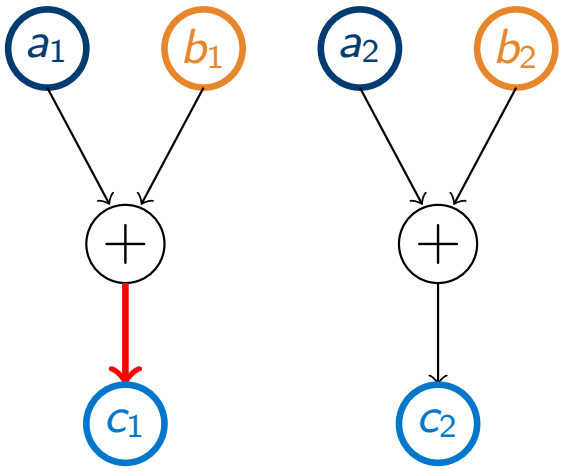
Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)



Example $t = 1$, $n = 2$

Output $c_1 = a_1 + b_1$,

RP Expansion

Parameters

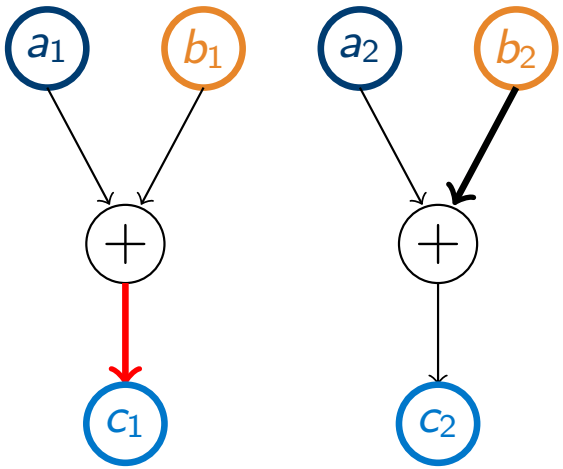
Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$$

d: amplification order (*i.e.* smallest failure set of internal wires)



Example $t = 1$, $n = 2$

Output $c_1 = a_1 + b_1$, **set** $\mathbf{W} = \{b_2\}$

RP Expansion

Parameters

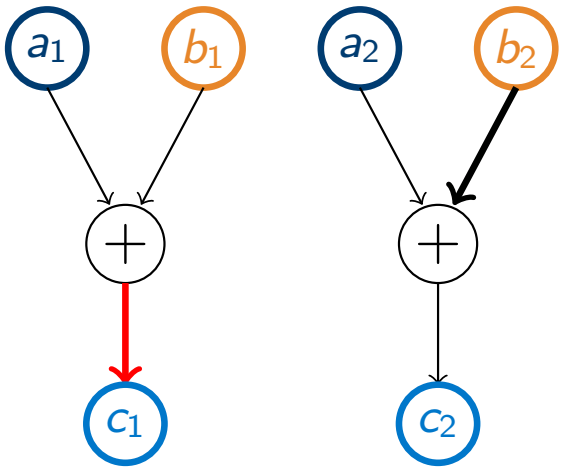
Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)



Example $t = 1$, $n = 2$

Output $c_1 = a_1 + b_1$, **set** $\mathbf{W} = \{b_2\}$

Simulation needs a_1 $(\leq t)$ and b_1, b_2 $(> t)$

RP Expansion

Parameters

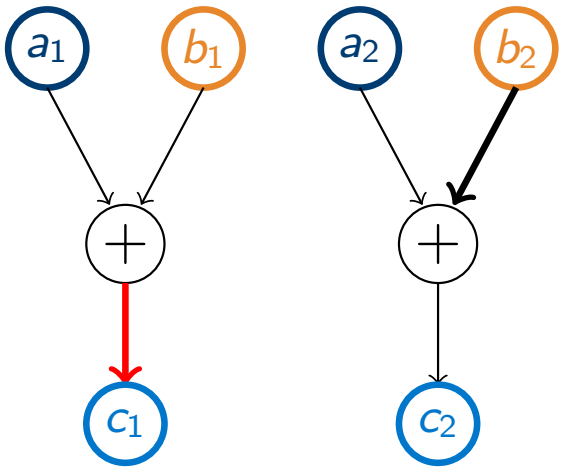
Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)



Example $t = 1$, $n = 2$

Output $c_1 = a_1 + b_1$, **set** $\mathbf{W} = \{b_2\}$

Simulation needs a_1 $(\leq t)$ and b_1, b_2 $(> t)$

Failure on b

RP Expansion

Parameters

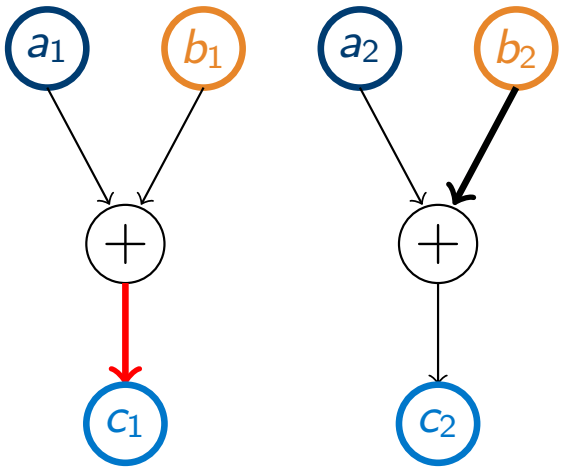
Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)



Example $t = 1$, $n = 2$

Output $c_1 = a_1 + b_1$, **set** $\mathbf{W} = \{b_2\}$

Simulation needs a_1 $(\leq t)$ and b_1, b_2 $(> t)$

Failure on $b \implies \mathbf{d} = |W| = \mathbf{1}$

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)

$$\varepsilon = f(p) = c_{\mathbf{d}} \cdot p^{\mathbf{d}} + \mathcal{O}(p^{\mathbf{d}+1})$$

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)

$$\varepsilon = f(p) = c_{\mathbf{d}} \cdot p^{\mathbf{d}} + \mathcal{O}(p^{\mathbf{d}+1})$$

- during expansion: $\varepsilon^k = f^{(k)}(p) = f(f(\ldots f(f(p))\ldots))$

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

Considering practical case with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$:

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}},\ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}})$

d: amplification order (*i.e.* smallest failure set of internal wires)

$$\varepsilon = f(p) = c_{\mathbf{d}} \cdot p^{\mathbf{d}} + \mathcal{O}(p^{\mathbf{d}+1})$$

- during expansion: $\varepsilon^k = f^{(k)}(p) = f(f(\ldots f(f(p))\ldots))$
- higher **d** $\implies$ faster decrease in failure probability ($d_{\max} = \frac{n+1}{2}$)

RP Expansion

Practical 3-share example

RP Expansion

Practical 3-share example

$$
\begin{aligned}
G_{copy} : v_0 &\leftarrow \mathbf{x_0} + r_0 + r_1; \;\; w_0 \leftarrow \mathbf{x_0} + r_3 + r_4 \\
v_1 &\leftarrow \mathbf{x_1} + r_1 + r_2; \;\; w_1 \leftarrow \mathbf{x_1} + r_4 + r_5 \\
v_2 &\leftarrow \mathbf{x_2} + r_2 + r_0; \;\; w_2 \leftarrow \mathbf{x_2} + r_5 + r_3
\end{aligned}
$$

RP Expansion

Practical 3-share example

$$
\begin{aligned}
G_{copy} : v_0 &\leftarrow \mathbf{x_0} + r_0 + r_1; \; w_0 \leftarrow \mathbf{x_0} + r_3 + r_4 \\
v_1 &\leftarrow \mathbf{x_1} + r_1 + r_2; \; w_1 \leftarrow \mathbf{x_1} + r_4 + r_5 \\
v_2 &\leftarrow \mathbf{x_2} + r_2 + r_0; \; w_2 \leftarrow \mathbf{x_2} + r_5 + r_3
\end{aligned}
$$

$$
\begin{aligned}
G_{add} : z_0 &\leftarrow \mathbf{x_0} + r_0 + r_4 + \mathbf{y_0} + r_1 + r_3 \\
z_1 &\leftarrow \mathbf{x_1} + r_1 + r_5 + \mathbf{y_1} + r_2 + r_4 \\
z_2 &\leftarrow \mathbf{x_2} + r_2 + r_3 + \mathbf{y_2} + r_0 + r_5
\end{aligned}
$$

RP Expansion

Practical 3-share example

$$G_{copy} : v_0 \leftarrow \mathbf{x_0} + r_0 + r_1;\ w_0 \leftarrow \mathbf{x_0} + r_3 + r_4$$
$$v_1 \leftarrow \mathbf{x_1} + r_1 + r_2;\ w_1 \leftarrow \mathbf{x_1} + r_4 + r_5$$
$$v_2 \leftarrow \mathbf{x_2} + r_2 + r_0;\ w_2 \leftarrow \mathbf{x_2} + r_5 + r_3$$

$$G_{add} : z_0 \leftarrow \mathbf{x_0} + r_0 + r_4 + \mathbf{y_0} + r_1 + r_3$$
$$z_1 \leftarrow \mathbf{x_1} + r_1 + r_5 + \mathbf{y_1} + r_2 + r_4$$
$$z_2 \leftarrow \mathbf{x_2} + r_2 + r_3 + \mathbf{y_2} + r_0 + r_5$$

$$G_{mult} : u_0 \leftarrow \mathbf{x_0} + r_5 + r_6; \quad u_1 \leftarrow \mathbf{x_1} + r_6 + r_7; \quad u_2 \leftarrow \mathbf{x_2} + r_7 + r_5$$
$$v_0 \leftarrow \mathbf{y_0} + r_8 + r_9; \quad v_1 \leftarrow \mathbf{y_1} + r_9 + r_{10}; \quad v_2 \leftarrow \mathbf{y_2} + r_{10} + r_8$$

RP Expansion

Practical 3-share example

$$G_{copy}: v_0 \leftarrow \mathbf{x_0} + r_0 + r_1;\ w_0 \leftarrow \mathbf{x_0} + r_3 + r_4$$
$$v_1 \leftarrow \mathbf{x_1} + r_1 + r_2;\ w_1 \leftarrow \mathbf{x_1} + r_4 + r_5$$
$$v_2 \leftarrow \mathbf{x_2} + r_2 + r_0;\ w_2 \leftarrow \mathbf{x_2} + r_5 + r_3$$

$$G_{add}: z_0 \leftarrow \mathbf{x_0} + r_0 + r_4 + \mathbf{y_0} + r_1 + r_3$$
$$z_1 \leftarrow \mathbf{x_1} + r_1 + r_5 + \mathbf{y_1} + r_2 + r_4$$
$$z_2 \leftarrow \mathbf{x_2} + r_2 + r_3 + \mathbf{y_2} + r_0 + r_5$$

$$G_{mult}: u_0 \leftarrow \mathbf{x_0} + r_5 + r_6;\quad u_1 \leftarrow \mathbf{x_1} + r_6 + r_7;\quad u_2 \leftarrow \mathbf{x_2} + r_7 + r_5$$
$$v_0 \leftarrow \mathbf{y_0} + r_8 + r_9;\quad v_1 \leftarrow \mathbf{y_1} + r_9 + r_{10};\quad v_2 \leftarrow \mathbf{y_2} + r_{10} + r_8$$

$$z_0 \leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2)$$
$$z_1 \leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3)$$
$$z_2 \leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4$$

RP Expansion

Practical 3-share example

$$
\begin{aligned}
G_{copy} : v_0 &\leftarrow \mathbf{x_0} + r_0 + r_1; \quad w_0 \leftarrow \mathbf{x_0} + r_3 + r_4 \\
v_1 &\leftarrow \mathbf{x_1} + r_1 + r_2; \quad w_1 \leftarrow \mathbf{x_1} + r_4 + r_5 \\
v_2 &\leftarrow \mathbf{x_2} + r_2 + r_0; \quad w_2 \leftarrow \mathbf{x_2} + r_5 + r_3
\end{aligned}
$$

$$
\begin{aligned}
G_{add} : z_0 &\leftarrow \mathbf{x_0} + r_0 + r_4 + \mathbf{y_0} + r_1 + r_3 \\
z_1 &\leftarrow \mathbf{x_1} + r_1 + r_5 + \mathbf{y_1} + r_2 + r_4 \\
z_2 &\leftarrow \mathbf{x_2} + r_2 + r_3 + \mathbf{y_2} + r_0 + r_5
\end{aligned}
$$

$$
\begin{aligned}
G_{mult} : u_0 &\leftarrow \mathbf{x_0} + r_5 + r_6; \quad u_1 \leftarrow \mathbf{x_1} + r_6 + r_7; \quad u_2 \leftarrow \mathbf{x_2} + r_7 + r_5 \\
v_0 &\leftarrow \mathbf{y_0} + r_8 + r_9; \quad v_1 \leftarrow \mathbf{y_1} + r_9 + r_{10}; \quad v_2 \leftarrow \mathbf{y_2} + r_{10} + r_8
\end{aligned}
$$

$$
\begin{aligned}
z_0 &\leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2) \\
z_1 &\leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3) \\
z_2 &\leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4
\end{aligned}
$$

$$
t = 1, \qquad f(p) \leq \sqrt{83}p^{3/2} + \mathcal{O}(p^2), \qquad \mathbf{N_{max}} = 21, \qquad p_{max} \approx 2^{-8}
$$

