Random Probing Security
Verification, Composition, Expansion and New Constructions

Sonia Belaïd ¹, Jean-Sébastien Coron ²
Emmanuel Prouff ³, Matthieu Rivain ¹
and Abdul Rahman Taleb ¹

¹ CryptoExperts, France
² University of Luxembourg
³ ANSSI, France

August 7, 2020
Side-Channel Attacks

- Plaintext
- Encryption Algorithm
- Secret Key
- Ciphertext
Side-Channel Attacks

Device (e.g. Smartcard)

Secret Key

Encryption Algorithm

 Plaintext

Ciphertext

Side Channel Analysis
Power Consumption
Execution Time
Electromagnetic Radiation
...

S. Belaid, JS. Coron, E. Prouff, M. Rivain, A. Taleb
Countermeasure
Higher-order Masking

Sensitive variable \( x \), group \( (G, \star) \):
Sensitive variable $x$, group $(G, \star)$:

$$x = x_0 \star \ldots \star x_{n-2} \star x_{n-1}$$
Sensitive variable $x$, group $(G, \ast)$:

$$x = x_0 \ast \ldots \ast x_{n-2} \ast x_{n-1}$$

uniformly at random from $G$
Sensitive variable $x$, group $(G, *)$:

$$x = \underbrace{x_0 \ast \cdots \ast x_{n-2}}_{\text{uniformly at random from } G} \ast \underbrace{x_{n-1}}_{x \ast x_0 \cdots x_{n-2}}$$
Countermeasure
Higher-order Masking

Sensitive variable $x$, group $(G, *)$:

$$x = \underbrace{x_0 \ast \ldots \ast x_{n-2}}_{\text{uniformly at random from } G} \ast \underbrace{x_{n-1}}_{x \ast x_0 \ast \ldots \ast x_{n-2}}$$

Security of masking schemes?
Leakage Models
Definitions

- Conventional
- Realistic

- t-probing model
- leaking variables

- Random probing model
- each variable leaks with proba. p

- Noisy Leakage model
- noisy leakage of all the variables
Leakage Models
Definitions

$t$-probing model
$t$ leaking variables
Leakage Models
Definitions

Convenient

Realistic

\textit{t-probing model}
\textit{t leaking variables}

Random probing model
\textit{each variable leaks with proba. }p
Leakage Models
Definitions

- **$t$-probing model**
  - $t$ leaking variables

- **Random probing model**
  - each variable leaks with proba. $p$

- **Noisy Leakage model**
  - noisy leakage of all the variables
Leakage Models

Existing Works

• Reduction property [Duc et al., 2014]

Probing Security \Rightarrow Random Probing Security \Rightarrow Noisy Leakage Security

Random Probing Constructions:

• [Ajtai, 2011, Andrychowicz et al., 2016] based on expander graphs
• [Ananth et al., 2018] based on secure multi-party computations

for a circuit $C$, tolerated leakage proba. $\approx 2^{-25}$.
Leakage Models

Existing Works

- Reduction property [Duc et al., 2014]

\[
\text{Probing Security} \Rightarrow \text{Random Probing Security} \Rightarrow \text{Noisy Leakage Security}
\]
Leakage Models

Existing Works

- Reduction property [Duc et al., 2014]

Probing Security $\implies$ Random Probing Security $\implies$ Noisy Leakage Security

Random Probing Constructions:
Leakage Models

Existing Works

- Reduction property [Duc et al., 2014]

Probing Security $\implies$ Random Probing Security $\implies$ Noisy Leakage Security

Random Probing Constructions:
- [Ajtai, 2011, Andrychowicz et al., 2016] based on expander graphs
Leakage Models

Existing Works

- Reduction property [Duc et al., 2014]

Probing Security $\implies$ Random Probing Security $\implies$ Noisy Leakage Security

Random Probing Constructions:

- [Ajtai, 2011, Andrychowicz et al., 2016] based on expander graphs
- [Ananth et al., 2018] based on secure multi-party computations $(O(|C|.poly(κ)))$ for a circuit $C$, tolerated leakage probability $\approx 2^{-25}$.
Random Probing Model

Contributions

• VRAPS Tool: (V)erifier of (RA)ndom (P)robing (S)ecurity.

• Random probing composability / expandability for global security level amplification (inspired from [Ananth et al., 2018]).

