Dynamic Random Probing Expansion with Quasi Linear Asymptotic Complexity

Sonia Belaïd [1], Matthieu Rivain [1],
Abdul Rahman Taleb [1,2] and Damien Vergnaud [2,3]

[1] CryptoExperts, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[3] Institut Universitaire de France, France

December 7, 2021

Side-Channel Attacks & Masking

Security against **side-channel attacks**

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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

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

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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

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

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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 $\mathbb{K}$

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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 $\mathbb{K}$

Side-Channel Attacks & Masking

Security against **side-channel attacks**

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 $n = 2$

Side-Channel Attacks & Masking

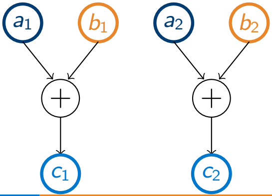
Security against **side-channel attacks**

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

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

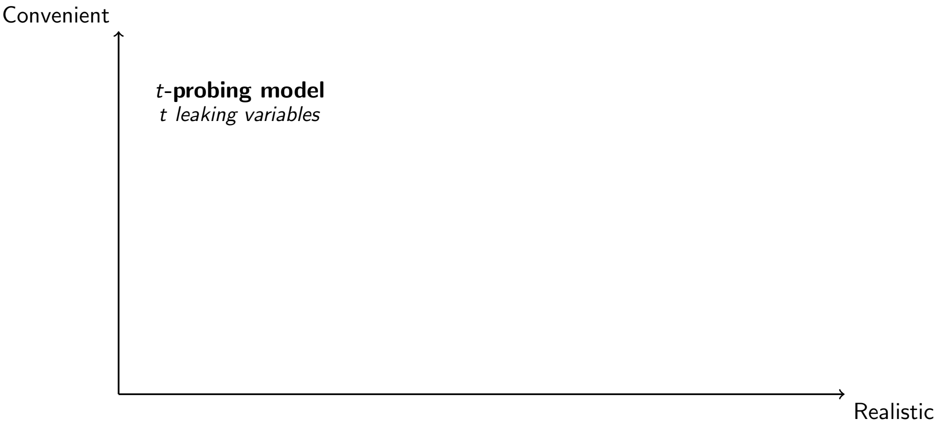
$(+, \times, ||)$ operations over $\mathbb{K} \longrightarrow (G_{\text{add}}, G_{\text{mult}}, G_{\text{copy}}, \mathbf{G}_{\text{refresh}})$ n-share circuits over $\mathbb{K}$