RP Expansion

Practical 3-share example

$$
\begin{aligned}
G_{copy} : v_0 &\leftarrow \mathbf{x_0} + r_0 + r_1; \quad w_0 \leftarrow \mathbf{x_0} + r_3 + r_4 \\
v_1 &\leftarrow \mathbf{x_1} + r_1 + r_2; \quad w_1 \leftarrow \mathbf{x_1} + r_4 + r_5 \\
v_2 &\leftarrow \mathbf{x_2} + r_2 + r_0; \quad w_2 \leftarrow \mathbf{x_2} + r_5 + r_3
\end{aligned}
$$

$$
\begin{aligned}
G_{add} : z_0 &\leftarrow \mathbf{x_0} + r_0 + r_4 + \mathbf{y_0} + r_1 + r_3 \\
z_1 &\leftarrow \mathbf{x_1} + r_1 + r_5 + \mathbf{y_1} + r_2 + r_4 \\
z_2 &\leftarrow \mathbf{x_2} + r_2 + r_3 + \mathbf{y_2} + r_0 + r_5
\end{aligned}
$$

$$
\begin{aligned}
G_{mult} : u_0 &\leftarrow \mathbf{x_0} + r_5 + r_6; \quad u_1 \leftarrow \mathbf{x_1} + r_6 + r_7; \quad u_2 \leftarrow \mathbf{x_2} + r_7 + r_5 \\
v_0 &\leftarrow \mathbf{y_0} + r_8 + r_9; \quad v_1 \leftarrow \mathbf{y_1} + r_9 + r_{10}; \quad v_2 \leftarrow \mathbf{y_2} + r_{10} + r_8
\end{aligned}
$$

$$
\begin{aligned}
z_0 &\leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2) \\
z_1 &\leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3) \\
z_2 &\leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4
\end{aligned}
$$

$$
t = 1, \qquad f(p) \leq \sqrt{83}p^{3/2} + \mathcal{O}(p^2), \qquad \mathbf{N_{max}} = 21, \qquad p_{max} \approx 2^{-8}
$$

$$
\mathcal{O}(|C|.\kappa^{7.5})
$$

RP Expansion

Practical 3-share example

RP Expansion

Practical 3-share example

Table 2: Complexity and execution time (in ms, on an Intel i7-8550U CPU) for compiled gadgets $G_{\text{add}}^{2(k)}$, $G_{\text{copy}}^{1(k)}$, $G_{\text{mult}}^{1(k)}$ from Section 6 implemented in C.

k	# shares	Gadget	Complexity $(N_{\text{a}}, N_{\text{c}}, N_{\text{m}}, N_{\text{r}})$	Execution time
1	3	$G_{\text{add}}^{2(1)}$	(15, 6, 0, 6)	$1,69.10^{-4}$
		$G_{\text{copy}}^{1(1)}$	(12, 9, 0, 6)	$1,67.10^{-4}$
		$G_{\text{mult}}^{1(1)}$	(28, 23, 9, 11)	$5,67.10^{-4}$
2	9	$G_{\text{add}}^{2(2)}$	(297, 144, 0, 144)	$2,21.10^{-3}$
		$G_{\text{copy}}^{1(2)}$	(288, 153, 0, 144)	$2,07.10^{-3}$
		$G_{\text{mult}}^{1(2)}$	(948, 582, 81, 438)	$9,91.10^{-3}$
3	27	$G_{\text{add}}^{2(3)}$	(6183, 3078, 0, 3078)	$9,29.10^{-2}$
		$G_{\text{copy}}^{1(3)}$	(6156, 3105, 0, 3078)	$9,84.10^{-2}$
		$G_{\text{mult}}^{1(3)}$	(23472, 12789, 729, 11385)	$3,67.10^{-1}$

RP Expansion

Practical 3-share example

Table 2: Complexity and execution time (in ms, on an Intel i7-8550U CPU) for compiled gadgets $G_{\text{add}}^{2(k)}$, $G_{\text{copy}}^{1(k)}$, $G_{\text{mult}}^{1(k)}$ from Section 6 implemented in C.

k	# shares	Gadget	Complexity $(N_{\text{a}}, N_{\text{c}}, N_{\text{m}}, N_{\text{r}})$	Execution time
1	3	$G_{\text{add}}^{2(1)}$	(15, 6, 0, 6)	$1,69.10^{-4}$
		$G_{\text{copy}}^{1(1)}$	(12, 9, 0, 6)	$1,67.10^{-4}$
		$G_{\text{mult}}^{1(1)}$	(28, 23, 9, 11)	$5,67.10^{-4}$
2	9	$G_{\text{add}}^{2(2)}$	(297, 144, 0, 144)	$2,21.10^{-3}$
		$G_{\text{copy}}^{1(2)}$	(288, 153, 0, 144)	$2,07.10^{-3}$
		$G_{\text{mult}}^{1(2)}$	(948, 582, 81, 438)	$9,91.10^{-3}$
3	27	$G_{\text{add}}^{2(3)}$	(6183, 3078, 0, 3078)	$9,29.10^{-2}$
		$G_{\text{copy}}^{1(3)}$	(6156, 3105, 0, 3078)	$9,84.10^{-2}$
		$G_{\text{mult}}^{1(3)}$	(23472, 12789, 729, 11385)	$3,67.10^{-1}$

Table 4: Standard and n-share AES-128 execution time (in ms, on an Intel i7-8550U CPU) using compiled gadgets $G_{\text{add}}^{2(k)}$, $G_{\text{copy}}^{1(k)}$, $G_{\text{mult}}^{1(k)}$.

AES Version	**Execution Time (in ms)**	
	Encryption	**Decryption**
Standard (no sharing)	0.06	0.05
3-share ($k = 1$)	1.08	1.07
9-share ($k = 2$)	11.71	10.26
27-share ($k = 3$)	200.29	197.70

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\mathsf{max}} = \dfrac{n+1}{2}$

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\mathsf{max}} = \dfrac{n+1}{2}$
- have good asymptotic complexity

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\max} = \dfrac{n+1}{2}$
- have good asymptotic complexity
- tolerate a good leakage rate $p_{\max}$

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\mathsf{max}} = \dfrac{n+1}{2}$
- have good asymptotic complexity
- tolerate a good leakage rate p_{max}

Goal: construct **generic** gadgets achieving d_{max}

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\mathsf{max}} = \dfrac{n+1}{2}$
- have good asymptotic complexity
- tolerate a good leakage rate p_{max}

Goal: construct **generic** gadgets achieving d_{max}

Idea:

RP Expansion

(t, p, ε)-RPE compiler with $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{mult}}$ should:

- achieve $d_{\mathsf{max}} = \dfrac{n+1}{2}$
- have good asymptotic complexity
- tolerate a good leakage rate p_{max}

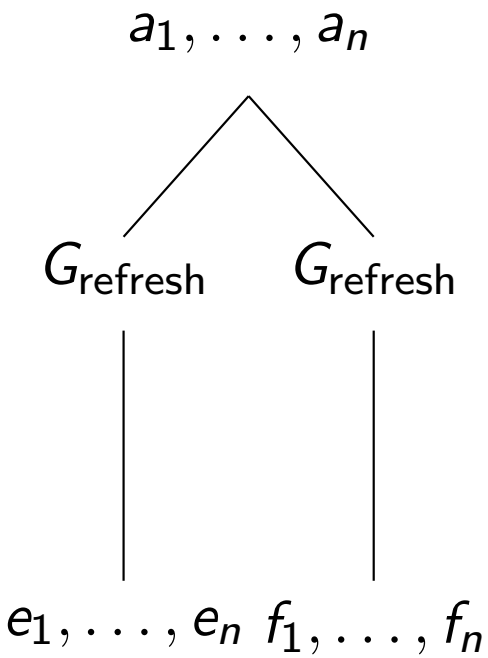
Goal: construct **generic** gadgets achieving d_{max}

Idea:

- build G_{add}, G_{copy} from single building block G_{refresh} (easier conception)
- use G_{mult} from state of the art (*e.g.* ISW, ...)

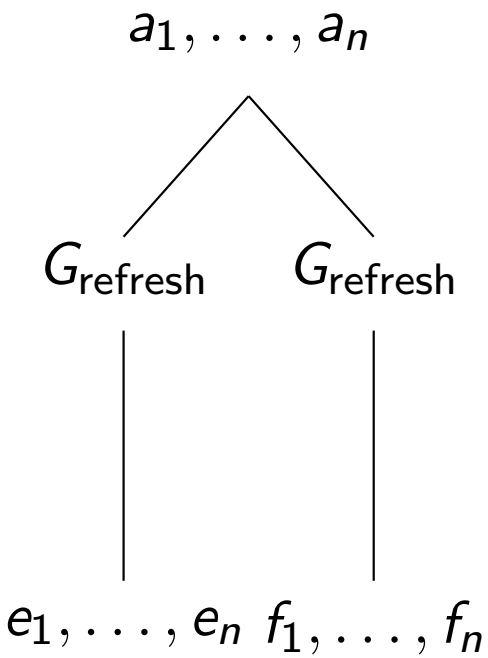
Generic Constructions

Copy Gadget



Generic Constructions

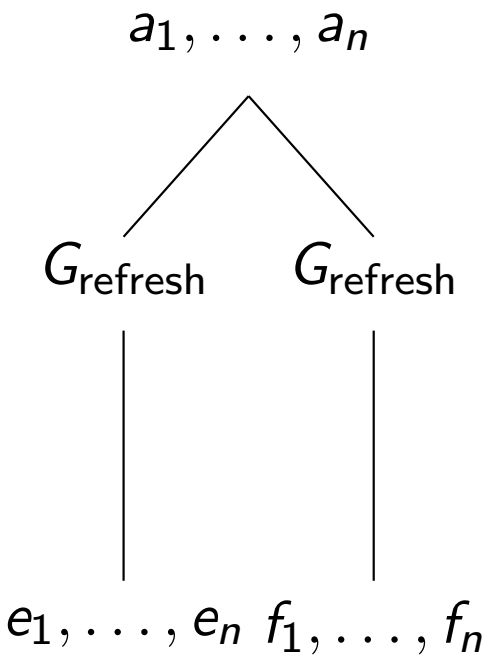
Copy Gadget



(t, p, ε)-RPE G_{refresh} achieves d_{max} $\Longrightarrow$
(t, p, ε')-RPE G_{copy} achieves d_{max}

Generic Constructions

Copy Gadget

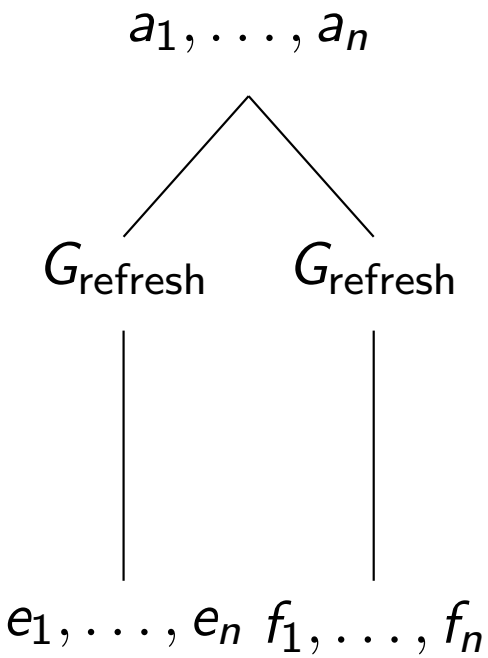


Idea:

(t, p, ε)-RPE G_{refresh} achieves d_{max} $\Longrightarrow$
(t, p, ε')-RPE G_{copy} achieves d_{max}

Generic Constructions

Copy Gadget



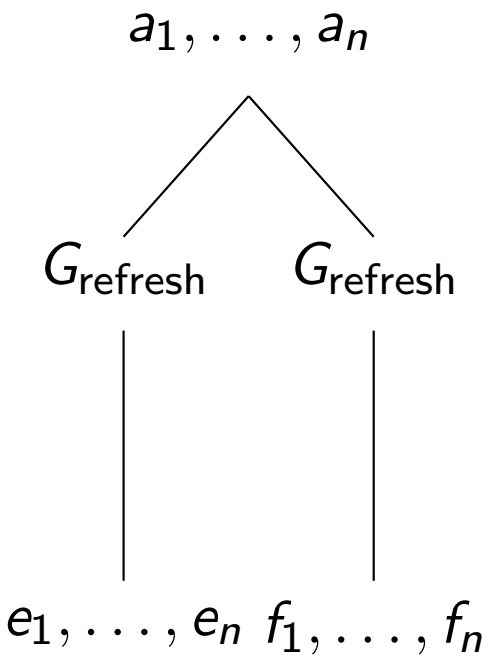
(t, p, ε)-RPE G_{refresh} achieves $d_{\max}$ $\Longrightarrow$
(t, p, ε')-RPE G_{copy} achieves $d_{\max}$

Idea:

- probes on G_{copy} = **left** probes on G_{refresh} $\cup$ **right** probes on G_{refresh} (independent)