• Efficient instantiation from base gadgets in $O(|\mathcal{C}| \cdot \kappa^7.5)$ tolerating leakage probability $\approx 2^{-8}$. 
Random Probing Model

Contributions

- **VRAPS Tool**: (V)erifier of (RA)ndom (P)robing (S)ecurity.
Random Probing Model

Contributions

- **VRAPS Tool**: (V)erifier of (RA)ndom (P)robing (S)ecurity.

- **Random probing composability / expandability** for global security level amplification (inspired from [Ananth et al., 2018]).
Random Probing Model

Contributions

- **VRAPS Tool**: (V)erifier of (RA)ndom (P)robing (S)ecurity.

  - **Random probing composability / expandability** for global security level amplification (inspired from [Ananth et al., 2018]).

  - Efficient instantiation from base gadgets in $O(|C| \cdot \kappa^{7.5})$ tolerating leakage probability $\approx 2^{-8}$. 
Random Probing Security

Definition

(p, \epsilon)-Random Probing Security

\begin{center}
\begin{tikzpicture}
\node (A) at (0,0) {\scriptsize Add};
\node (B) at (1,0) {\scriptsize Mult.};
\node (C) at (-1,0) {\scriptsize Copy};
\node (D) at (0,1) {\scriptsize Random};
\draw [->] (A) -- (B);
\draw [->] (B) -- (C);
\draw [->] (C) -- (D);
\end{tikzpicture}
\end{center}
Random Probing Security

Definition

\( (p, \epsilon) \)-Random Probing Security

\( W \) set of wires

\( \oplus \) Add  \( \otimes \) Mult.

\( \bigcirc \) Copy  \( r \) Random
Random Probing Security

Definition

$(p, \epsilon)$-Random Probing Security

$W$ set of wires

Independent from secret inputs?

Add $\oplus$  
Mult. $\times$

Copy $\square$  
Random $r$
Random Probing Security

Definition

\((p, \epsilon)\)-Random Probing Security

\(W\) set of wires

Independent from secret inputs?

\[
\begin{array}{c}
\text{yes} \\
\text{no}
\end{array}
\]
Random Probing Security

Definition

\[(p, \epsilon)\text{-Random Probing Security}\]

\[W\] set of wires

Independent from secret inputs?

\[\begin{array}{c}
yes \\
no \\
\end{array}\]

Simulation Success

Add, Mult.

Copy, Random
Random Probing Security

Definition

$(p, \epsilon)$-Random Probing Security

Let $W$ be a set of wires. Is it independent from the secret inputs?

- **Simulation Success**: $W$ is independent from the secret inputs.
- **Simulation Failure**: $W$ is not independent from the secret inputs.

$+$ Add  $\times$ Mult.
$\square$ Copy  $r$ Random
Random Probing Security

Definition

\((p, \epsilon)\)-Random Probing Security

\[ W \text{ set of wires} \]

Independent from secret inputs?

- yes
- no

Simulation Success

Simulation Failure

Failure Probability \(\epsilon\)
Random Probing Security

Formal Verification: Method

\[ \text{s: number of wires} \]

\[ s = \sum_{i=1}^{\text{number of wires}} c_i p_i (1 - p) s - i \]

\[ \text{Failure probability} \epsilon \]

\[ \epsilon = \text{sum over all sizes} \text{ of wires} \]
Random Probing Security

Formal Verification: Method

\[ Pr(W) = p^{|W|}(1 - p)^{s-|W|} \]

\( s \): number of wires

\( W \): set of wires
\( W \) set of wires

\[
Pr(W) = p^{|W|}(1 - p)^{s - |W|}
\]

Failure probability \( \epsilon \)

\[
\epsilon = f(p) = \sum_{W} p^{|W|}(1 - p)^{s - |W|}
\]

Failure on \( W \)

\( s \): number of wires
**Random Probing Security**

**Formal Verification: Method**

\( W \) set of wires

\[
Pr(W) = p^{|W|}(1 - p)^{s-|W|}
\]

Failure probability \( \epsilon \)

\[
\epsilon = f(p) = \sum_{W} p^{|W|}(1 - p)^{s-|W|}
\]

\( c_i \): number of \( W \) of size \( i \) with *Simulation Failure*

\( s \): number of wires
Random Probing Security
Formal Verification: Method

\( W \) set of wires

\[
Pr(W) = p^{|W|}(1 - p)^{s - |W|}
\]

Failure probability \( \epsilon \)

\[
\epsilon = f(p) = \sum_{W} p^{|W|}(1 - p)^{s - |W|}
\]