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

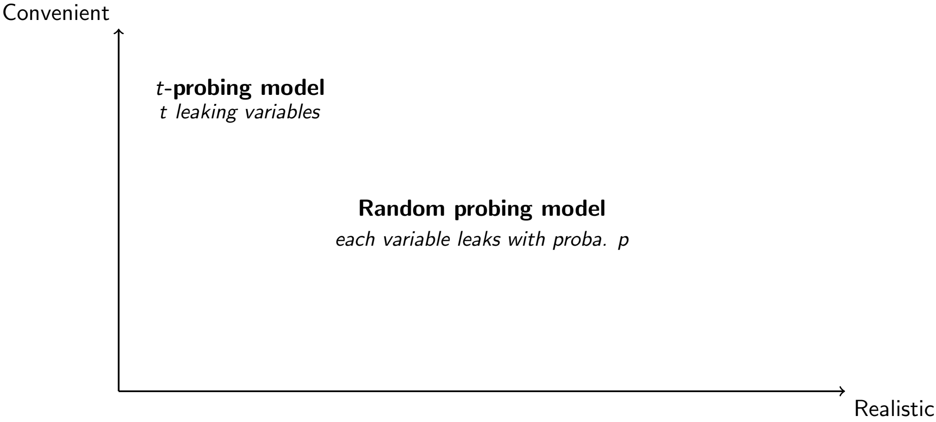


Leakage Models

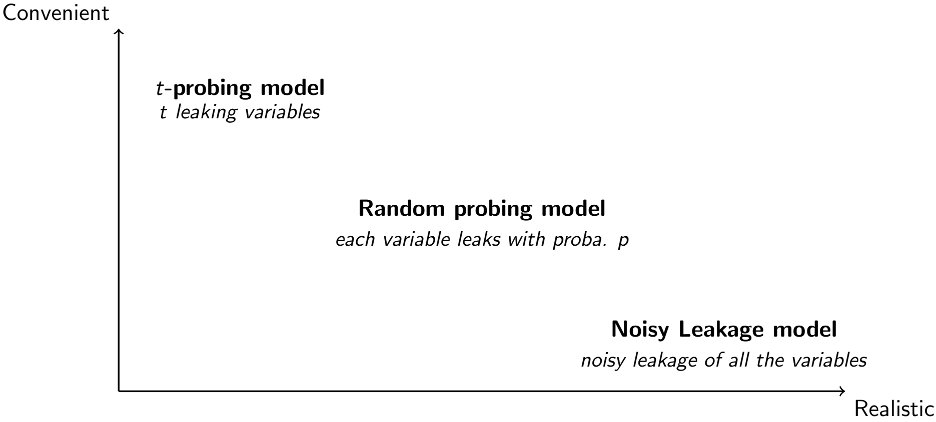
Leakage Models



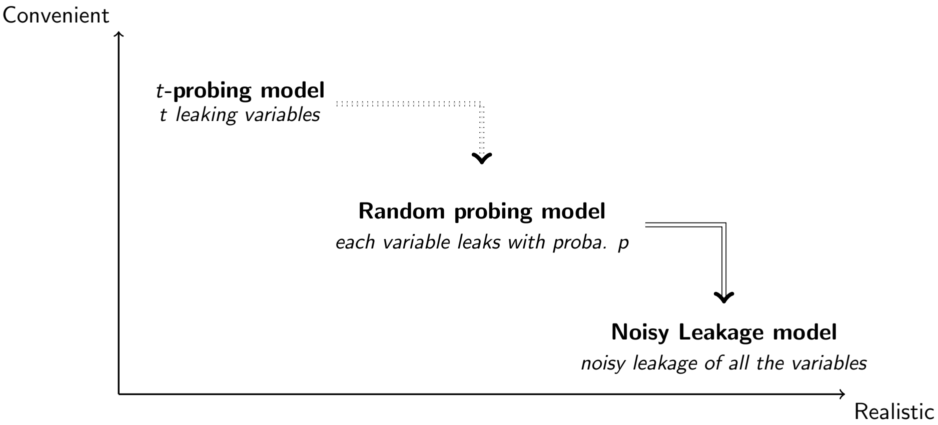
Leakage Models



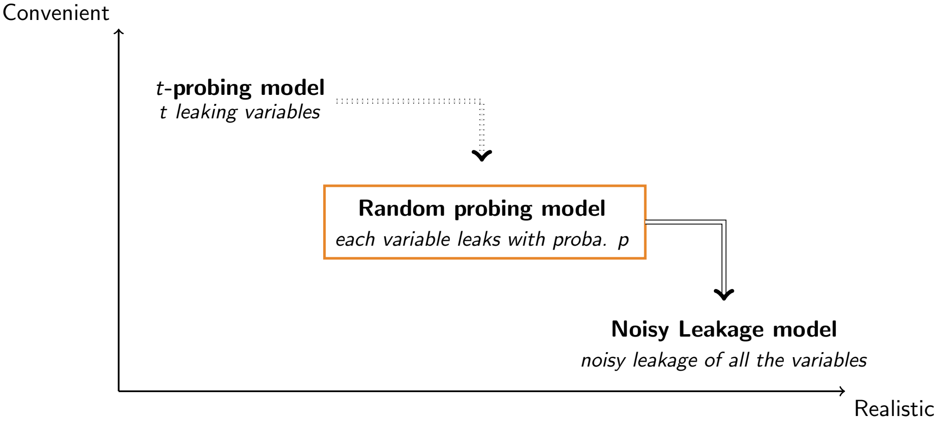
Leakage Models



Leakage Models



Leakage Models



Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

[EUROCRYPT 2021] On the Power of Expansion: More Efficient Constructions in the Random Probing Model. *Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb*.

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

[EUROCRYPT 2021] On the Power of Expansion: More Efficient Constructions in the Random Probing Model. *Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb*.

- In-depth analysis of RP expansion

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

[EUROCRYPT 2021] On the Power of Expansion: More Efficient Constructions in the Random Probing Model. *Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb*.

- In-depth analysis of RP expansion
- Generic constructions for RP expansion with improved complexities

Prior Works

[CRYPTO 2020] Random probing security: verification, composition, expansion and new constructions. *Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Abdul Rahman Taleb*.

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

[EUROCRYPT 2021] On the Power of Expansion: More Efficient Constructions in the Random Probing Model. *Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb*.

- In-depth analysis of RP expansion
- Generic constructions for RP expansion with improved complexities
- Concrete instantiations for RP expansion tolerating a leakage rate of $p \approx 2^{-7.5}$

Contributions

- Introduction of **Dynamic** Random Probing Expansion (RPE)

Contributions

- Introduction of **Dynamic** Random Probing Expansion (RPE)
- Generalization of RPE to support any basic operations (*e.g.* multiplication by a constant G_{cmult})

Contributions

- Introduction of **Dynamic** Random Probing Expansion (RPE)
- Generalization of RPE to support any basic operations (*e.g.* multiplication by a constant G_{cmult})
- Construction of n-share RPE-secure $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity (using G_{refresh} by *Battistello et al. - CHES 2016*)

Contributions

- Introduction of **Dynamic** Random Probing Expansion (RPE)
- Generalization of RPE to support any basic operations (*e.g.* multiplication by a constant G_{cmult})
- Construction of n-share RPE-secure $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity (using G_{refresh} by *Battistello et al. - CHES 2016*)
- Construction of RPE-secure G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables, from:

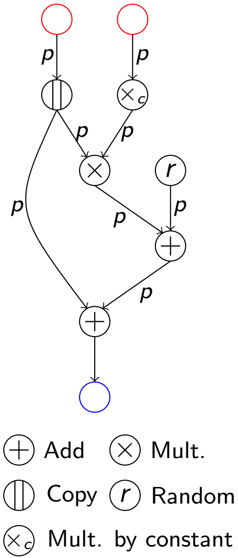
Contributions

- Introduction of **Dynamic** Random Probing Expansion (RPE)
- Generalization of RPE to support any basic operations (*e.g.* multiplication by a constant G_{cmult})
- Construction of n-share RPE-secure $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity (using G_{refresh} by *Battistello et al. - CHES 2016*)
- Construction of RPE-secure G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables, from:
  - extension of sub-multiplication gadget $G_{\text{submult}} : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^{2n+1}$ by *Belaïd et al. - Crypto 2017*

Contributions

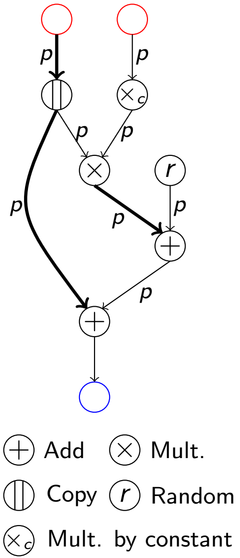
- Introduction of **Dynamic** Random Probing Expansion (RPE)
- Generalization of RPE to support any basic operations (*e.g.* multiplication by a constant G_{cmult})
- Construction of n-share RPE-secure $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity (using G_{refresh} by *Battistello et al. - CHES 2016*)
- Construction of RPE-secure G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables, from:
  - extension of sub-multiplication gadget $G_{\text{submult}} : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^{2n+1}$ by *Belaïd et al. - Crypto 2017*
  - new compression gadget $G_{\text{compress}} : \mathbb{K}^{2n+1} \to \mathbb{K}^n$

RP Security



(p, ε)-RP Security

RP Security



Add Mult.