Generic Constructions

Copy Gadget



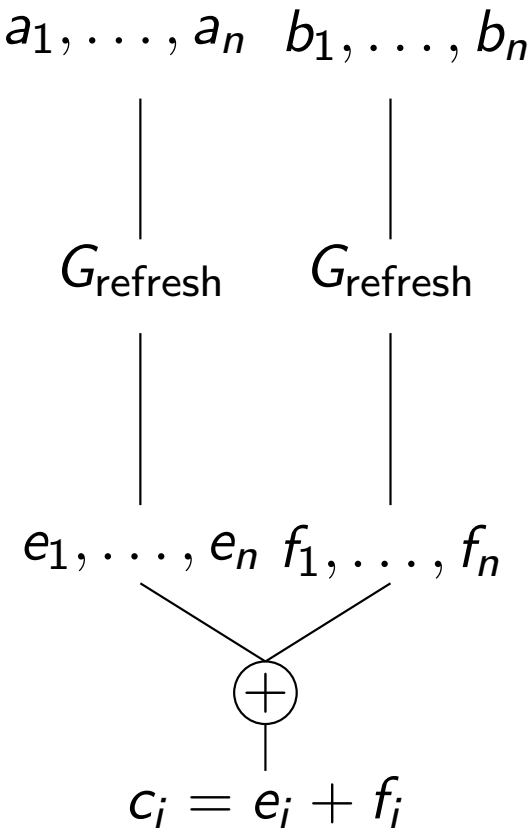
(t, p, ε)-RPE G_{refresh} achieves $d_{\text{max}} \implies$
(t, p, ε')-RPE G_{copy} achieves d_{max}

Idea:

- probes on G_{copy} = **left** probes on $G_{\text{refresh}} \cup$ **right** probes on G_{refresh} (independent)
- **left** simulation success + **right** simulation success $\implies$ overall simulation success

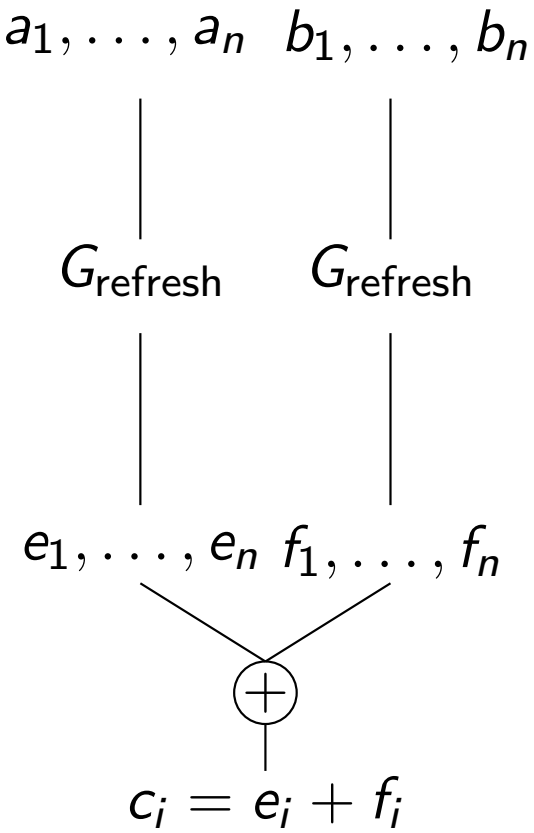
Generic Constructions

Addition Gadget



Generic Constructions

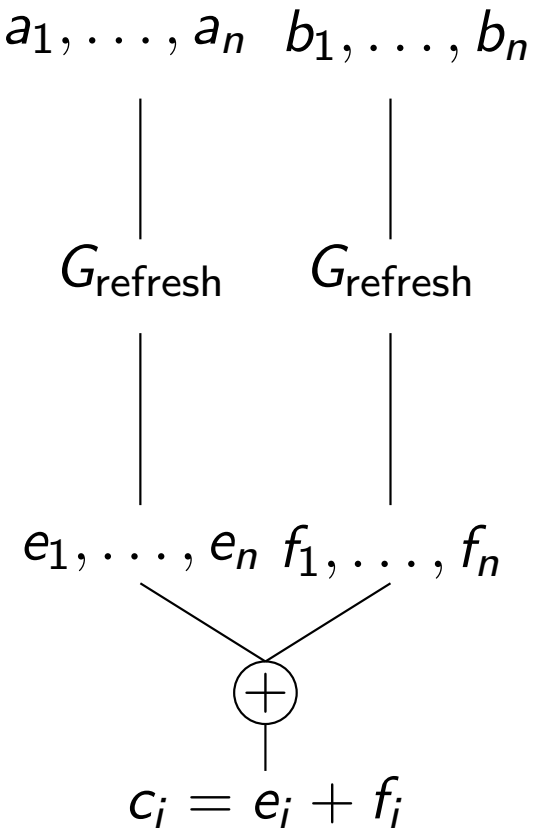
Addition Gadget



(t, p, ε)-RPE G_{refresh} achieves $d_{\max} \implies$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Generic Constructions

Addition Gadget

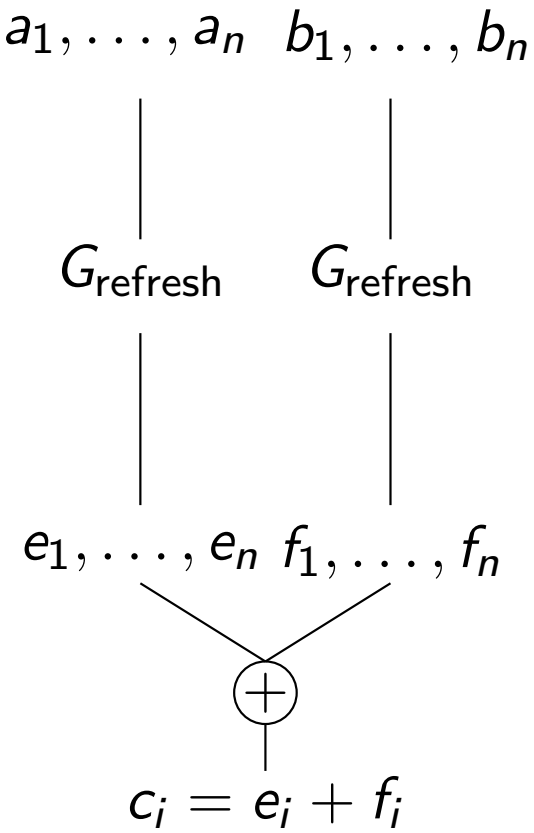


(t, p, ε)-RPE G_{refresh} achieves $d_{\max}$ $\Longrightarrow$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Idea:

Generic Constructions

Addition Gadget



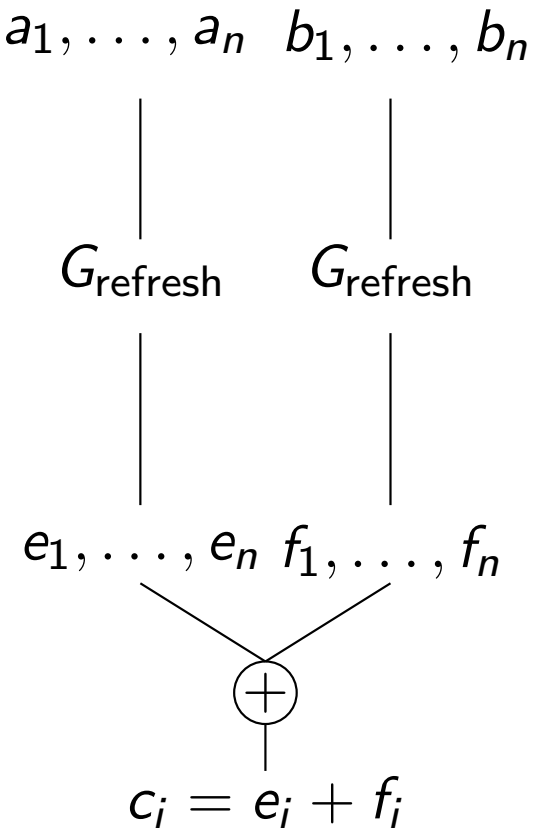
(t, p, ε)-RPE G_{refresh} achieves $d_{\max} \implies$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Idea:

- similar to the case of G_{copy} (left probes + right probes)

Generic Constructions

Addition Gadget



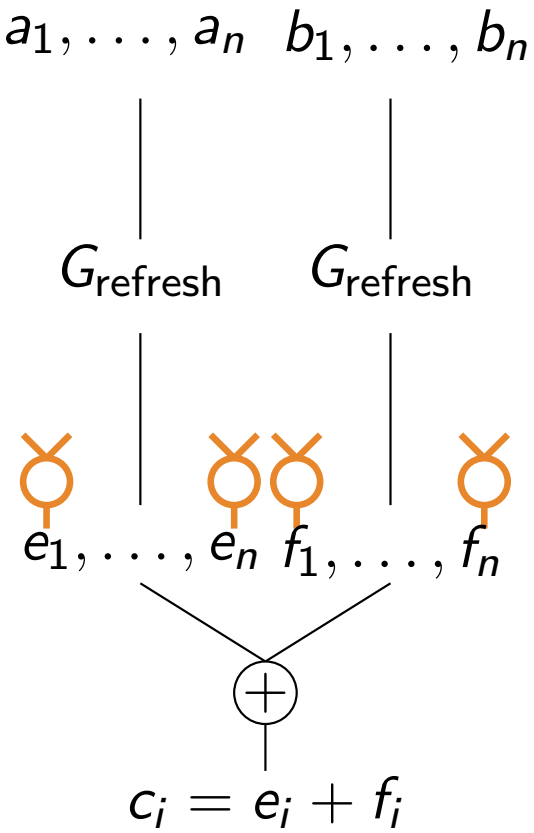
(t, p, ε)-RPE G_{refresh} achieves $d_{\max} \implies$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Idea:

- similar to the case of G_{copy} (left probes + right probes)
- **exception:** potential internal probes on $\{e_i\}_{i\in[n]}$, $\{f_i\}_{i\in[n]}$ (output shares of G_{refresh}s)

Generic Constructions

Addition Gadget



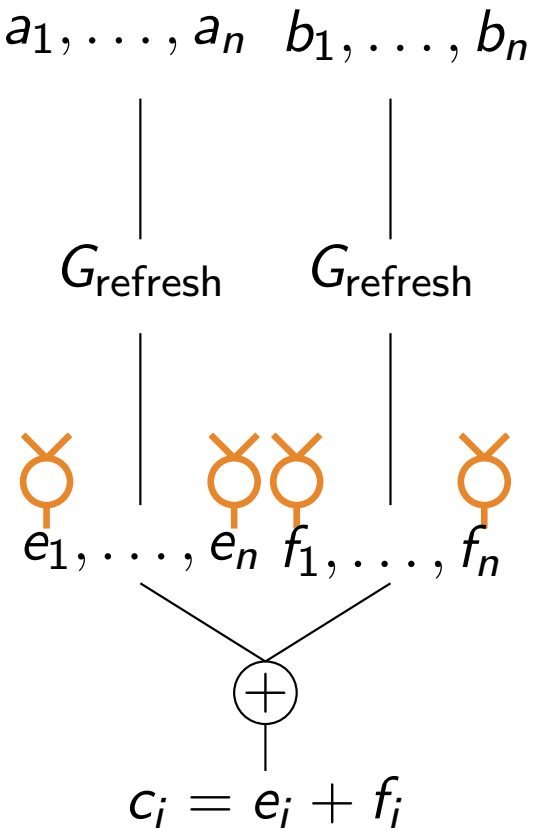
(t, p, ε)-RPE G_{refresh} achieves $d_{\max}$ $\Longrightarrow$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Idea:

- similar to the case of G_{copy} (left probes + right probes)
- **exception:** potential internal probes on $\{e_i\}_{i\in[n]}$, $\{f_i\}_{i\in[n]}$ (output shares of G_{refresh}s)
- **solution:** replace each by 2 wires input of corresponding output gate

Generic Constructions

Addition Gadget



(t, p, ε)-RPE G_{refresh} achieves $d_{\max}$ $\Longrightarrow$
(t, p, ε')-RPE G_{add} achieves **at least** $\frac{d_{\max}}{2}$

Idea:

- similar to the case of G_{copy} (left probes + right probes)
- **exception:** potential internal probes on $\{e_i\}_{i\in[n]}$, $\{f_i\}_{i\in[n]}$ (output shares of G_{refresh}s)
- **solution:** replace each by 2 wires input of corresponding output gate
- double number of probes $\Longrightarrow$ at least $d_{\max}/2$

Generic Constructions

ISW Gadget

Generic Constructions

ISW-based construction

Generic Constructions

ISW-based construction

ISW $G_{\mathsf{refresh}} \longrightarrow$ RPE achieving $d_{\mathsf{max}} = \dfrac{n+1}{2}$

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\text{max}} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\text{max}} = \dfrac{n+1}{2}$

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

ISW $G_{\text{mult}} \longrightarrow$ RPE achieving $\dfrac{d_{\max}}{2} = \dfrac{n+1}{4}$ **!!**

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

ISW $G_{\text{mult}} \longrightarrow$ RPE achieving $\dfrac{d_{\max}}{2} = \dfrac{n+1}{4}$ **!!**

Why?

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

ISW $G_{\text{mult}} \longrightarrow$ RPE achieving $\dfrac{d_{\max}}{2} = \dfrac{n+1}{4}$ **!!**

Why?

- direct product of input shares

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

ISW $G_{\text{mult}} \longrightarrow$ RPE achieving $\dfrac{d_{\max}}{2} = \dfrac{n+1}{4}$ **!!**

Why?

- direct product of input shares
- $W = \{a_1 \times b_1, \ldots, a_{t+1} \times b_{t+1}\}$ simulation needs $t+1$ shares of a **and** b

Generic Constructions

ISW-based construction

ISW $G_{\text{refresh}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{copy}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$

ISW $G_{\text{add}} \longrightarrow$ RPE achieving $d_{\max} = \dfrac{n+1}{2}$ (**Better than generic result**)

ISW $G_{\text{mult}} \longrightarrow$ RPE achieving $\dfrac{d_{\max}}{2} = \dfrac{n+1}{4}$ **!!**

Why?