\( c_i \): number of \( W \) of size \( i \) with Simulation Failure

\[
\epsilon = \sum_{i=1}^{s} c_i p^i (1 - p)^{s - i}
\]
Formal Verification: Algorithm (VRAPS Tool)

Input: circuit with \( s \) wires

Output: coefficients \( c_1, \ldots, c_s \)

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

7: \textbf{return} \hspace{0.5em} c
Random Probing 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 $c$
Random Probing 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.  $L \leftarrow \{\text{all } W \text{ of size } i\}$
4.  end for
5. return $c$
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. $L \leftarrow \{\text{all } W \text{ of size } i\}$
4. Apply rules inspired from *maskVerif* on $L$ [Barthe et al., 2015]
5. end for
6. return $c$
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. \hspace{1em} $L \leftarrow \{$all $W$ of size $i\}$
4. \hspace{1em} Apply rules inspired from maskVerif on $L$ [Barthe et al., 2015]
5. \hspace{1em} $c_i \leftarrow$ Nb. of failures in $L$
6. end for
7. return $c$
Goal: Achieve global random probing security
Goal: Achieve global random probing security

\((t, p, \epsilon)\)-Random probing composable \(n\)-share gadgets

\(G_{add}\) \hspace{2cm} \(G_{copy}\) 

\(G_{mult}\)
**Goal:** Achieve global random probing security

\[(t, p, \epsilon)\text{-Random probing composable } n\text{-share gadgets}\]

\[G_{\text{add}} \quad G_{\text{copy}} \quad G_{\text{mult}} \implies (p, |C|\cdot\epsilon)\text{-Random probing secure circuit } C\]
Random Probing Composability

Definition

Input Sharing

Output Sharing

1-to-1 3-share gadget

$s$: number of wires
Random Probing Composability

Definition

\((t, p, \epsilon)\)-Random Probing Composability

\(W\) set of wires

\begin{align*}
\text{yes} \quad & \text{Simulation Success} \\
\text{no} \quad & \text{Simulation Failure} \\
\end{align*}

\[\text{Failure Probability } \epsilon\]
Random Probing Composability

Definition

\[(t, p, \epsilon)\text{-Random Probing Composability}\]

\(W\) set of wires

Any set \(J\) of \(\leq t\) output wires

Simulation Success

Simulation Failure

Failure Probability \(\epsilon\)

1-to-1 3-share gadget

\(s\): number of wires
**Random Probing Composability**

**Definition**

\[(t, p, \epsilon)-\text{Random Probing Composability}\]

- **Input Sharing**
  - \(p\)
  - 1-to-1 3-share gadget
  - \(s\): number of wires

- **Output Sharing**

\(W\) set of wires

Any set \(J\) of \(\leq t\) output wires

Simulated from \(t\) shares of each secret input?

- yes
- no

**Simulation Success**

**Simulation Failure**

**Failure Probability** \(\epsilon\)
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{\text{add}}$, $G_{\text{copy}}$, $G_{\text{mult}}$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using \( n \)-share gadgets \( G_{\text{add}}, G_{\text{copy}}, G_{\text{mult}} \)

Leakage probability \( p \)
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$

Leakage probability $p$

$\text{First Expansion}$

First Expansion

$\text{Simulation Failure } \epsilon = f(p)$

$\epsilon = f^2(p)$

Condition: $f(p) < p$
Random Probing Expandability
Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$

Leakage probability $p$

First Expansion

Second Expansion

Condition: $f(p) < p$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{\text{add}}$, $G_{\text{copy}}$, $G_{\text{mult}}$

Leakage probability $p$

Simulation Failure $\epsilon = f(p)$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using \( n \)-share gadgets \( G_{add}, G_{copy}, G_{mult} \)

Leakage probability \( p \)

Simulation Failure \( \epsilon = f(p) \)

First Expansion

Second Expansion
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using \( n \)-share gadgets \( G_{add}, G_{copy}, G_{mult} \)

- \( n=2 \) First Expansion
  - \( G_{add}, G_{copy}, G_{mult} \)

- \( n^2=4 \) Second Expansion
  - \( G_{add}, G_{copy}, G_{mult} \)

Leakage probability \( p \)

Simulation Failure \( \epsilon = f(p) \)

\[ \text{Leakage probability} \quad p \]

\[ \text{Simulation Failure} \quad \epsilon = f(p) \]
Random Probing Expandability
Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$