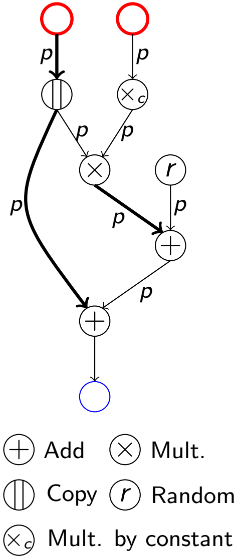
Copy Random

Mult. by constant

(p, ε)-RP Security

W set of wires

RP Security

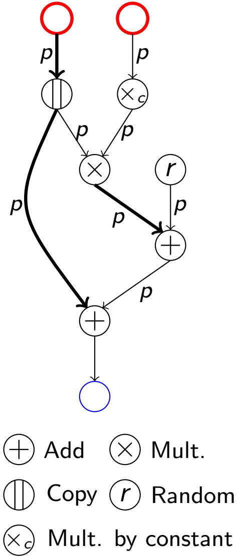


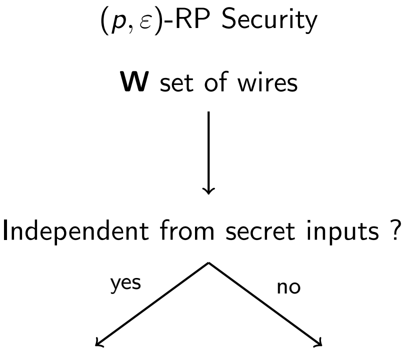
(p, ε)-RP Security

W set of wires

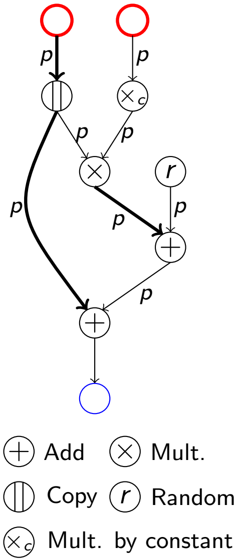
↓

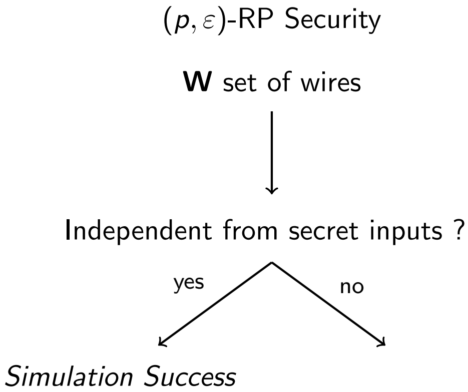
Independent from secret inputs ?

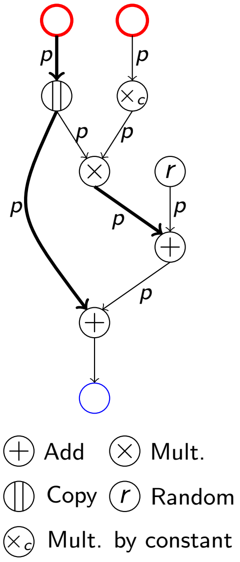

p
p
p
p
r
p
p
p
p
Add
Mult.
Copy
Random
Mult. by constant

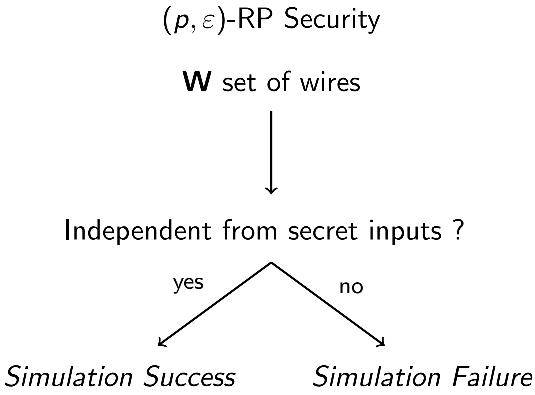

(p, ε)-RP Security
W set of wires
Independent from secret inputs ?
yes
no

RP Security

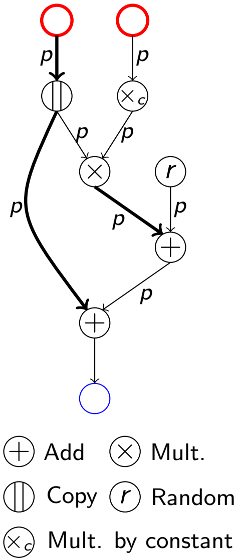

p
p
p
p
r
p
p
p
p
Add
Mult.
Copy
Random
Mult. by constant


(p, ε)-RP Security
W set of wires
Independent from secret inputs ?
yes
no
Simulation Success

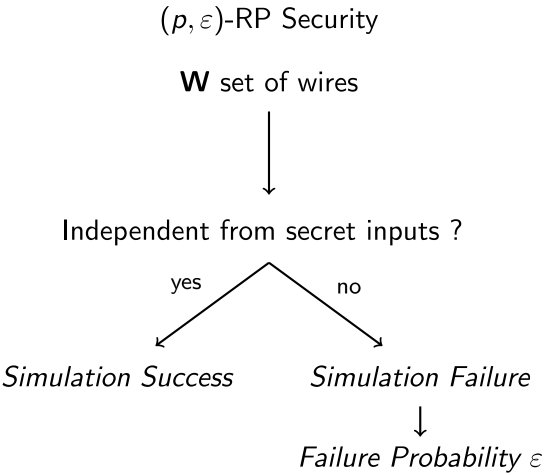

p
p
p
p
p
p
p
r
p
Add
Mult.
Copy
Random
Mult. by constant


(p, ε)-RP Security
W set of wires
Independent from secret inputs ?
yes
no
Simulation Success
Simulation Failure

RP Security



p
p
p
p
p
p
p
p
r
Add
Mult.
Copy
Random
Mult. by constant



(p, ε)-RP Security
W set of wires
Independent from secret inputs ?
yes
no
Simulation Success
Simulation Failure
Failure Probability ε

RP Expansion
Illustration

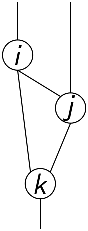
Using n-share gadgets $G_1, \ldots, G_\beta$

RP Expansion
Illustration

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

RP Expansion
Illustration

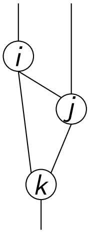
Using n-share gadgets $G_1, \ldots, G_\beta$



