

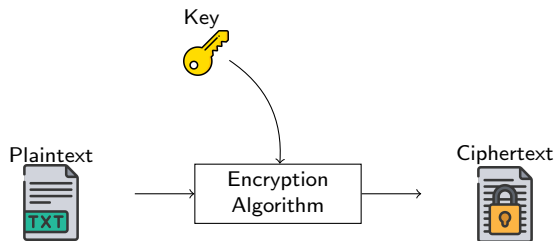
IronMaskArithmetic : Comprehensive Verification of Arithmetic Masking Security

Victor Normand

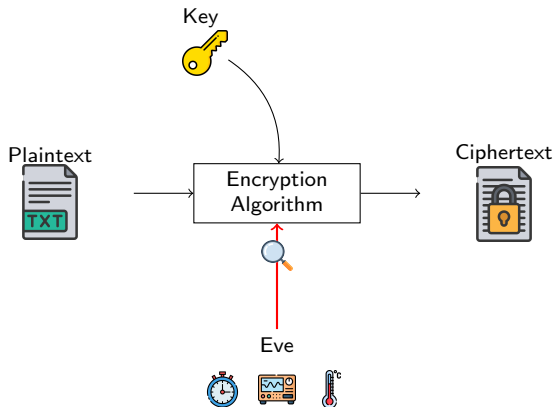
CryptoExperts

April 2025

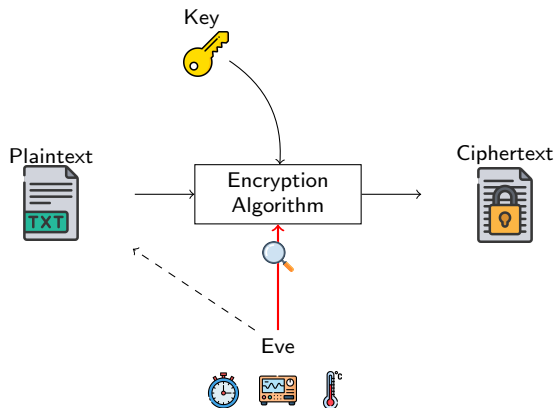
Side Channel Attacks



Side Channel Attacks



Side Channel Attacks



Masking countermeasure

- ▶ Replace a sensitive value x of the Algorithm by a tuple $(x_1, \dots, x_n)$.
- ▶ Any set of $(n - 1)$ shares is independent from x .
- ▶ Arithmetic masking :

$$x = \sum_{i=1}^n x_i$$

- ▶ Operations on sensitive values $\implies$ Operations on their respective shares.
- ▶ Masking with $n = 2$,

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$$

$$\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$$

- ▶ Addition without masking

$$\mathbf{z} \leftarrow \mathbf{x} + \mathbf{y}$$

Gadget Addition

$$\mathbf{z}_1 \leftarrow \mathbf{x}_1 + \mathbf{y}_1$$

$$\mathbf{z}_2 \leftarrow \mathbf{x}_2 + \mathbf{y}_2$$

$$\mathbf{z} \leftarrow \mathbf{z}_1 + \mathbf{z}_2$$

- ▶ Operations on sensitive values $\implies$ Operations on their respective shares.
- ▶ Masking with $n = 2$,

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$$

$$\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$$

- ▶ Multiplication without masking

$$\mathbf{z} \leftarrow \mathbf{x} \cdot \mathbf{y}$$

Gadget Multiplication

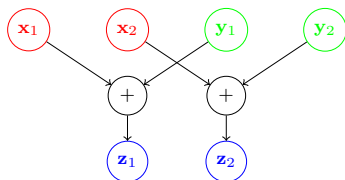
$$r \xleftarrow{\$} \mathbb{K}$$

$$\mathbf{z}_1 \leftarrow \mathbf{x}_1 \cdot \mathbf{y}_1 + r + \mathbf{x}_1 \cdot \mathbf{y}_2$$

$$\mathbf{z}_2 \leftarrow \mathbf{x}_2 \cdot \mathbf{y}_1 - r + \mathbf{x}_2 \cdot \mathbf{y}_2$$

$$\mathbf{z} \leftarrow \mathbf{z}_1 + \mathbf{z}_2$$

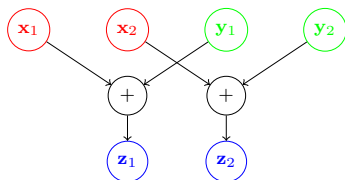
Circuits



▶ Circuits

- ▶ Directed acyclic graph.
- ▶ Each vertex is an input/output share or arithmetic gate.
- ▶ Each edge is a wire of the circuit.

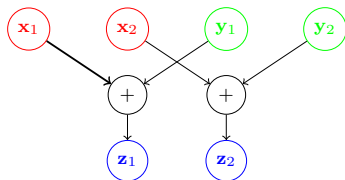
t-probing model



▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independant from the secrets.

t-probing model



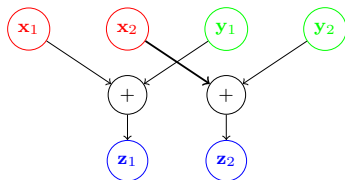
▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independent from the secrets.

Example : 1-probing Secure ?

- ▶ Leaks the value of x_1 .
- ▶ Don't leak the value of $x_2 \implies$ independent from x .
- ▶ Don't leak shares values of $y \implies$ independent from y .

t-probing model



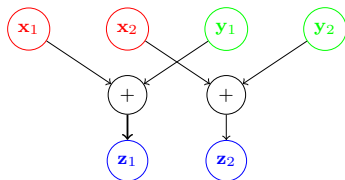
▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independent from the secrets.

Example : 1-probing Secure ?