Leakage probability $p$

Simulation Failure $\epsilon = f(p)$

First Expansion $n=2$

Second Expansion $n^2=4$

$\epsilon^2 = f^2(p)$
Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{add}$, $G_{copy}$, $G_{mult}$

First Expansion

Leakage probability $p$

Simulation Failure $\epsilon = f(p)$

Second Expansion

$\epsilon^2 = f^2(p)$

$G_{copy}$

$G_{add}$

$G_{mult}$

$G_{copy}^{(2)}$

$G_{add}^{(2)}$

$G_{mult}^{(2)}$

$n^k \rightarrow \cdots$
Random Probing Expandability
Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using \( n \)-share gadgets \( G_{add}, G_{copy}, G_{mult} \)

\[
\begin{align*}
\text{Leakage probability} & \quad p \\
\text{Simulation Failure} & \quad \epsilon = f(p) \\
\end{align*}
\]

Random Probing Expandability

Expansion Strategy (Revisited approach from [Ananth et al., 2018])

Using $n$-share gadgets $G_{\text{add}}$, $G_{\text{copy}}$, $G_{\text{mult}}$

Leakage probability $p$

Simulation Failure $\epsilon = f(p)$

Condition: $f(p) < p$
Random Probing Expandability

Expansion Security

2-to-1 3-share gadget
(\(t, \epsilon\))-Random Probing Expandability

\(W\) set of wires

Any set \(J\) of \(\leq t\) output wires

Can be simulated from \(t\) shares of 1 and 2?

Simulation Success  Simulation Failure

\(\text{Simulation Success}\) \quad \text{Simulation Failure}\n
Output

2-to-1 3-share gadget

Input 1  Input 2

\(+\)  \(\times\)  \(\oplus\)  \(\oplus\)
Random Probing Expandability

Expansion Security

\( (t, \epsilon) \)-Random Probing Expandability

\( W \) set of wires

Any set \( J \) of \( \leq t \) output wires

Can be simulated from \( t \) shares of 1 and 2?

Simulation Success: yes

Simulation Failure: no

Failure Proba. on 1 = \( \epsilon \)

Failure Proba. on 2 = \( \epsilon \)

Failure Proba. on 1 \( \land \) 2 = \( \epsilon^2 \)
(t, ϵ)-Random Probing Expandability

W set of wires
Any set J of ≤ t output wires
and for a chosen set J’ of n − 1 output wires

Can be simulated from t shares of 1 and 2?

Simulation Success

Simulation Failure

Failure Proba. on 1 = ϵ
Failure Proba. on 2 = ϵ
Failure Proba. on \(1 \wedge 2\) = \(\epsilon^2\)

2-to-1 3-share gadget
(t, \epsilon)-Random Probing Expandability

\textbf{W} set of wires

Any set \textbf{J} of \leq t output wires

and for a chosen set \textbf{J}' of n - 1 output wires

Can be simulated from t shares of 1 and 2 ?

Simulation Success Simulation Failure

\begin{align*}
\text{Failure Proba. on 1} & = \epsilon \\
\text{Failure Proba. on 2} & = \epsilon \\
\text{Failure Proba. on 1} \land 2 & = \epsilon^2
\end{align*}
Random Probing Expandability

Expansion Security

(t, \epsilon)\text{-Random Probing Expandability}

\[ W \text{ set of wires} \]
Any set \( J \) of \( \leq t \) output wires
and for a chosen set \( J' \) of \( n - 1 \) output wires

Can be simulated from \( t \) shares of 1 and 2?

Simulation Success

Simulation Failure

\begin{align*}
\text{Failure Proba. on 1} &= \epsilon \\
\text{Failure Proba. on 2} &= \epsilon \\
\text{Failure Proba. on 1} \land 2 &= \epsilon^2
\end{align*}
$(t, \epsilon)$-Random Probing Expandability

\( W \) set of wires
Any set \( J \) of \( \leq t \) output wires
and for a chosen set \( J' \) of \( n - 1 \) output wires
Can be simulated from \( t \) shares of 1 and 2?