- direct product of input shares
- $W = \{a_1 \times b_1, \ldots, a_{t+1} \times b_{t+1}\}$ simulation needs $t+1$ shares of a **and** b
- Failure on both inputs with independent failure events

Inputs a,b (illustration with 3 shares)

$$\begin{pmatrix} & & \end{pmatrix} + \begin{pmatrix} & & \end{pmatrix} = \begin{pmatrix} & & \end{pmatrix}$$

Inputs a,b (illustration with 3 shares)

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix} + \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

Inputs a,b (illustration with 3 shares)

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix} + \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

Inputs a,b (illustration with 3 shares)

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix} + \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

Inputs a,b (illustration with 3 shares)

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix} + \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

$$\mathbf{v} = \begin{pmatrix} p_{1,1} + p_{1,2} + p_{1,3} \\ p_{2,1} + p_{2,2} + p_{2,3} \\ p_{3,1} + p_{3,2} + p_{3,3} \end{pmatrix}$$

Inputs a, b (illustration with 3 shares)

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix} + \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} p_{1,1} + p_{1,2} + p_{1,3} \\ p_{2,1} + p_{2,2} + p_{2,3} \\ p_{3,1} + p_{3,2} + p_{3,3} \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} r_1 + r_4 + r_7 \\ r_2 + r_5 + r_8 \\ r_3 + r_6 + r_9 \end{pmatrix}$$

Output $C = V + X$

Non-standard G_{mult}

Non-standard G_{mult}

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix}$$

where b, b, b are three independent fresh copies of b using G_{refresh}

Non-standard G_{mult}

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix}$$

where b, b, b are three independent fresh copies of b using G_{refresh}

- G_{refresh} (t, p, ε)-RPE achieves $d_{\text{max}} \implies G_{\text{mult}}$ (t, p, ε)-RPE achieves d_{max}

Non-standard G_{mult}

$$\begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{pmatrix}$$

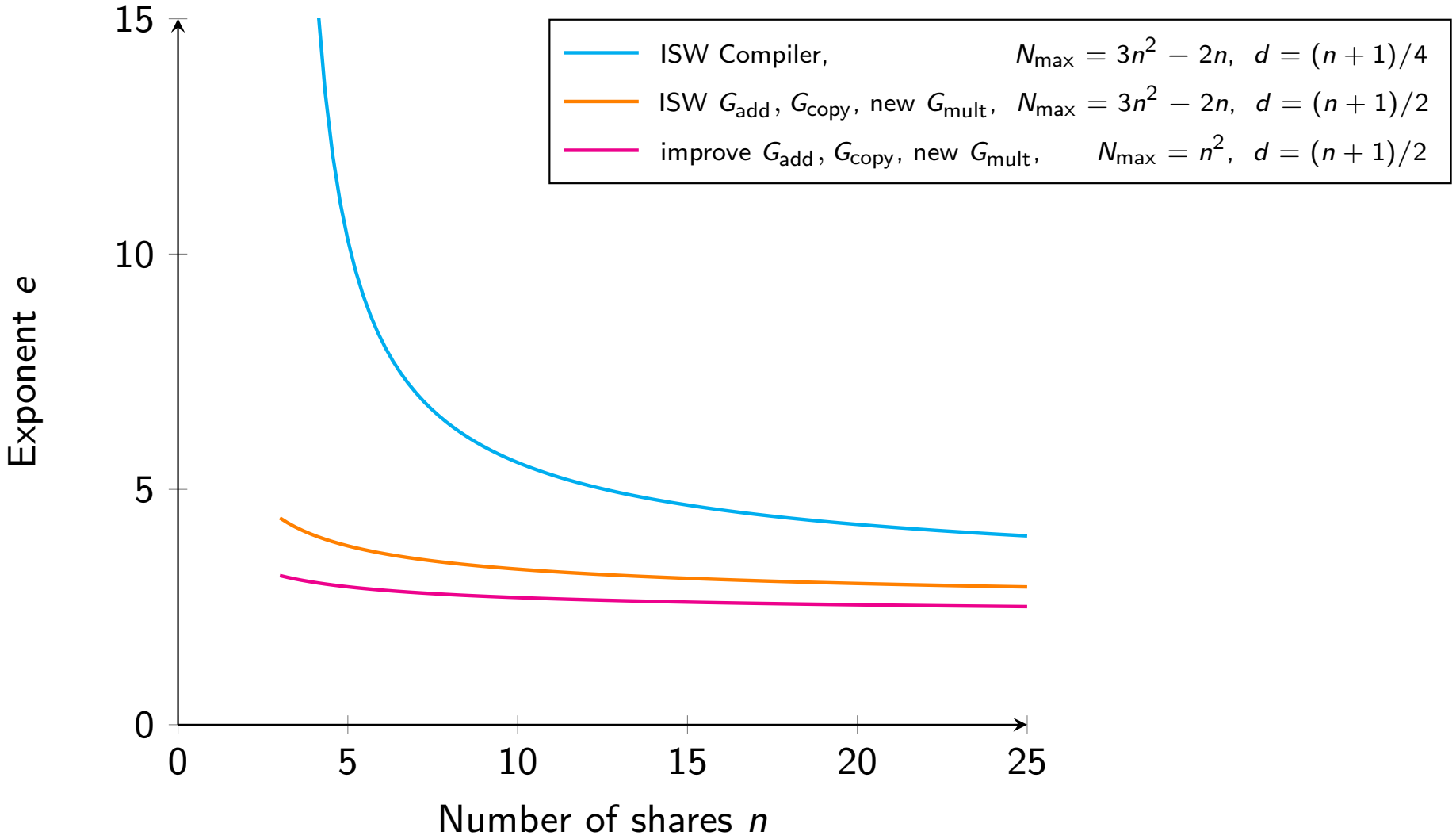
where b, b, b are three independent fresh copies of b using G_{refresh}

- G_{refresh} (t, p, ε)-RPE achieves d_{max} $\implies$ G_{mult} (t, p, ε)-RPE achieves d_{max}
- New G_{mult} achieves $d_{\text{max}} = \dfrac{n+1}{2}$ (unlike ISW mult.)

Generic Constructions

Asymptotic Complexity

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



Concrete RPE instantiations

Concrete RPE instantiations

3-share

G_{ref} : $c_1 \leftarrow a_1 + r_1$

$c_2 \leftarrow a_2 + r_2$

$c_3 \leftarrow (r_1 + r_2) + a_3$

Concrete RPE instantiations

3-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow a_1 + r_1 \\
c_2 &\leftarrow a_2 + r_2 \\
c_3 &\leftarrow (r_1 + r_2) + a_3
\end{aligned}
$$

5-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow (r_1 + r_2) + a_1 \\
c_2 &\leftarrow (r_2 + r_3) + a_2 \\
c_3 &\leftarrow (r_3 + r_4) + a_3 \\
c_4 &\leftarrow (r_4 + r_5) + a_4 \\
c_5 &\leftarrow (r_5 + r_1) + a_5
\end{aligned}
$$

Concrete RPE instantiations

3-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow a_1 + r_1 \\
c_2 &\leftarrow a_2 + r_2 \\
c_3 &\leftarrow (r_1 + r_2) + a_3
\end{aligned}
$$

5-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow (r_1 + r_2) + a_1 \\
c_2 &\leftarrow (r_2 + r_3) + a_2 \\
c_3 &\leftarrow (r_3 + r_4) + a_3 \\
c_4 &\leftarrow (r_4 + r_5) + a_4 \\
c_5 &\leftarrow (r_5 + r_1) + a_5
\end{aligned}
$$

- G_{copy}, G_{add} based on G_{refresh}

Concrete RPE instantiations

3-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow a_1 + r_1 \\
c_2 &\leftarrow a_2 + r_2 \\
c_3 &\leftarrow (r_1 + r_2) + a_3
\end{aligned}
$$

5-share

$$
\begin{aligned}
G_{ref} : c_1 &\leftarrow (r_1 + r_2) + a_1 \\
c_2 &\leftarrow (r_2 + r_3) + a_2 \\
c_3 &\leftarrow (r_3 + r_4) + a_3 \\
c_4 &\leftarrow (r_4 + r_5) + a_4 \\
c_5 &\leftarrow (r_5 + r_1) + a_5
\end{aligned}
$$

- G_{copy}, G_{add} based on G_{refresh}
- G_{mult} from new generic construction using G_{refresh}

Concrete RPE instantiations

3-share

$$G_{ref} : \begin{aligned} c_1 &\leftarrow a_1 + r_1 \\ c_2 &\leftarrow a_2 + r_2 \\ c_3 &\leftarrow (r_1 + r_2) + a_3 \end{aligned}$$

5-share

$$G_{ref} : \begin{aligned} c_1 &\leftarrow (r_1 + r_2) + a_1 \\ c_2 &\leftarrow (r_2 + r_3) + a_2 \\ c_3 &\leftarrow (r_3 + r_4) + a_3 \\ c_4 &\leftarrow (r_4 + r_5) + a_4 \\ c_5 &\leftarrow (r_5 + r_1) + a_5 \end{aligned}$$

- G_{copy}, G_{add} based on G_{refresh}
- G_{mult} from new generic construction using G_{refresh}

Construction	$d_{\max}$	Complexity	Tolerated Leakage rate
[C20] 3-share	$3/2$	$\mathcal{O}(\lvert C\rvert \cdot \kappa^{7.5})$	$p = 2^{-8}$
[EC21] 3-share	2	$\mathcal{O}(\lvert C\rvert \cdot \kappa^{3.9})$	$p = 2^{-7.5}$
[EC21] 5-share	3	$\mathcal{O}(\lvert C\rvert \cdot \kappa^{3.2})$	$2^{-12} \leq p \leq 2^{-6}$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1}$$

Leakage rate p

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1$$

Leakage rate p

$n_1^{k_1}$ shares

$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2}$$

Leakage rate p

$n_1^{k_1}$ shares

$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2 \xrightarrow[\ldots]{\ldots} \ldots$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2 \xrightarrow[\cdots]{\cdots} \cdots \xrightarrow[k_\ell \text{ times}]{CC_\ell}$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2 \xrightarrow[\cdots]{\cdots} \cdots \xrightarrow[k_\ell \text{ times}]{CC_\ell} \hat{C}_\ell$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares	$n_\ell^{k_\ell} \cdots n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$	$\varepsilon_\ell^{k_\ell} = f_\ell^{(k_\ell)}(\ldots(f_1^{(k_1)}(p)\ldots)$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2 \xrightarrow[\cdots]{\cdots} \cdots \xrightarrow[k_\ell \text{ times}]{CC_\ell} \hat{C}_\ell$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares	$n_\ell^{k_\ell} \cdots n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$	$\varepsilon_\ell^{k_\ell} = f_\ell^{(k_\ell)}(\ldots(f_1^{(k_1)}(p)\ldots)$

Conditions: $\varepsilon_1 < p, \quad \varepsilon_2 < \varepsilon_1^{k_1}, \quad \ldots \quad , \quad \varepsilon_\ell < \varepsilon_{\ell-1}^{k_{\ell-1}}$

Dynamic RP Expansion

Idea

Using RPE compilers $CC_1, \ldots, CC_\ell$ with numbers of shares $n_1, \ldots, n_\ell$

$$C \xrightarrow[k_1 \text{ times}]{CC_1} \hat{C}_1 \xrightarrow[k_2 \text{ times}]{CC_2} \hat{C}_2 \xrightarrow[\cdots]{\cdots} \cdots \xrightarrow[k_\ell \text{ times}]{CC_\ell} \hat{C}_\ell$$

	$n_1^{k_1}$ shares	$n_2^{k_2} \cdot n_1^{k_1}$ shares	$n_\ell^{k_\ell} \cdots n_1^{k_1}$ shares
Leakage rate p	$\varepsilon_1^{k_1} = f_1^{(k_1)}(p)$	$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$	$\varepsilon_\ell^{k_\ell} = f_\ell^{(k_\ell)}(\ldots(f_1^{(k_1)}(p)\ldots)$

Conditions: $\varepsilon_1 < p, \quad \varepsilon_2 < \varepsilon_1^{k_1}, \quad \ldots \quad , \quad \varepsilon_\ell < \varepsilon_{\ell-1}^{k_{\ell-1}}$

Why?