RP Expansion

Illustration

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

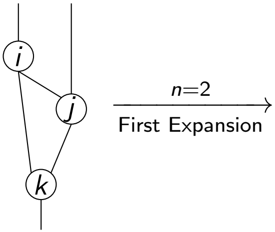


Leakage probability
p

RP Expansion

Illustration

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



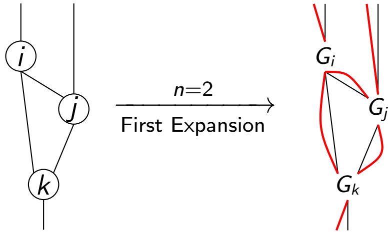
Leakage probability

p

RP Expansion

Illustration

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

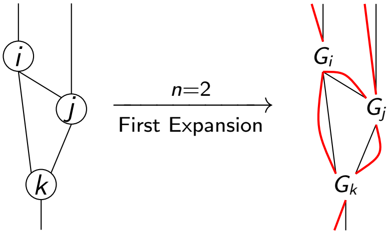


Leakage probability
p

RP Expansion

Illustration

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

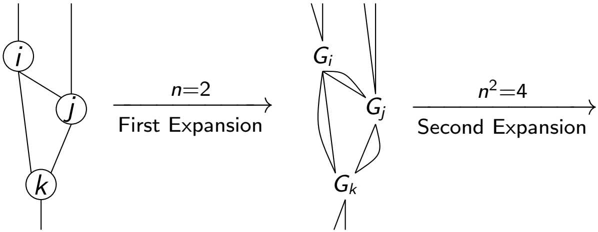


Leakage probability
p

Simulation Failure
ε

RP Expansion
Illustration

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



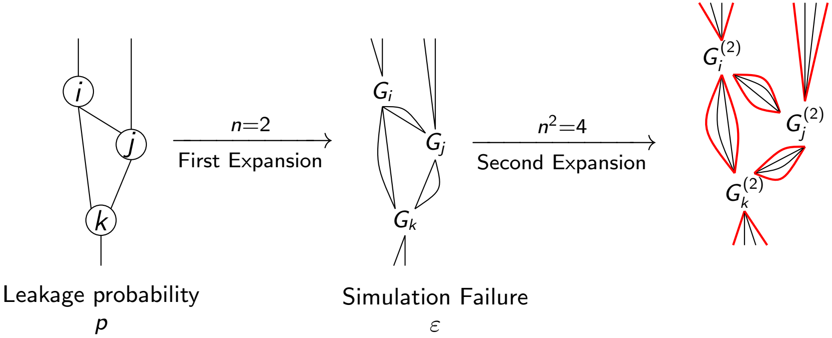
Leakage probability
p

Simulation Failure
ε

RP Expansion

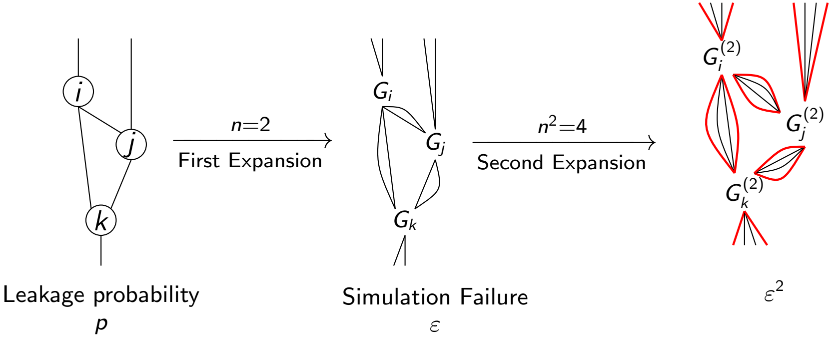
Illustration

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



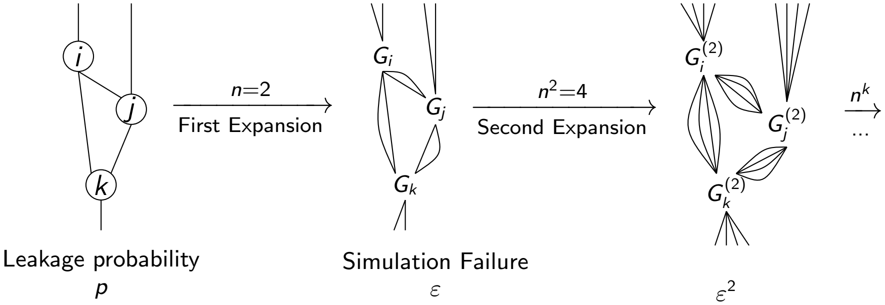
RP Expansion
Illustration

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



RP Expansion
Illustration

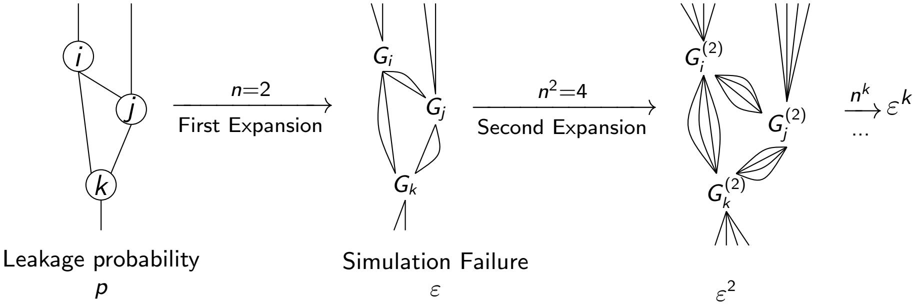
Using n-share gadgets $G_1, \ldots, G_\beta$



RP Expansion

Illustration

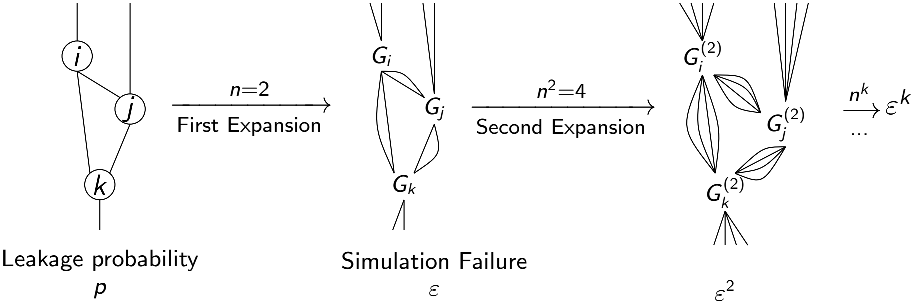
Using n-share gadgets $G_1, \ldots, G_\beta$



RP Expansion

Illustration

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



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 is stronger than RP)

RP Expansion

Definition

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

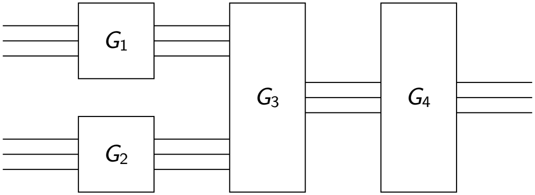
- (p, ε)-RP security of G (RPE is stronger than RP)
- **composition** of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares

RP Expansion

Definition

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

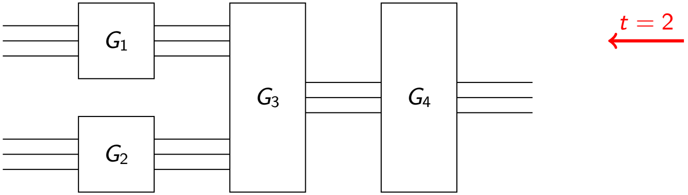
- (p, ε)-RP security of G (RPE is stronger than RP)
- **composition** of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Expansion
Definition