Simulation Success

Simulation Failure

Failure Proba. on 1  \( = \epsilon \)
Failure Proba. on 2  \( = \epsilon \)
Failure Proba. on 1 \& 2  \( = \epsilon^2 \)
Random Probing Expandability

Expansion Security

For an $n$-share gadget $G$:

Random Probing Expandable, $\epsilon = f(p)$

$\Rightarrow G(k)$ Random Probing Expandable, $\epsilon' = f(k(p))$

Random Probing Composable, $\epsilon' = f(k(p))$

For a circuit $C$, using $G$ add, $G$ copy, $G$ mult:

Random Probing Expandable, $\epsilon = f(p)$

$\Rightarrow$ Compiled circuit $(p, 2f(k))$-Random Probing Secure

S. Belaid, JS. Coron, E. Prouff, M. Rivain, A. Taleb
For an n-share gadget $G$: 

\[
G \text{ Random Probing Expandable}, \epsilon = f(p) \Rightarrow G(k) \text{ Random Probing Expandable}, \epsilon' = f(k(p))
\]

For a circuit $C$, using $G_{\text{add}}$, $G_{\text{copy}}$, $G_{\text{mult}}$: 

\[
G_{\text{add}}, G_{\text{copy}}, G_{\text{mult}} \text{ Random Probing Expandable}, \epsilon = f(p) \Rightarrow \text{Compiled circuit } (p, 2.f(k)) - \text{Random Probing Secure}
\]
Random Probing Expandability

Expansion Security

For an n-share gadget $G$:

$G$ Random Probing Expandable, $\epsilon = f(p)$

$\implies$

$G^{(k)}$ Random Probing Expandable, $\epsilon' = f^k(p)$
Random Probing Expandability

Expansion Security

For an n-share gadget $G$:

\[ G \text{ Random Probing Expandable, } \epsilon = f(p) \]
\[ \implies \]
\[ G^{(k)} \text{ Random Probing Expandable, } \epsilon' = f^k(p) \]

\[ G \text{ Random Probing Expandable, } \epsilon = f(p) \]
\[ \implies \]
\[ G \text{ Random Probing Composable, } \epsilon' = 2f(p) \]
Random Probing Expandability

Expansion Security

For an n-share gadget $G$:

$G$ Random Probing Expandable, $\epsilon = f(p)$ \quad \Rightarrow \quad G^{(k)}$ Random Probing Expandable, $\epsilon' = f^k(p)$

$G$ Random Probing Expandable, $\epsilon = f(p)$ \quad \Rightarrow \quad G$ Random Probing Composable, $\epsilon' = 2f(p)$

For a circuit $C$, using $G_{add}$, $G_{copy}$, $G_{mult}$:
Random Probing Expandability

Expansion Security

For an n-share gadget $G$:

\[
G \text{ Random Probing Expandable, } \epsilon = f(p) \implies G^{(k)} \text{ Random Probing Expandable, } \epsilon' = f^k(p)
\]

\[
G \text{ Random Probing Expandable, } \epsilon = f(p) \implies G \text{ Random Probing Composable, } \epsilon' = 2f(p)
\]

For a circuit $C$, using $G_{add}$, $G_{copy}$, $G_{mult}$:

\[
G_{add}, G_{copy}, G_{mult} \text{ Random Probing Expandable, } \epsilon = f(p) \implies \text{Compiled circuit } (p, 2.f^{(k)})\text{-Random Probing Secure}
\]
Random Probing Expandability

3-share gadgets Construction

\[ G \text{copy} : v_0 \leftarrow x_0 + r_0 + r_1; w_0 \leftarrow x_0 + r_3 + r_4 \]
\[ v_1 \leftarrow x_1 + r_1 + r_2; w_1 \leftarrow x_1 + r_4 + r_5 \]
\[ v_2 \leftarrow x_2 + r_2 + r_0; w_2 \leftarrow x_2 + r_5 + r_3 \]

\[ G \text{add} : z_0 \leftarrow x_0 + r_0 + r_4 + y_0 + r_1 + r_3 \]
\[ z_1 \leftarrow x_1 + r_1 + r_5 + y_1 + r_2 + r_4 \]
\[ z_2 \leftarrow x_2 + r_2 + r_3 + y_2 + r_0 + r_5 \]

\[ G \text{mult} : u_0 \leftarrow x_0 + r_5 + r_6; u_1 \leftarrow x_1 + r_6 + r_7; u_2 \leftarrow x_2 + r_7 + r_5 \]
\[ v_0 \leftarrow y_0 + r_8 + r_9; v_1 \leftarrow y_1 + r_9 + r_{10}; v_2 \leftarrow y_2 + r_{10} + r_8 \]

\[ z_0 \leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2) \]
\[ z_1 \leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3) \]
\[ z_2 \leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4 \]
Random Probing Expandability