Dynamic RP Expansion

Motivation

n-share RPE compilers:

Dynamic RP Expansion

Motivation

n-share RPE compilers:

- **small** n: fewer sets of probes that reveal the secret $\Longrightarrow$ tolerate better leakage rate p

Dynamic RP Expansion

Motivation

n-share RPE compilers:

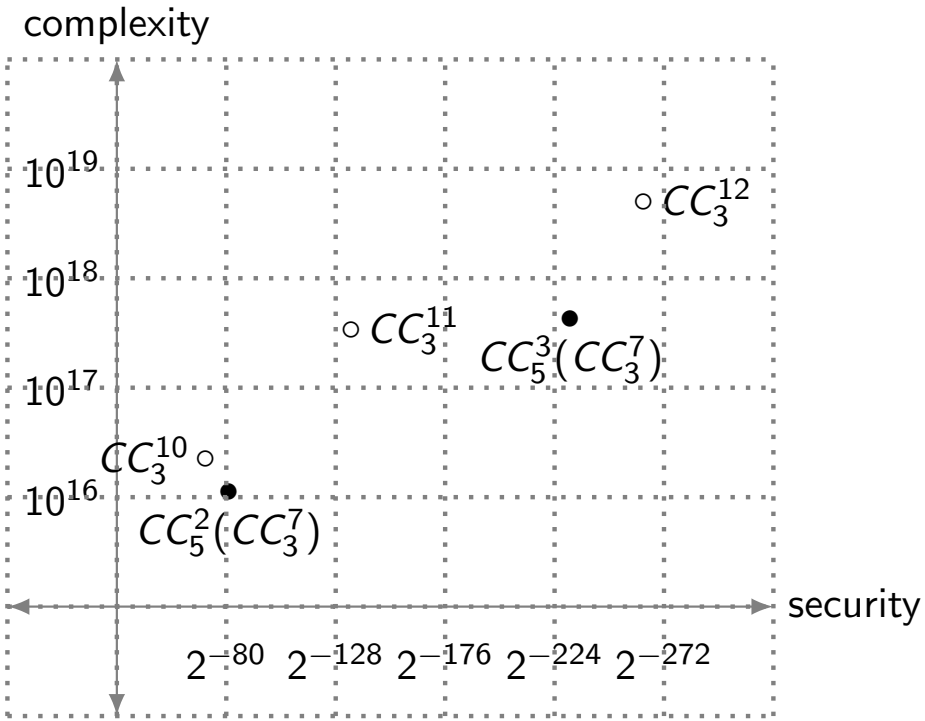
- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \dfrac{n+1}{2}$ $\implies$ have better asymptotic complexity

Dynamic RP Expansion

Motivation

n-share RPE compilers:

- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \frac{n+1}{2}$ $\implies$ have better asymptotic complexity



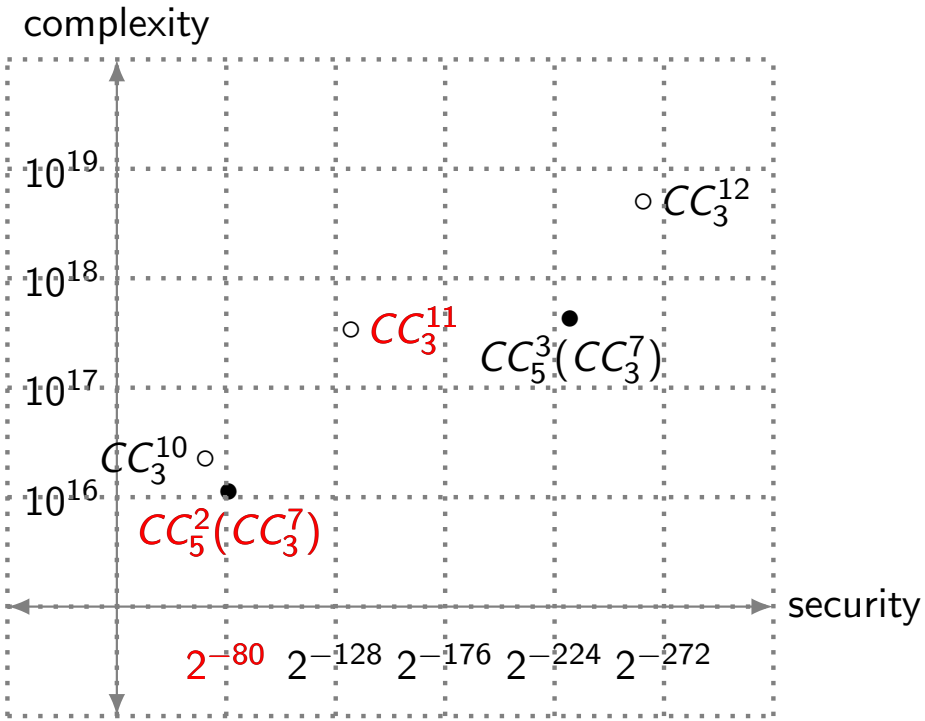
Complexity and security level of RP AES starting from tolerated leakage of $p = 2^{-7.6}$ using 3-share CC_3 and 5-share CC_5 from EC21

Dynamic RP Expansion

Motivation

n-share RPE compilers:

- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \dfrac{n+1}{2}$ $\implies$ have better asymptotic complexity



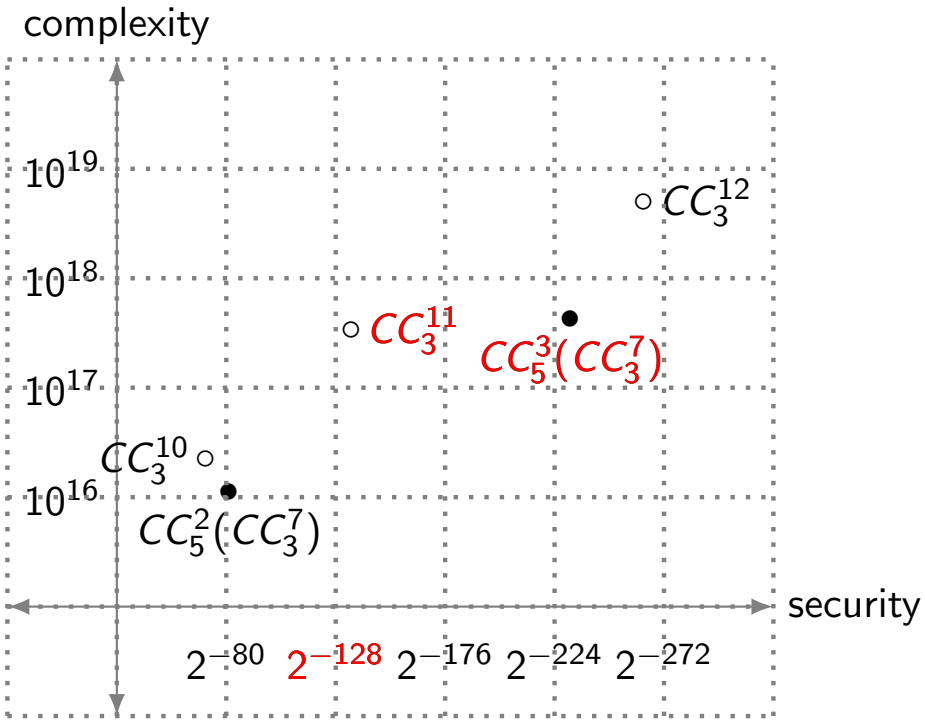
Complexity and security level of RP AES starting from tolerated leakage of $p = 2^{-7.6}$ using 3-share CC_3 and 5-share CC_5 from EC21

Dynamic RP Expansion

Motivation

n-share RPE compilers:

- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \dfrac{n+1}{2}$ $\implies$ have better asymptotic complexity



Complexity and security level of RP AES starting from tolerated leakage of $p = 2^{-7.6}$ using 3-share CC_3 and 5-share CC_5 from EC21

Dynamic RP Expansion

Motivation

2 possible directions:

Dynamic RP Expansion

Motivation

2 possible directions:

- look for gadgets with **small** number of shares tolerating the best leakage rate (eventually with high complexity)

Dynamic RP Expansion

Motivation

2 possible directions:

- look for gadgets with **small** number of shares tolerating the best leakage rate (eventually with high complexity)

- look for gadgets which achieve maximal amp. order for **any** number shares with low asymptotic complexity

Linear Gadgets

Building Block

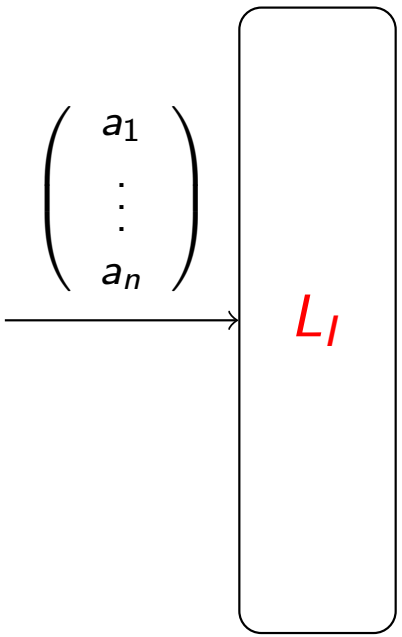
$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:

$$\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \longrightarrow$$

Linear Gadgets

Building Block

$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



$n/2$ randoms

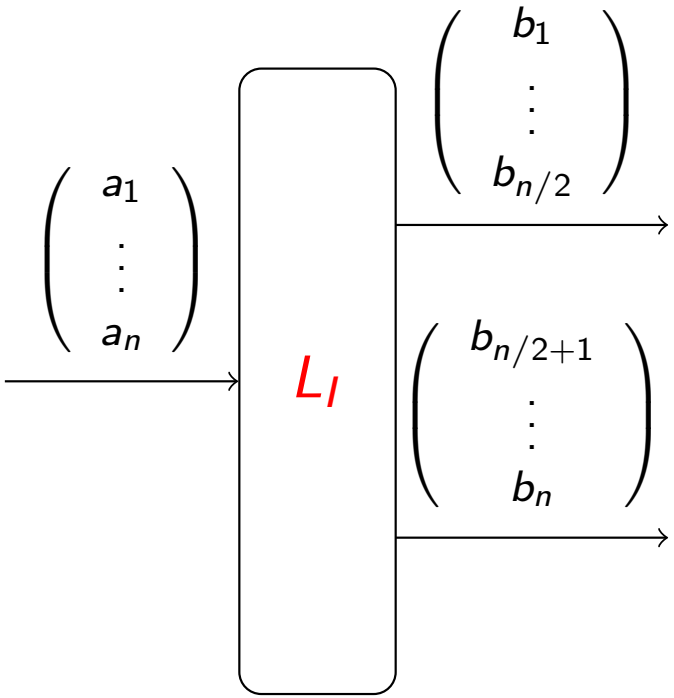
$b_i \leftarrow a_i + r_i$

$b_{n/2+i} \leftarrow a_{n/2+i} + r_i$

Linear Gadgets

Building Block

$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



$n/2$ randoms

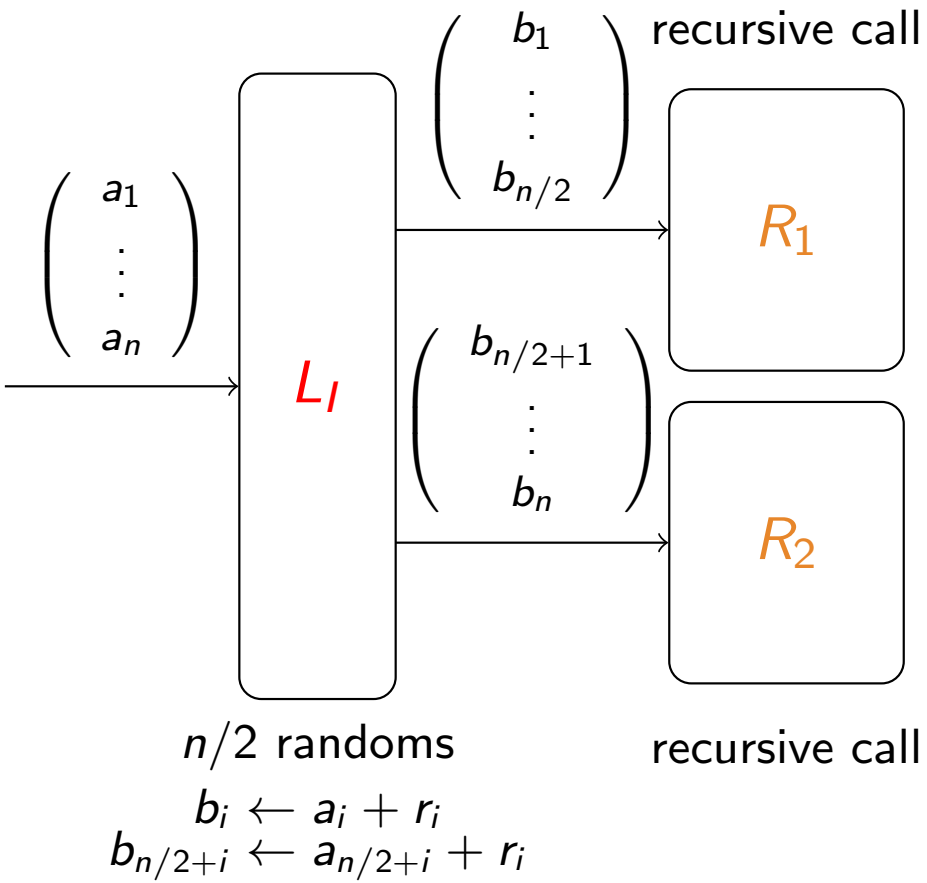
$b_i \leftarrow a_i + r_i$

$b_{n/2+i} \leftarrow a_{n/2+i} + r_i$

Linear Gadgets

Building Block

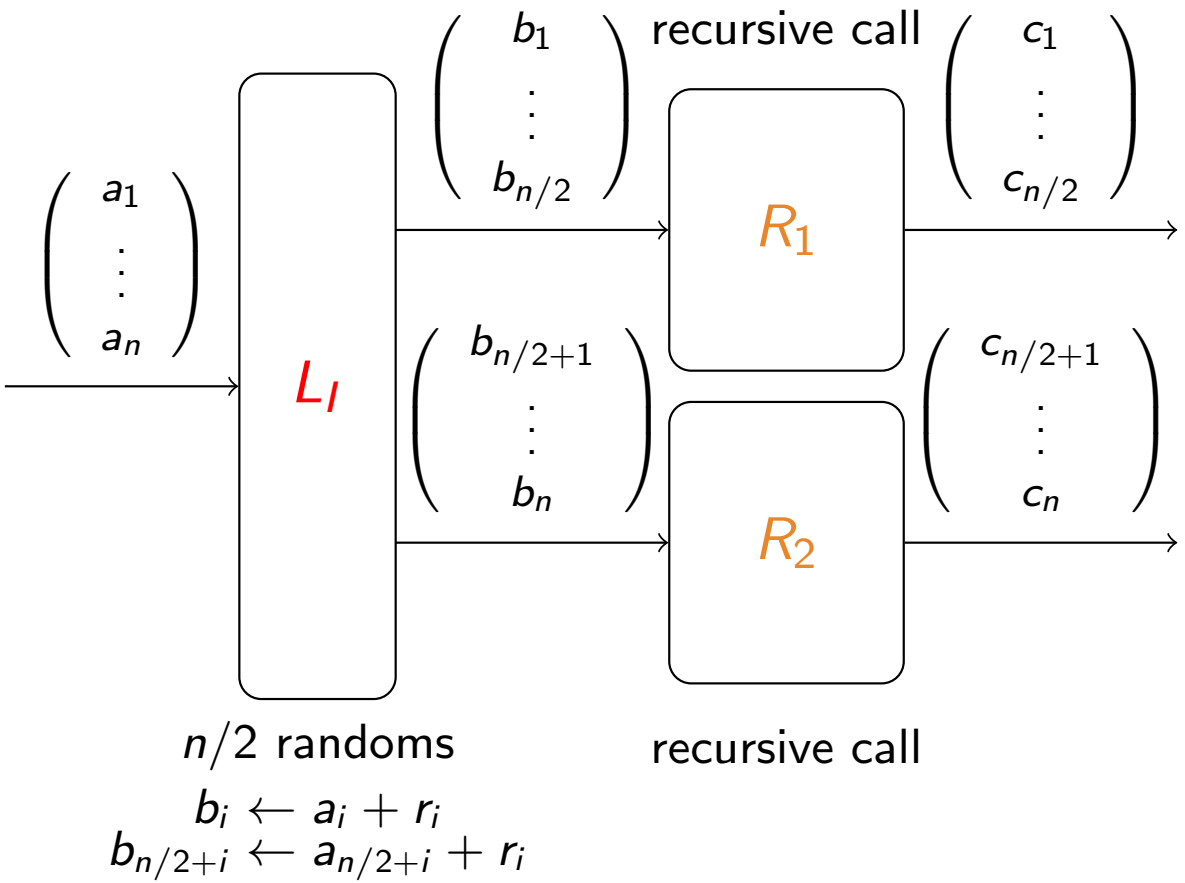
$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



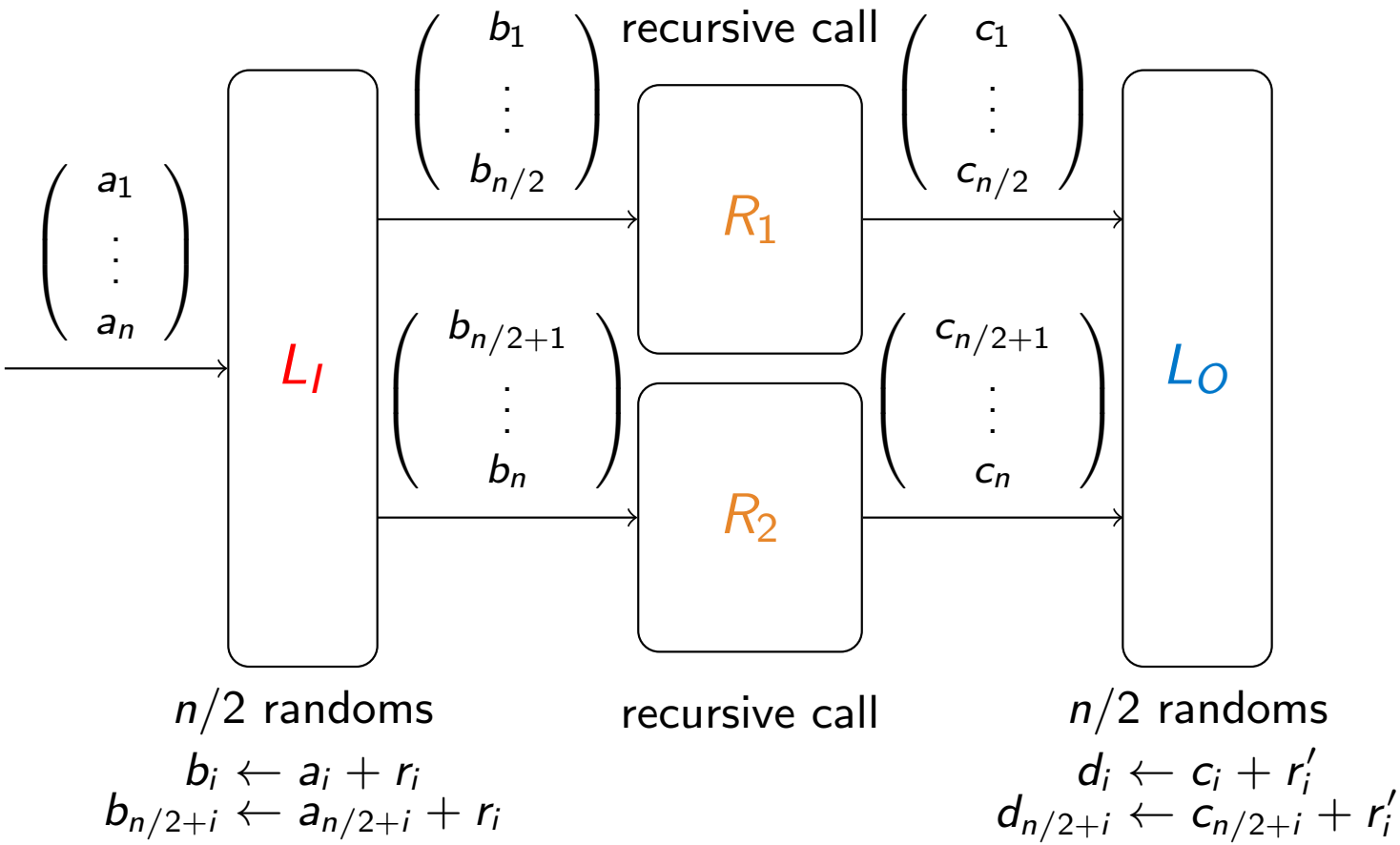
Linear Gadgets

Building Block

$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



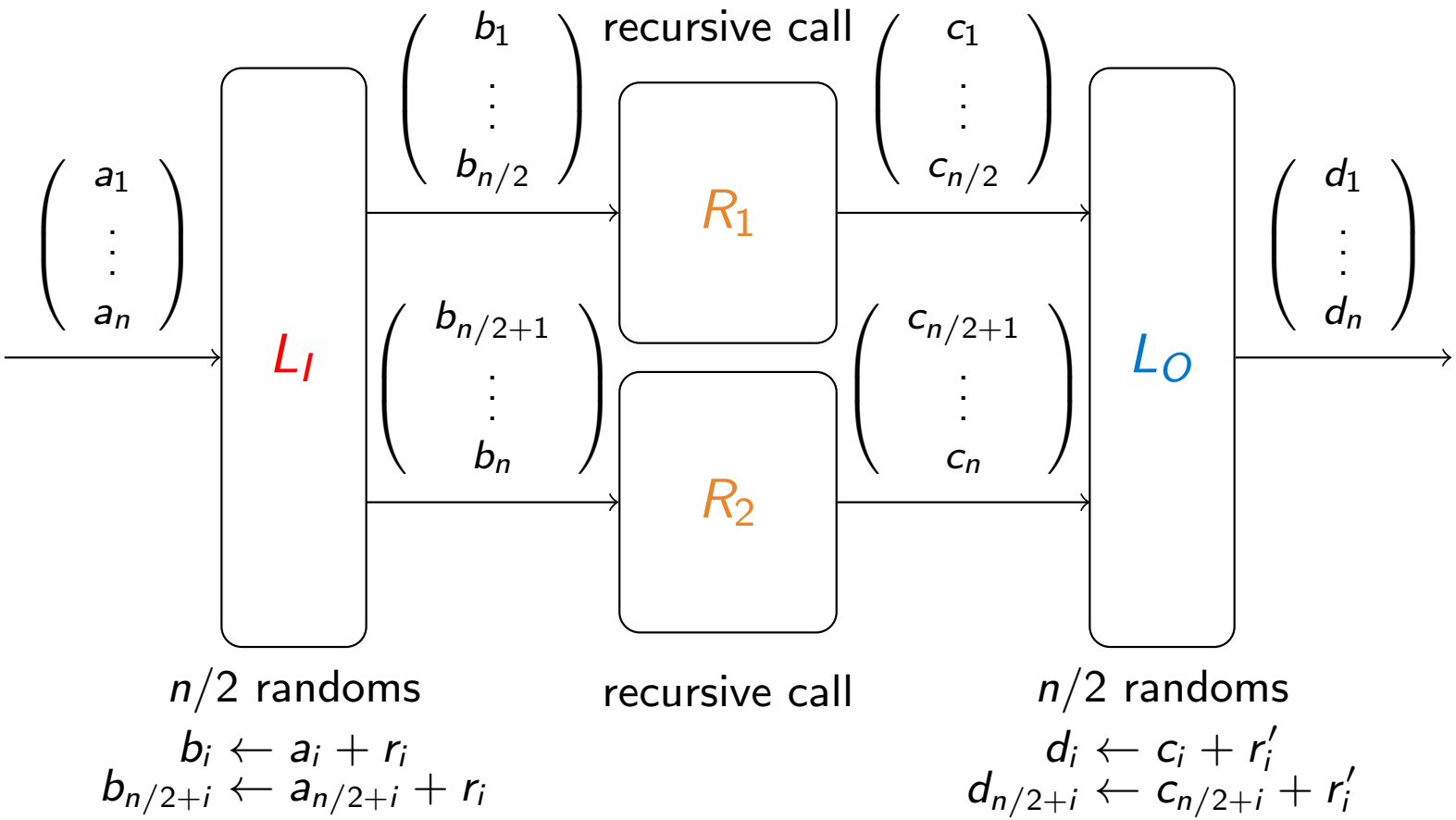
$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



Linear Gadgets

Building Block

$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



Linear Gadgets

Building Block

Example (4 shares):

$$
\begin{aligned}
d_1 &\leftarrow (a_1 + r_1) + r_3 + r_5 \\
d_2 &\leftarrow (a_2 + r_2) + r_3 + r_6 \\
d_3 &\leftarrow (a_3 + r_1) + r_4 + r_5 \\
d_4 &\leftarrow (a_4 + r_2) + r_4 + r_6
\end{aligned}
$$

Linear Gadgets

Building Block

Example (4 shares):

$$
\begin{aligned}
d_1 &\leftarrow (a_1 + r_1) + r_3 + r_5 \\
d_2 &\leftarrow (a_2 + r_2) + r_3 + r_6 \\
d_3 &\leftarrow (a_3 + r_1) + r_4 + r_5 \\
d_4 &\leftarrow (a_4 + r_2) + r_4 + r_6
\end{aligned}
$$

- proven by *Battistello et al.* to be $(n-1)$-SNI in the probing model

Linear Gadgets

Building Block

Example (4 shares):

$$
\begin{aligned}
d_1 &\leftarrow (a_1 + r_1) + r_3 + r_5 \\
d_2 &\leftarrow (a_2 + r_2) + r_3 + r_6 \\
d_3 &\leftarrow (a_3 + r_1) + r_4 + r_5 \\
d_4 &\leftarrow (a_4 + r_2) + r_4 + r_6
\end{aligned}
$$

- proven by *Battistello et al.* to be $(n-1)$-SNI in the probing model
- proven in **our work** to satisfy stronger requirements to be used as a building block for RPE secure constructions (extension of requirements proposed by *Belaïd et al. - EuroCrypt 2021*)