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

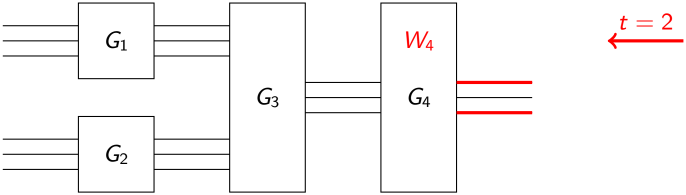
- (p, ε)-RP security of G (RPE is stronger than RP)
- **composition** of G with other RP secure gadgets: ability to simulate any set W of internal wires and t output shares using t input shares



RP Expansion
Definition

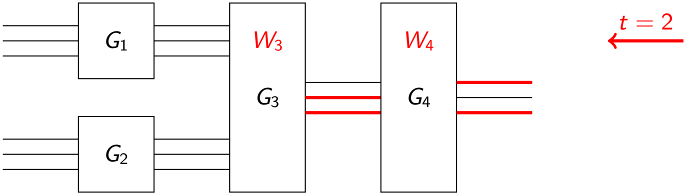
$(\mathbf{t}, p, \varepsilon)$-**RP expandability** (RPE) of gadget G garantees:

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



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

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

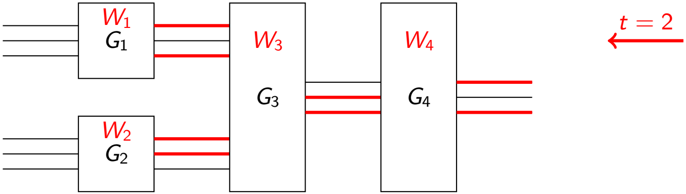


RP Expansion

Definition

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

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

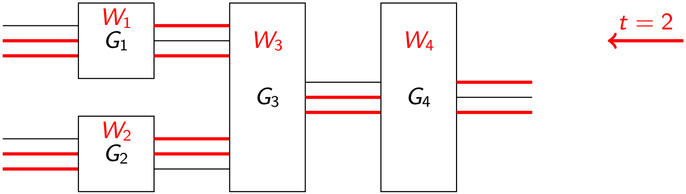


RP Expansion

Definition

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

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

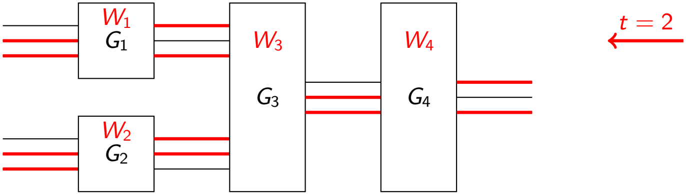


RP Expansion

Definition

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

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



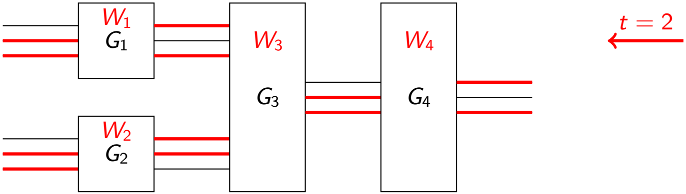
- Independent failure probability on each input sharing

RP Expansion

Definition

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

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



- 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:

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

RP Expansion
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

RP Expansion

Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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

RP Expansion

Parameters

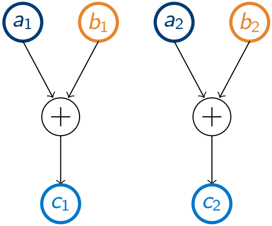
Complexity of expanded circuit C of security parameter κ:

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

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

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

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



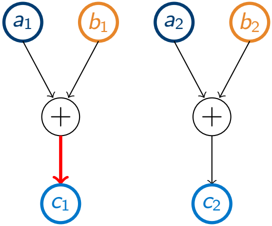
RP Expansion
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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

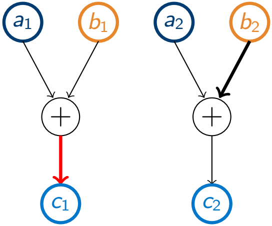
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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



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

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

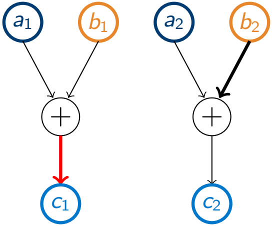
RP Expansion
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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



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

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

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

RP Expansion

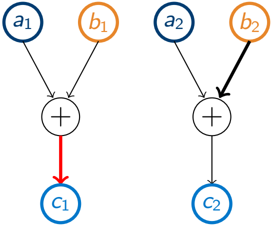
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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

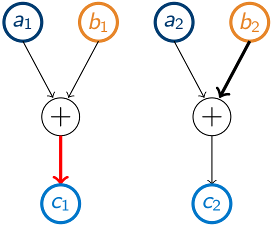
RP Expansion
Parameters

Complexity of expanded circuit C of security parameter κ:

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

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

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:

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

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

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:

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

$\mathbf{N}_{\mathbf{max}} \approx \max(\# \times \text{ in } G_{\text{mult}}, \ \#(+, ||) \text{ in } G_{\text{add}}, G_{\text{copy}}, \ \# \times_c \text{ in } G_{\text{cmult}})$

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:

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

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

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 ($d_{\max} = \dfrac{n+1}{2}$)

Dynamic RP Expansion
Idea

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

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

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 $n_1, \ldots, n_\ell$

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

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 $n_1, \ldots, n_\ell$

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

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

Dynamic RP Expansion

Idea

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

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

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

Dynamic RP Expansion

Idea

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

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

Leakage rate p

$n_1^{k_1}$ shares

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

$n_2^{k_2} \cdot n_1^{k_1}$ shares

$\varepsilon_2^{k_2} = f_2^{(k_2)}(f_1^{(k_1)}(p))$

Dynamic RP Expansion

Idea

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

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

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_\ell$
	$n_1^{k_1}$ 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 $n_1, \ldots, n_\ell$

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

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

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 $n_1, \ldots, n_\ell$

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

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_\ell$
	$n_1^{k_1}$ 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?

n-share RPE compilers:

Dynamic RP Expansion

Motivation

n-share RPE compilers:

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

Dynamic RP Expansion

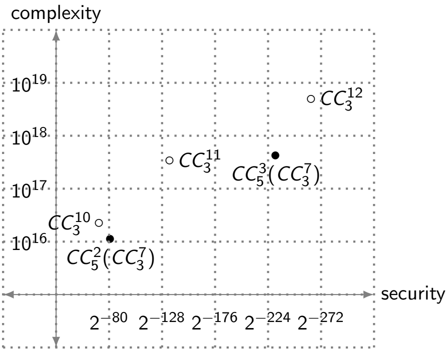
Motivation

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

n-share RPE compilers:

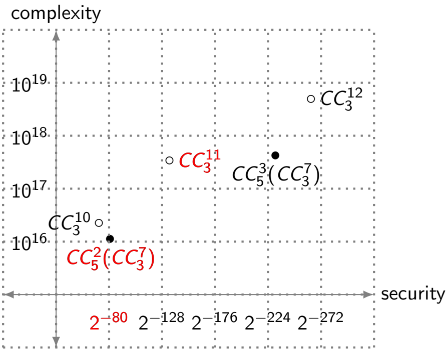
- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \dfrac{n+1}{2} \implies$ have better asymptotic complexity



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 by *Belaïd et al. - EuroCrypt 2021*

n-share RPE compilers:

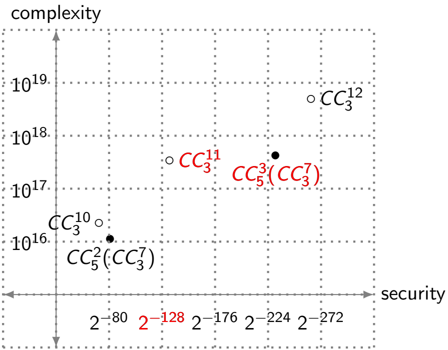
- **small** n: fewer sets of probes that reveal the secret $\implies$ tolerate better leakage rate p
- **big** n: have higher amp. order $d_{\max} = \frac{n+1}{2}$ $\implies$ have better asymptotic complexity



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 by *Belaïd et al. - EuroCrypt 2021*

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 by *Belaïd et al. - EuroCrypt 2021*

Dynamic RP Expansion

Motivation

2 possible directions:

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

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

In this work:

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

In this work:

- construction of n-share linear $G_{\mathrm{add}}, G_{\mathrm{copy}}, G_{\mathrm{cmult}}$ with $\mathcal{O}(n \log n)$ asymptotic complexity and maximal amp. order

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

In this work:

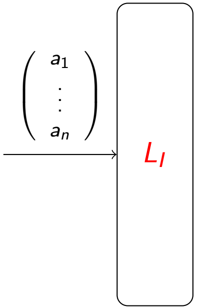
- construction of n-share linear $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ asymptotic complexity and maximal amp. order
- construction of n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables

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

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

Linear Gadgets
Building Block

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



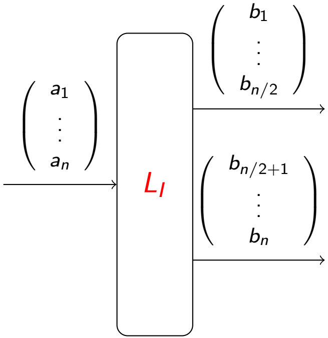
$n/2$ randoms

$b_i \leftarrow a_i + r_i$

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

Linear Gadgets
Building Block

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



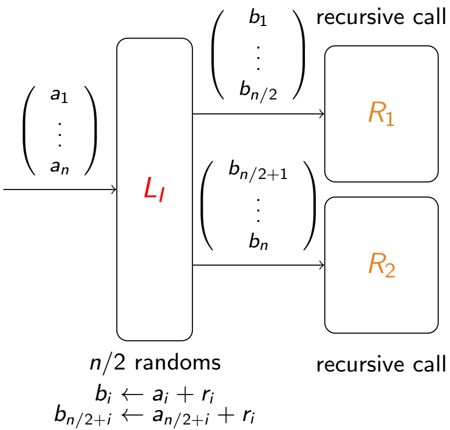
$n/2$ randoms

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

Linear Gadgets

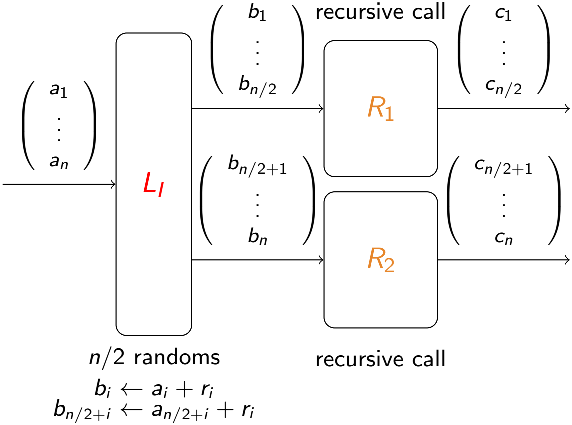
Building Block

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



Linear Gadgets
Building Block

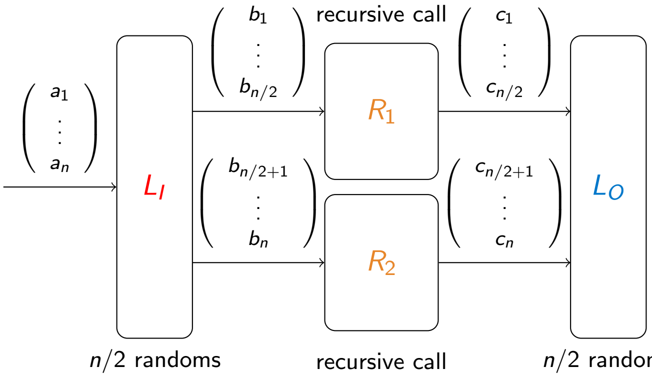
$\mathcal{O}(n \log n)$ refresh gadget G_{refresh} by *Battistello et al. - CHES 2016*:



Linear Gadgets

Building Block

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



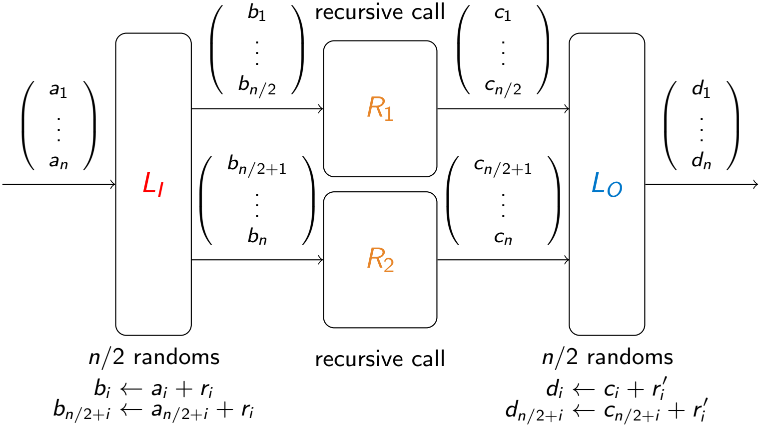
$n/2$ randoms

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

$n/2$ randoms

$d_i \leftarrow c_i + r'_i$
$d_{n/2+i} \leftarrow c_{n/2+i} + r'_i$

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



Linear Gadgets

Building Block

Example (4 shares):

$$d_1 \leftarrow (a_1 + r_1) + r_3 + r_5$$
$$d_2 \leftarrow (a_2 + r_2) + r_3 + r_6$$
$$d_3 \leftarrow (a_3 + r_1) + r_4 + r_5$$
$$d_4 \leftarrow (a_4 + r_2) + r_4 + r_6$$

Linear Gadgets
Building Block

Example (4 shares):

$$d_1 \leftarrow (a_1 + r_1) + r_3 + r_5$$
$$d_2 \leftarrow (a_2 + r_2) + r_3 + r_6$$
$$d_3 \leftarrow (a_3 + r_1) + r_4 + r_5$$
$$d_4 \leftarrow (a_4 + r_2) + r_4 + r_6$$

- 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*)

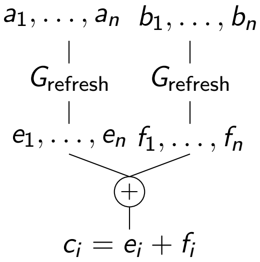
Linear Gadgets

Constructions

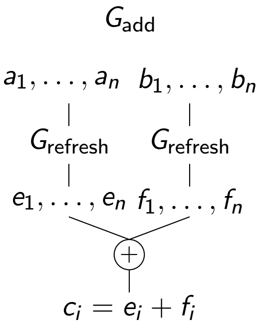
Using $\mathcal{O}(n \log n)$ G_{refresh}

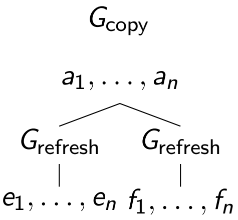
Using $\mathcal{O}(n \log n)$ G_{refresh}

G_{add}

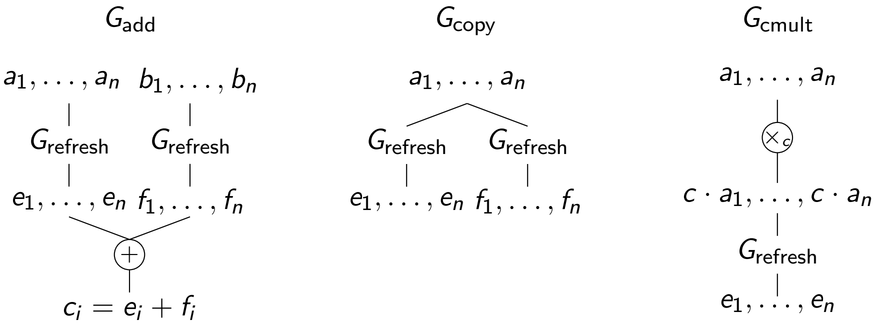


Using $\mathcal{O}(n \log n)$ G_{refresh}

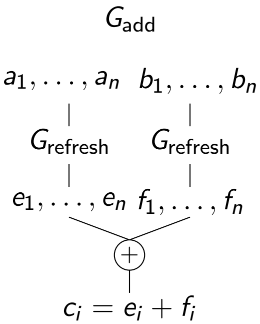


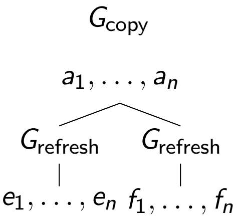


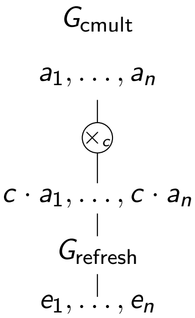
Using $\mathcal{O}(n \log n)$ G_{refresh}



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

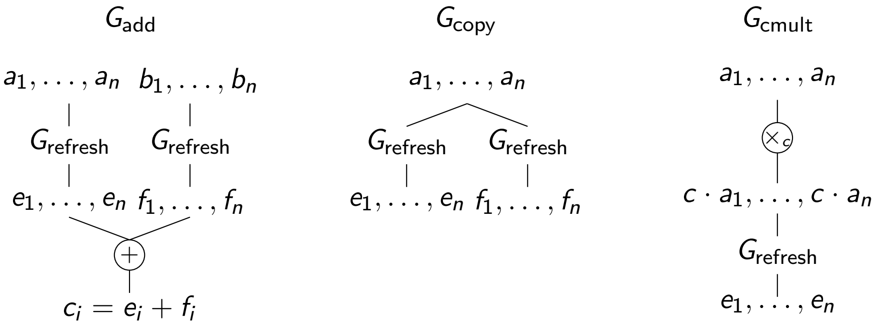






- Complexity in $\mathcal{O}(n \log n)$

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



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

Multiplication Gadget

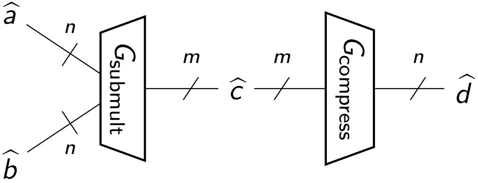
Construction from $G_{\text{submult}}, G_{\text{compress}}$

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

Multiplication Gadget

Construction from G_{submult}, G_{compress}

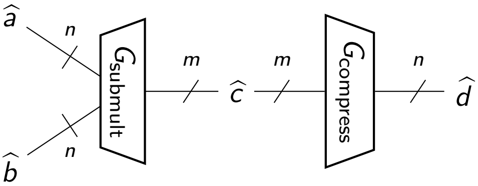
G_{mult} (over $\mathbb{K}$) construction from 2 subgadgets