3-share gadgets Construction

\[ G_{\text{copy}} : v_0 \leftarrow x_0 + r_0 + r_1; \quad w_0 \leftarrow x_0 + r_3 + r_4 \]

\[ v_1 \leftarrow x_1 + r_1 + r_2; \quad w_1 \leftarrow x_1 + r_4 + r_5 \]

\[ v_2 \leftarrow x_2 + r_2 + r_0; \quad w_2 \leftarrow x_2 + r_5 + r_3 \]
3-share gadgets Construction

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

\[ G_{\text{add}} : z_0 \leftarrow x_0 + r_0 + r_4 + y_0 + r_1 + r_3 \]
\[ z_1 \leftarrow x_1 + r_1 + r_5 + y_1 + r_2 + r_4 \]
\[ z_2 \leftarrow x_2 + r_2 + r_3 + y_2 + r_0 + r_5 \]
Random Probing Expandability

3-share gadgets Construction

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

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

\(G_{\text{mult}}: u_0 \leftarrow x_0 + r_5 + r_6; \ u_1 \leftarrow x_1 + r_6 + r_7; \ u_2 \leftarrow x_2 + r_7 + r_5\)
\(v_0 \leftarrow y_0 + r_8 + r_9; \ v_1 \leftarrow y_1 + r_9 + r_{10}; \ v_2 \leftarrow y_2 + r_{10} + r_8\)
Random Probing Expandability

3-share gadgets Construction

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

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

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

\[ z_0 \leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2) \]
\[ z_1 \leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3) \]
\[ z_2 \leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4 \]
Random Probing Expandability

3-share gadgets Construction

\(G_{\text{copy}}: v_0 \leftarrow x_0 + r_0 + r_1; w_0 \leftarrow x_0 + r_3 + r_4\)

\(v_1 \leftarrow x_1 + r_1 + r_2; w_1 \leftarrow x_1 + r_4 + r_5\)

\(v_2 \leftarrow x_2 + r_2 + r_0; w_2 \leftarrow x_2 + r_5 + r_3\)

\(G_{\text{add}}: z_0 \leftarrow x_0 + r_0 + r_4 + y_0 + r_1 + r_3\)

\(z_1 \leftarrow x_1 + r_1 + r_5 + y_1 + r_2 + r_4\)

\(z_2 \leftarrow x_2 + r_2 + r_3 + y_2 + r_0 + r_5\)

\(G_{\text{mult}}: u_0 \leftarrow x_0 + r_5 + r_6; u_1 \leftarrow x_1 + r_6 + r_7; u_2 \leftarrow x_2 + r_7 + r_5\)

\(v_0 \leftarrow y_0 + r_8 + r_9; v_1 \leftarrow y_1 + r_9 + r_{10}; v_2 \leftarrow y_2 + r_{10} + r_8\)

\(z_0 \leftarrow (u_0 \cdot v_0 + r_0) + (u_0 \cdot v_1 + r_1) + (u_0 \cdot v_2 + r_2)\)

\(z_1 \leftarrow (u_1 \cdot v_0 + r_1) + (u_1 \cdot v_1 + r_4) + (u_1 \cdot v_2 + r_3)\)

\(z_2 \leftarrow (u_2 \cdot v_0 + r_2) + (u_2 \cdot v_1 + r_3) + (u_2 \cdot v_2 + r_0) + r_4\)

\(t = 1, \quad f(p) \leq \sqrt{83} p^{3/2} + O(p^2), \quad p_{\text{max}} \approx 2^{-8}\)
Random Probing Expandability

Asymptotic Complexity

\[ N = (N_{\text{add}}, N_{\text{copy}}, N_{\text{mult}}, N_{\text{rand}}) \]
Random Probing Expandability
Asymptotic Complexity

\[ N = (N_{\text{add}}, N_{\text{copy}}, N_{\text{mult}}, N_{\text{rand}}) \]

On previous 3-share gadgets:
Random Probing Expandability

Asymptotic Complexity

\[ N = (N_{add}, N_{copy}, N_{mult}, N_{rand}) \]

On previous 3-share gadgets:

\[
M = \begin{pmatrix}
N_{G_{add}}^T & N_{G_{copy}}^T & N_{G_{mult}}^T & N_{rand}^T \\
15 & 12 & 28 & 0 \\
6 & 9 & 23 & 0 \\
0 & 0 & 9 & 0 \\
6 & 6 & 11 & 3
\end{pmatrix}
\]
Random Probing Expandability
Asymptotic Complexity

\[ N = (N_{\text{add}}, N_{\text{copy}}, N_{\text{mult}}, N_{\text{rand}}) \]