Linear Gadgets

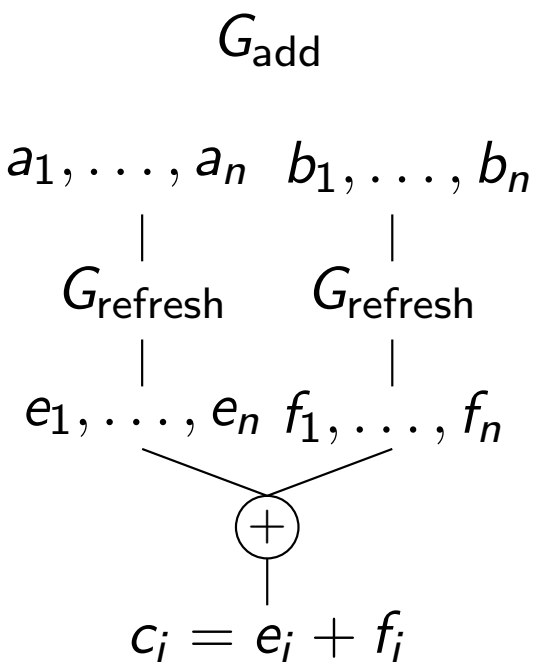
Constructions

Using $\mathcal{O}(n \log n)$ G_{refresh}

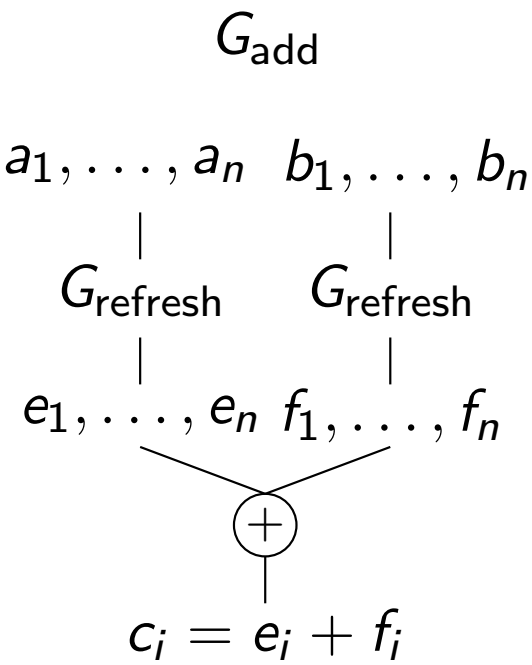
Linear Gadgets

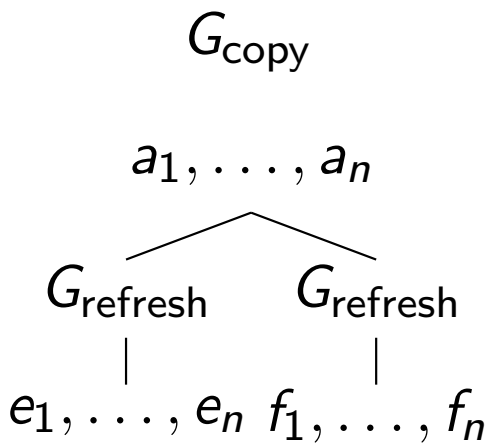
Constructions

Using $\mathcal{O}(n \log n)$ G_{refresh}



Using $\mathcal{O}(n \log n)$ G_{refresh}



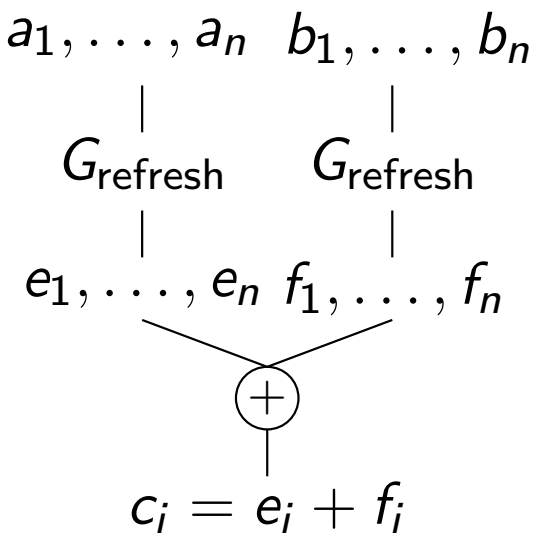


Linear Gadgets

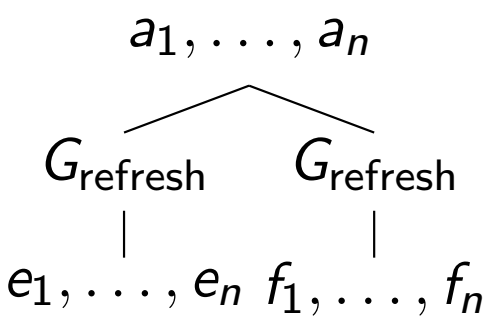
Constructions

Using $\mathcal{O}(n \log n)$ G_{refresh}

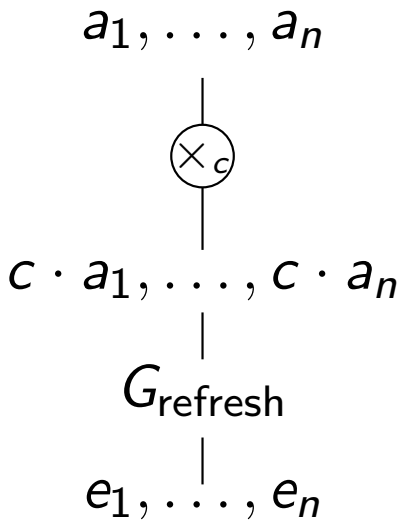
G_{add}



G_{copy}



G_{cmult}

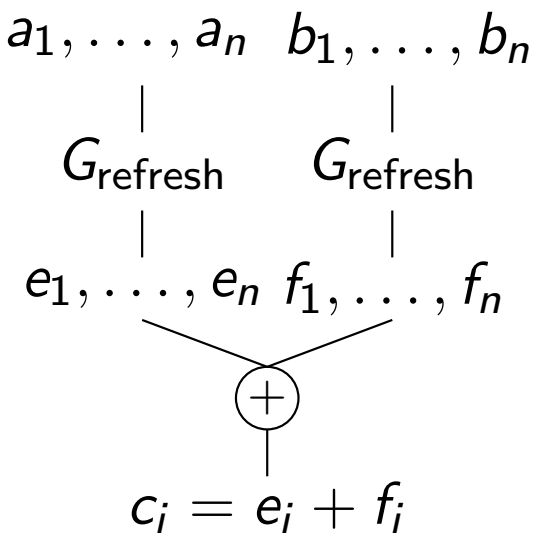


Linear Gadgets

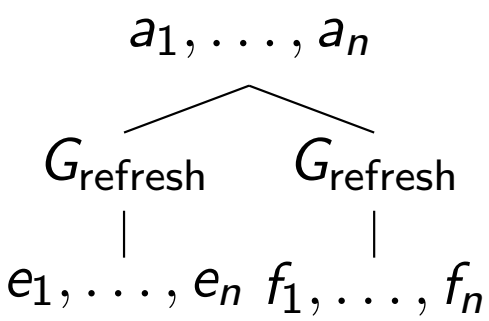
Constructions

Using $\mathcal{O}(n \log n)$ G_{refresh}

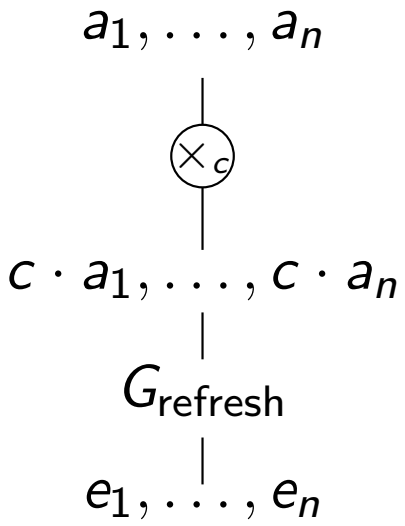
G_{add}



G_{copy}



G_{cmult}



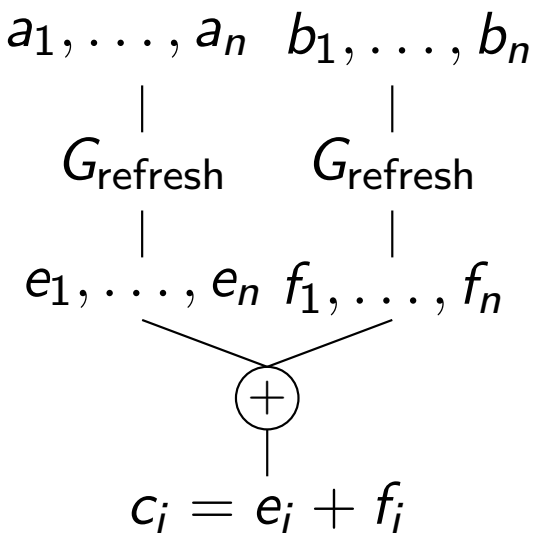
- Complexity in $\mathcal{O}(n \log n)$

Linear Gadgets

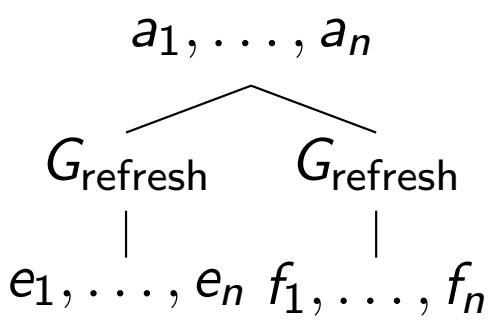
Constructions

Using $\mathcal{O}(n \log n)$ G_{refresh}

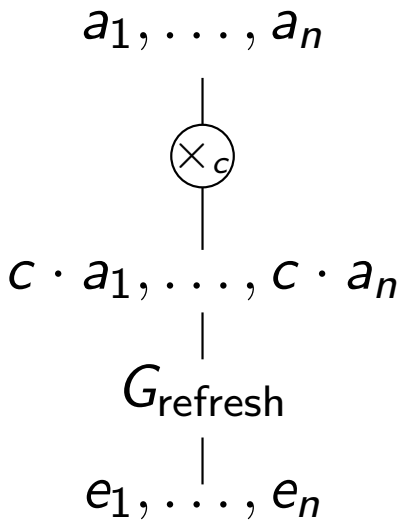
G_{add}



G_{copy}



G_{cmult}



- Complexity in $\mathcal{O}(n \log n)$
- RPE secure with $d = d_{\max} = \dfrac{n+1}{2}$

Multiplication Gadget

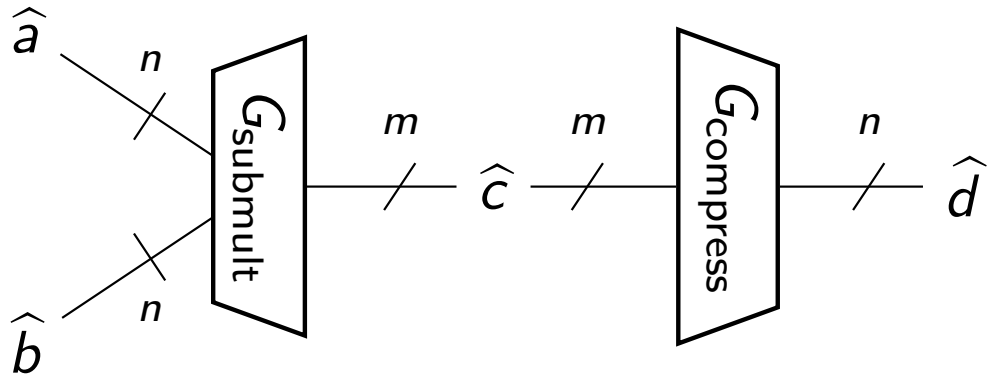
Construction from G_{submult}, G_{compress}

G_{mult} (over $\mathbb{K}$) construction from 2 subgadgets

Multiplication Gadget

Construction from G_{submult}, G_{compress}

G_{mult} (over $\mathbb{K}$) construction from 2 subgadgets



Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

Inputs a,b (illustration with 3 shares), field $\mathbb{K}$

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

Inputs a,b (illustration with 3 shares), field $\mathbb{K}$

$$\gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \gamma_{2,1} & \gamma_{2,2} & \gamma_{2,3} \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{pmatrix} \qquad \delta = \begin{pmatrix} 1-\gamma_{1,1} & 1-\gamma_{2,1} & 1-\gamma_{3,1} \\ 1-\gamma_{1,2} & 1-\gamma_{2,2} & 1-\gamma_{3,2} \\ 1-\gamma_{1,3} & 1-\gamma_{2,3} & 1-\gamma_{3,3} \end{pmatrix}$$

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

Inputs a,b (illustration with 3 shares), field $\mathbb{K}$

$$\gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \gamma_{2,1} & \gamma_{2,2} & \gamma_{2,3} \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{pmatrix} \qquad \delta = \begin{pmatrix} 1-\gamma_{1,1} & 1-\gamma_{2,1} & 1-\gamma_{3,1} \\ 1-\gamma_{1,2} & 1-\gamma_{2,2} & 1-\gamma_{3,2} \\ 1-\gamma_{1,3} & 1-\gamma_{2,3} & 1-\gamma_{3,3} \end{pmatrix}$$

$$c_1 \leftarrow \big((r_1 + a_1) + (r_2 + a_2) + (r_3 + a_3)\big) \cdot \big((s_1 + b_1) + (s_2 + b_2) + (s_3 + b_3)\big)$$

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

Inputs a, b (illustration with 3 shares), field $\mathbb{K}$

$$\gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \gamma_{2,1} & \gamma_{2,2} & \gamma_{2,3} \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{pmatrix} \qquad \delta = \begin{pmatrix} 1-\gamma_{1,1} & 1-\gamma_{2,1} & 1-\gamma_{3,1} \\ 1-\gamma_{1,2} & 1-\gamma_{2,2} & 1-\gamma_{3,2} \\ 1-\gamma_{1,3} & 1-\gamma_{2,3} & 1-\gamma_{3,3} \end{pmatrix}$$

$$c_1 \leftarrow \big((r_1 + a_1) + (r_2 + a_2) + (r_3 + a_3)\big) \cdot \big((s_1 + b_1) + (s_2 + b_2) + (s_3 + b_3)\big)$$

$$c_2 \leftarrow -r_1 \cdot \big((\delta_{1,1} \cdot s_1 + b_1) + (\delta_{1,2} \cdot s_2 + b_2) + (\delta_{1,3} \cdot s_3 + b_3)\big)$$

$$c_3 \leftarrow -r_2 \cdot \big((\delta_{2,1} \cdot s_1 + b_1) + (\delta_{2,2} \cdot s_2 + b_2) + (\delta_{2,3} \cdot s_3 + b_3)\big)$$

$$c_4 \leftarrow -r_3 \cdot \big((\delta_{3,1} \cdot s_1 + b_1) + (\delta_{3,2} \cdot s_2 + b_2) + (\delta_{3,3} \cdot s_3 + b_3)\big)$$