Multiplication Gadget

Construction from G_{submult}, G_{compress}

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

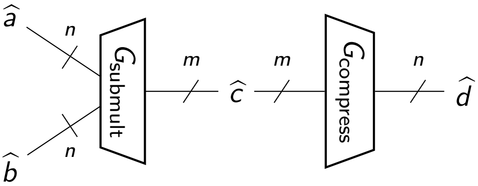


- In classical constructions, $m = \mathcal{O}(n^2)$

Multiplication Gadget

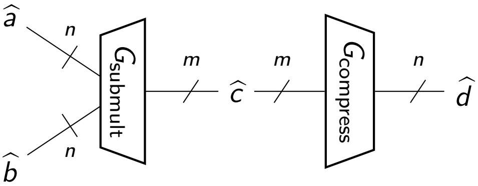
Construction from G_{submult}, G_{compress}

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



- In classical constructions, $m = \mathcal{O}(n^2)$
- G_{mult} must be RPE secure $\Longrightarrow$ **composition** of G_{submult} and G_{compress} must be RPE secure

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

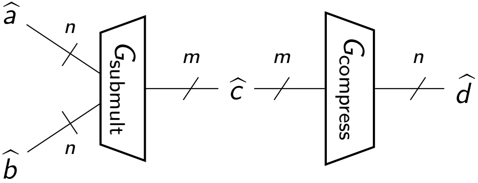


- In classical constructions, $m = \mathcal{O}(n^2)$
- G_{mult} must be RPE secure $\implies$ **composition** of G_{submult} and G_{compress} must be RPE secure
- extension of G_{submult} by *Belaïd et al. - Crypto 2017* with $\mathbf{m} = \mathbf{2n} + \mathbf{1}$

Multiplication Gadget

Construction from G_{submult}, G_{compress}

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



- In classical constructions, $m = \mathcal{O}(n^2)$
- G_{mult} must be RPE secure $\implies$ **composition** of G_{submult} and G_{compress} must be RPE secure
- extension of G_{submult} by *Belaïd et al. - Crypto 2017* with $\mathbf{m = 2n + 1}$
- New G_{compress} with complexity in $\mathcal{O}(m \log m)$

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 $\mathbb{K}$

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

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

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

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

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

Multiplication Gadget

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

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

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

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

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

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

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}

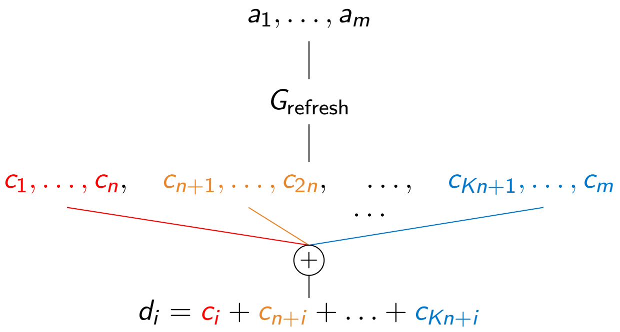
The $[m:n]$-compression gadget proposed by *Belaïd et al. - Crypto 2017* is not secure as claimed

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



New G_{compress}

Multiplication Gadget

New Construction of G_{compress}

New G_{compress}

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

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 $\mathcal{O}(m \log m)$

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 $d = d_{\max} = \dfrac{n+1}{2}$

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 $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

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

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

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 $d = \frac{n+1}{2}$

Complexity of expansion of a circuit C:

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

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:

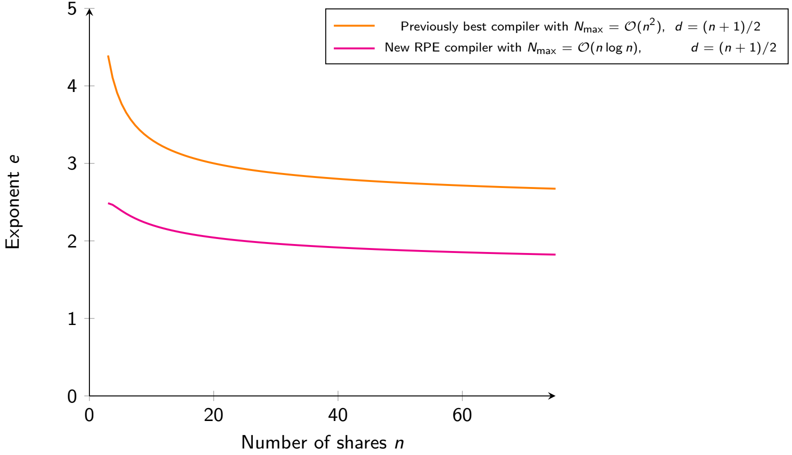
$$\mathcal{O}(|C|.\kappa^e), \quad e = \frac{log(\mathrm{N}_{\max})}{log(\mathrm{d})}$$

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

New RPE Compiler

With Quasi-Linear Asymptotic Complexity

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



Conclusion

- Construction of new RPE compiler with quasilinear complexity from

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\mathsf{add}}, G_{\mathsf{copy}}, G_{\mathsf{cmult}}$ with $\mathcal{O}(n \log n)$ complexity

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity
  - n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity
  - n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables
- Dynamic RPE (different compilers) is more interesting than static RPE (single compiler)

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity
  - n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables
- Dynamic RPE (different compilers) is more interesting than static RPE (single compiler)
  - start with RPE compiler with small nb. of shares tolerating the best leakage rate

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\mathrm{add}}, G_{\mathrm{copy}}, G_{\mathrm{cmult}}$ with $\mathcal{O}(n \log n)$ complexity
  - n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables
- Dynamic RPE (different compilers) is more interesting than static RPE (single compiler)
  - start with RPE compiler with small nb. of shares tolerating the best leakage rate
  - continue with RPE compiler with best asymptotic complexity (*e.g.* our new RPE compiler)

Conclusion

- Construction of new RPE compiler with quasilinear complexity from
  - n-share $G_{\text{add}}, G_{\text{copy}}, G_{\text{cmult}}$ with $\mathcal{O}(n \log n)$ complexity
  - n-share G_{mult} with $\mathcal{O}(n \log n)$ **randomness** and $\mathcal{O}(n)$ **multiplications** between variables
- Dynamic RPE (different compilers) is more interesting than static RPE (single compiler)
  - start with RPE compiler with small nb. of shares tolerating the best leakage rate
  - continue with RPE compiler with best asymptotic complexity (*e.g.* our new RPE compiler)
- Future work: Find gadgets with small nb. of shares (*e.g.* 3 shares) which tolerate the **best possible** leakage rate