On previous 3-share gadgets:

\[
M = \begin{pmatrix}
N_{G_{\text{add}}}^T & N_{G_{\text{copy}}}^T & N_{G_{\text{mult}}}^T & N_{\text{rand}}^T \\
15 & 12 & 28 & 0 \\
6 & 9 & 23 & 0 \\
0 & 0 & 9 & 0 \\
6 & 6 & 11 & 3
\end{pmatrix} = Q \Lambda Q^{-1}
\]
Random Probing Expandability

Asymptotic Complexity

\[ N = (N_{\text{add}}, N_{\text{copy}}, N_{\text{mult}}, N_{\text{rand}}) \]

On previous 3-share gadgets:

\[
M = \begin{pmatrix}
N_{G_{\text{add}}}^T & N_{G_{\text{copy}}}^T & N_{G_{\text{mult}}}^T & N_{\text{rand}}^T \\
15 & 12 & 28 & 0 \\
6 & 9 & 23 & 0 \\
0 & 0 & 9 & 0 \\
6 & 6 & 11 & 3
\end{pmatrix} = Q \Lambda Q^{-1}
\]

Compiling a circuit \( C \): \( N_{\hat{C}} = M^k N_C = Q \Lambda^k Q^{-1} N_C \)
Random Probing Expandability

Asymptotic Complexity

\[ N = (N_{add}, N_{copy}, N_{mult}, N_{rand}) \]

On previous 3-share gadgets:

\[
M = \begin{pmatrix}
  N^T_{G_{add}} & N^T_{G_{copy}} & N^T_{G_{mult}} & N^T_{rand} \\
  M_{ac} & 28 & 23 & 0 \\
  0 & 0 & N_m & 0 \\
  6 & 6 & 1 & 3
\end{pmatrix} = Q \cdot \Lambda \cdot Q^{-1}
\]

Compiling a circuit \( C \): \( \hat{N}_C = M^k N_C = Q \cdot \Lambda^k \cdot Q^{-1} \cdot N_C \)
\[ N = (N_{\text{add}}, N_{\text{copy}}, N_{\text{mult}}, N_{\text{rand}}) \]

On previous 3-share gadgets:

\[
M = \begin{pmatrix}
N_{G_{\text{add}}}^T & N_{G_{\text{copy}}}^T & N_{G_{\text{mult}}}^T & N_{\text{rand}}^T \\
28 & 0 & N_{m} & 0 \\
0 & 0 & 1 & 3
\end{pmatrix}
= Q \cdot \Lambda \cdot Q^{-1}
\]

Compiling a circuit \( C \):

\[
N_{\hat{C}} = M^k N_C = Q \cdot \Lambda^k \cdot Q^{-1} \cdot N_C
\]

\[
|\hat{C}| = \mathcal{O}(|C| \cdot N_{\text{max}}^k), \quad N_{\text{max}} = \max \left( \text{eigenvalues}(M_{\text{ac}}), N_{\text{mult}} \right)
\]
For a security parameter $\kappa$, and $f(p) = c_d p^d + O(p^{d+1})$ of amplification order $d$, 

For a security parameter $\kappa$, and $f(p) = c_d p^d + \mathcal{O}(p^{d+1})$ of amplification order $d$, we need $f^{(k)}(p) \leq 2^{-\kappa}$.
For a security parameter $\kappa$, and $f(p) = cdp^d + O(p^{d+1})$ of amplification order $d$, we need $f^{(k)}(p) \leq 2^{-\kappa}$:

$$|\hat{C}| = O(|C|\kappa^e), \quad e = \frac{\log(N_{\text{max}})}{\log(d)}$$
For a security parameter $\kappa$, and $f(p) = \sqrt{83}p^{3/2} + O(p^2)$ of amplification order $3/2$, we need $f^{(k)}(p) \leq 2^{-\kappa}$:

$$|\hat{C}| = O(|C|.\kappa^{7.5}), \quad e = \frac{\log(21)}{\log(3/2)}$$
### Random Probing Expandability

**Comparison with [Ananth et al., 2018]**

<table>
<thead>
<tr>
<th>Our Expansion Strategy</th>
<th>[Ananth et al., 2018] Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**O**($|C| \cdot \kappa$)