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

Inputs a, b (illustration with 3 shares), field $\mathbb{K}$

$$\gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \gamma_{2,1} & \gamma_{2,2} & \gamma_{2,3} \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{pmatrix} \qquad \delta = \begin{pmatrix} 1-\gamma_{1,1} & 1-\gamma_{2,1} & 1-\gamma_{3,1} \\ 1-\gamma_{1,2} & 1-\gamma_{2,2} & 1-\gamma_{3,2} \\ 1-\gamma_{1,3} & 1-\gamma_{2,3} & 1-\gamma_{3,3} \end{pmatrix}$$

$$c_1 \leftarrow \big((r_1 + a_1) + (r_2 + a_2) + (r_3 + a_3)\big) \cdot \big((s_1 + b_1) + (s_2 + b_2) + (s_3 + b_3)\big)$$

$$c_2 \leftarrow -r_1 \cdot \big((\delta_{1,1} \cdot s_1 + b_1) + (\delta_{1,2} \cdot s_2 + b_2) + (\delta_{1,3} \cdot s_3 + b_3)\big)$$
$$c_3 \leftarrow -r_2 \cdot \big((\delta_{2,1} \cdot s_1 + b_1) + (\delta_{2,2} \cdot s_2 + b_2) + (\delta_{2,3} \cdot s_3 + b_3)\big)$$
$$c_4 \leftarrow -r_3 \cdot \big((\delta_{3,1} \cdot s_1 + b_1) + (\delta_{3,2} \cdot s_2 + b_2) + (\delta_{3,3} \cdot s_3 + b_3)\big)$$

$$c_5 \leftarrow -s_1 \cdot \big((\gamma_{1,1} \cdot r_1 + a_1) + (\gamma_{1,2} \cdot r_2 + a_2) + (\gamma_{1,3} \cdot r_3 + a_3)\big)$$
$$c_6 \leftarrow -s_2 \cdot \big((\gamma_{2,1} \cdot r_1 + a_1) + (\gamma_{2,2} \cdot r_2 + a_2) + (\gamma_{2,3} \cdot r_3 + a_3)\big)$$
$$c_7 \leftarrow -s_3 \cdot \big((\gamma_{3,1} \cdot r_1 + a_1) + (\gamma_{3,2} \cdot r_2 + a_2) + (\gamma_{3,3} \cdot r_3 + a_3)\big)$$

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

- uses $2n$ random values

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

- uses $2n$ random values
- outputs $2n + 1$ shares

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

- uses $2n$ random values
- outputs $2n + 1$ shares
- performs $2n + 1$ multiplications operations

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

- uses $2n$ random values
- outputs $2n + 1$ shares
- performs $2n + 1$ multiplications operations
- performs $2n^2$ multiplications by a constant

Multiplication Gadget

Extension of G_{submult} by *Belaïd et al. - Crypto 2017*

G_{submult}

- uses $2n$ random values
- outputs $2n + 1$ shares
- performs $2n + 1$ multiplications operations
- performs $2n^2$ multiplications by a constant
- is proven to be secure for G_{mult} RPE secure construction, **for the right choice of constants in** γ (can be chosen uniformly at random if the field is large enough)

Multiplication Gadget

New Construction of G_{compress}

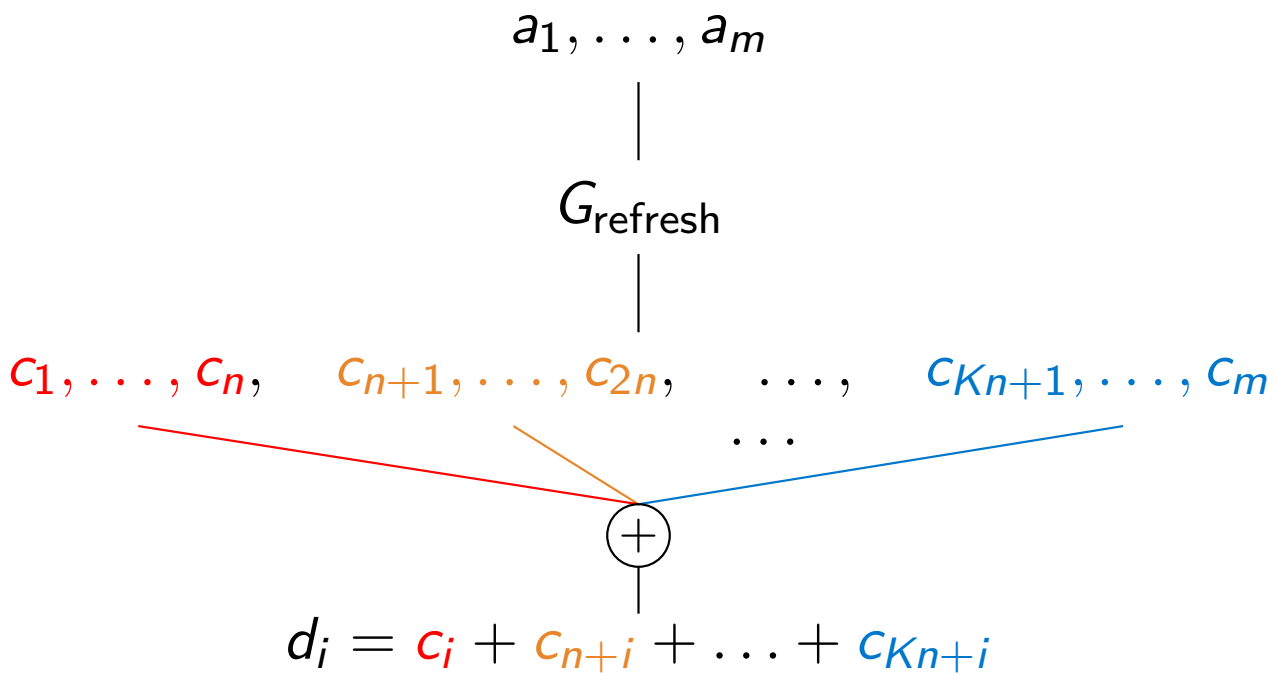
The $[m : n]$-compression gadget proposed by *Belaïd et al. - Crypto 2017* is not secure as claimed

Multiplication Gadget

New Construction of G_{compress}

The $[m : n]$-compression gadget proposed by *Belaïd et al. - Crypto 2017* is not secure as claimed

New Compression gadget



Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\mathsf{refresh}}| + m)$

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$
- is proven secure for G_{mult} RPE secure construction

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$
- is proven secure for G_{mult} RPE secure construction

Using G_{submult} described earlier, and new G_{compress}, we get G_{mult}:

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$
- is proven secure for G_{mult} RPE secure construction

Using G_{submult} described earlier, and new G_{compress}, we get G_{mult}:

- performs $\mathcal{O}(n)$ multiplications between variables

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$
- is proven secure for G_{mult} RPE secure construction

Using G_{submult} described earlier, and new G_{compress}, we get G_{mult}:

- performs $\mathcal{O}(n)$ multiplications between variables
- uses $\mathcal{O}(n \log n)$ random values

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

- is of size $\mathcal{O}(|G_{\text{refresh}}| + m)$
- using $\mathcal{O}(n \log n)$ G_{refresh}, has complexity $\mathcal{O}(m \log m)$
- With $m = \mathcal{O}(n)$ (from G_{submult}), has complexity $\mathcal{O}(n \log n)$
- is proven secure for G_{mult} RPE secure construction

Using G_{submult} described earlier, and new G_{compress}, we get G_{mult}:

- performs $\mathcal{O}(n)$ multiplications between variables
- uses $\mathcal{O}(n \log n)$ random values
- is RPE secure with amplification order $d = d_{\text{max}} = \dfrac{n+1}{2}$

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

New Linear gadgets $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

New Linear gadgets $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

New G_{mult} with $\mathcal{O}(n)$ multiplications between variables

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

New Linear gadgets $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

New G_{mult} with $\mathcal{O}(n)$ multiplications between variables

All gadgets of amplification order $d = \dfrac{n+1}{2}$

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

New Linear gadgets $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

New G_{mult} with $\mathcal{O}(n)$ multiplications between variables

All gadgets of amplification order $d = \dfrac{n+1}{2}$

Complexity of expansion of a circuit C:

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

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

New Linear gadgets $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

New G_{mult} with $\mathcal{O}(n)$ multiplications between variables

All gadgets of amplification order $d = \dfrac{n+1}{2}$

Complexity of expansion of a circuit C:

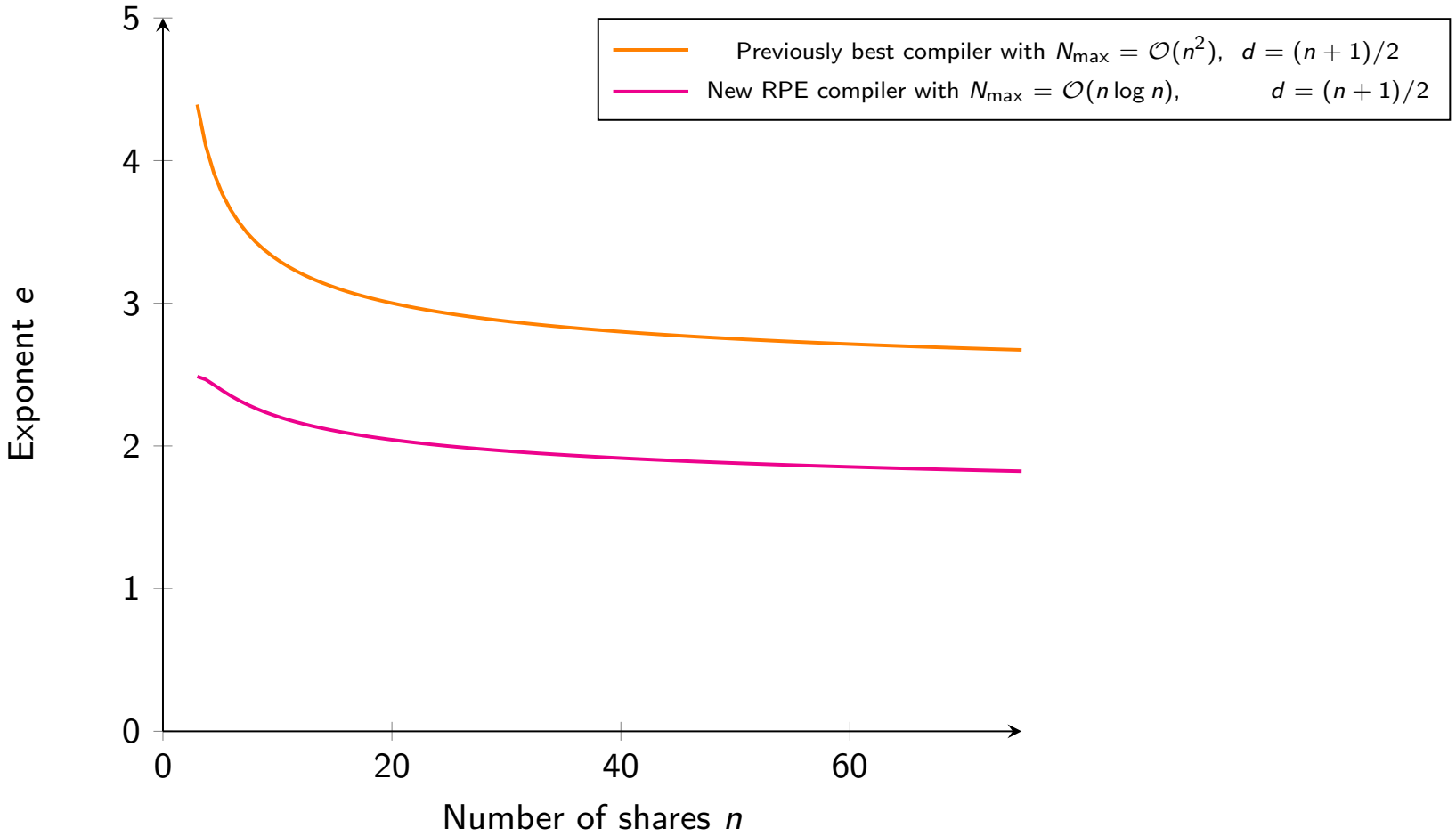
$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{log(\mathsf{N}_{\mathsf{max}})}{log(\mathsf{d})}$$

$\mathbf{N_{max}} \approx \max(\#\times \text{ in } G_{\mathsf{mult}}, \ \#(+,||) \text{ in } G_{\mathsf{add}}, G_{\mathsf{copy}}, \ \#\times_c \text{ in } G_{\mathsf{cmult}}) = \mathcal{O}(n \log n)$

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

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



Conclusion

Conclusion

- RP security analysis is gaining popularity

Conclusion

- RP security analysis is gaining popularity
- RP expansion is a promising strategy

Conclusion

- RP security analysis is gaining popularity
- RP expansion is a promising strategy
- Automatic formal verification tools are regularly updated

Conclusion

- RP security analysis is gaining popularity
- RP expansion is a promising strategy
- Automatic formal verification tools are regularly updated

Future Work

Conclusion

- RP security analysis is gaining popularity
- RP expansion is a promising strategy
- Automatic formal verification tools are regularly updated

Future Work

- Look for gadgets which tolerate the best leakage rate

Conclusion

- RP security analysis is gaining popularity
- RP expansion is a promising strategy
- Automatic formal verification tools are regularly updated

Future Work

- Look for gadgets which tolerate the best leakage rate
- Upcoming practical projects: analysis of RP secure circuits in real-life leakage settings