$p_{\text{max}} \approx 2^{-8}$

**Strategy**

$m = 5$, $c = 2$
## Our Expansion Strategy

$(t, f)$-RPE Security

## [Ananth et al., 2018] Strategy

$(p, \epsilon)$-Composable Security
## Random Probing Expandability

Comparison with [Ananth et al., 2018]

<table>
<thead>
<tr>
<th>Our Expansion Strategy</th>
<th>[Ananth et al., 2018] Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(t, f)$-RPE Security</td>
<td>$(p, \epsilon)$-Composable Security</td>
</tr>
<tr>
<td>Secure $(t, f)$-RPE gadgets</td>
<td>$(m, c)$-MPC protocols</td>
</tr>
</tbody>
</table>

$O\left(|\mathcal{C}|.\kappa^7.5\right)$  
$p_{\text{max}} \approx 2^{-8}$

$O\left(|\mathcal{C}|.\kappa^7.87\right)$  
$p_{\text{max}} \approx 2^{-25}$
Random Probing Expandability
Comparison with [Ananth et al., 2018]

<table>
<thead>
<tr>
<th>Our Expansion Strategy</th>
<th>[Ananth et al., 2018] Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(t, f)$-RPE Security</td>
<td>$(p, \epsilon)$-Composable Security</td>
</tr>
<tr>
<td>Secure $(t, f)$-RPE gadgets</td>
<td>$(m, c)$-MPC protocols</td>
</tr>
<tr>
<td>Instantiation with $(1, f)$-RPE 3-share $G_{\text{add}}$, $G_{\text{copy}}$, $G_{\text{mult}}$</td>
<td>Instantiation with [Maurer, 2006] $(m = 5, c = 2)$-MPC protocol</td>
</tr>
</tbody>
</table>
# Random Probing Expandability

Comparison with [Ananth et al., 2018]

## Our Expansion Strategy

- $(t, f)$-RPE Security
  - Secure $(t, f)$-RPE gadgets
  - Instantiation with $(1, f)$-RPE
  - $3$-share $G_{add}$, $G_{copy}$, $G_{mult}$
  - $\mathcal{O}(|C|.\kappa^{7.5})$

## [Ananth et al., 2018] Strategy

- $(p, \epsilon)$-Composable Security
  - $(m, c)$-MPC protocols
  - Instantiation with [Maurer, 2006]
  - $(m = 5, \ c = 2)$-MPC protocol
  - $\mathcal{O}(|C|.\kappa^{7.87})$
## Random Probing Expandability

Comparison with [Ananth et al., 2018]

<table>
<thead>
<tr>
<th>Our Expansion Strategy</th>
<th>[Ananth et al., 2018] Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(t, f)$-RPE Security</td>
<td>$(p, \epsilon)$-Composable Security</td>
</tr>
<tr>
<td>Secure $(t, f)$-RPE gadgets</td>
<td>$(m, c)$-MPC protocols</td>
</tr>
<tr>
<td>Instantiation with $(1, f)$-RPE 3-share $G_{add}$, $G_{copy}$, $G_{mult}$</td>
<td>Instantiation with [Maurer, 2006] $(m = 5, c = 2)$-MPC protocol</td>
</tr>
<tr>
<td>$O(</td>
<td>C</td>
</tr>
<tr>
<td>$p_{\text{max}} \approx 2^{-8}$</td>
<td>$p_{\text{max}} \approx 2^{-25}$</td>
</tr>
</tbody>
</table>
• **VRAPS** tool for verification of Random Probing Security: https://github.com/CryptoExperts/VRAPS
Conclusion

- **VRAPS** tool for verification of Random Probing Security: https://github.com/CryptoExperts/VRAPS
- New gadget composition / expansion properties for random probing security
Conclusion

- **VRAPS** tool for verification of Random Probing Security: https://github.com/CryptoExperts/VRAPS

- New gadget composition / expansion properties for random probing security

- New 3-share construction achieving random probing security with tolerated leakage proba. \( \approx 2^{-8} \), and a complexity of \( O(|C| \cdot \kappa^{7.5}) \)
Conclusion

- **VRAPS** tool for verification of Random Probing Security: https://github.com/CryptoExperts/VRAPS

- New gadget composition / expansion properties for random probing security

- New 3-share construction achieving random probing security with tolerated leakage proba. $\approx 2^{-8}$, and a complexity of $O(|C| \cdot \kappa^{7.5})$

- Implementation of the expansion strategy, and an implementation of a secure $n^k$-share AES128: https://github.com/CryptoExperts/poc-expanding-compiler