- ▶ Leaks the value of x_2 .
- ▶ Don't leak the value of $x_1 \implies$ independent from x .
- ▶ Don't leak shares values of $y \implies$ independent from y .

t-probing model



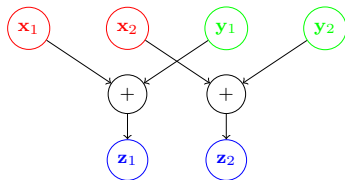
▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independent from the secrets.

Example : 1-probing Secure ?

- ▶ Leaks the value of x_1 and y_1 .
- ▶ Don't leak the value of $x_2 \implies$ independent from x .
- ▶ Don't leak the value of $y_2 \implies$ independent from y .

t-probing model



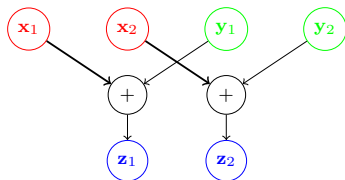
▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independant from the secrets.

Example : 1-probing Secure ?

- ▶ All combination of 1 wire $\implies$ independent from the secrets.
- ▶ The circuit is 1-probing secure.

t-probing model



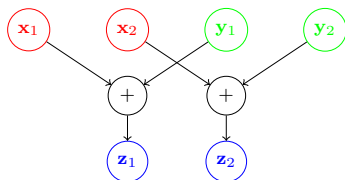
▶ t-Probing Model

- ▶ Up to t wires of the circuits leak its value.
- ▶ Secure if any leakages sets is independant from the secrets.

Example : 2-probing Secure ?

- ▶ Leaks the values of x_1 and x_2 .
- ▶ $x = x_1 + x_2 \implies$ we can find x from the values of x_1 and x_2 .
- ▶ The circuit is not 2-probing secure.

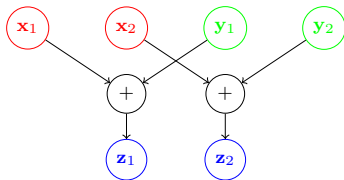
(p, ε) -random-probing-model



▶ (p, ε) -Random-Probing Model

- ▶ Every wires leak its value with a probability $p \in [0, 1]$.
- ▶ Secure if there is a negligible probability ($\leq \varepsilon$) to get information on the secrets.

(p, ε) -random-probing-model



► (p, ε) -Random-Probing Model

- Every wires leak its value with a probability $p \in [0, 1]$.
- Secure if there is a negligible probability ($\leq \varepsilon$) to get information on the secrets.

$$f(p) := \sum_{i=1}^{|C|} c_i p^i (1-p)^{|C|-i} \leq \varepsilon$$

$c_i \implies$ Failure Tuples of i wires.

Random Probing Security :

$$f(p) = \sum_{i=1}^{|C|} c_i p^i (1-p)^{|C|-i}$$

$c_i \implies$ Failure Tuples of i wires.

- ▶ Verification exponential in the size of the circuit.
- ▶ Automatic Tools $\implies$ faster security verification.

- ▶ IronMask : Formal tool to verify the security in the t -probing model and (p, ε) -random probing model.
- ▶ Circuits on the boolean field $\mathbb{F}_2$.
- ▶ Comprehensive : Compute each coefficient c_i .

- ▶ IronMask : Formal tool to verify the security in the t -probing model and (p, ε) -random probing model.
 - ▶ Circuits on the boolean field $\mathbb{F}_2$.
 - ▶ Comprehensive : Compute each coefficient c_i .
-
- ▶ IronMaskArithmetic: An extension of the formal tool IronMask .
 - ▶ Circuits on any arithmetic field $\mathbb{F}_q$ with q prime.

Parsing gadgets

```
#SHARES 2
#IN a b
#RANDOMS r01 rr01
#OUT e
```

```
c0 = a0 + r01
c1 = a1 + r01
```

```
d0 = b0 + rr01
d1 = b1 + rr01
```

```
e0 = c0 + d0
e1 = c1 + d1
```

(a) Parsing with IronMask

```
#SHARES 2
#IN a b
#RANDOMS r01 rr01
#OUT e
#CAR 3329
```

```
c0 = a0 + 2 r01
c1 = a1 + -2 r01
```

```
d0 = b0 + 3 rr01
d1 = b1 + -3 rr01
```

```
e0 = c0 + d0
e1 = c1 + d1
```

(b) Parsing with IronMaskArithmetic

IronMask - Exhaustive Approach

- ▶ To find failure tuples of size i :
 - ▶ Consider a tuple.
 - ▶ Determine if it is a failure tuple.
 - ▶ Reitere for all the tuples of size i .
- ▶ $\binom{|C|}{i}$ tuples to evaluate.

IronMaskArithmetic- Constructive Approach - Incompressible tuple

- ▶ q_1, q_2, q_3 leaked values of the circuit.
- ▶ (q_1, q_2, q_3) is a failure tuple.
- ▶ All sub tuples of (q_1, q_2, q_3) are not failure tuples.

Not a failure tuple

q_1, q_2

Not a failure tuple

q_1, q_3

Failure tuple

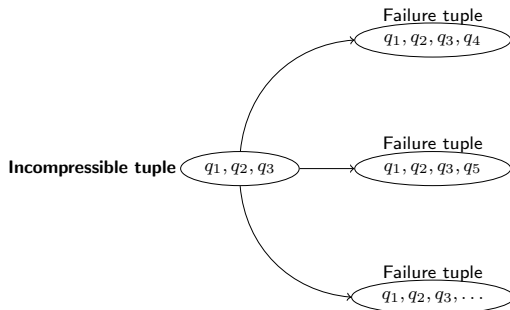
q_1, q_2, q_3

Not a failure tuple

q_2, q_3

Figure: Illustration of an incompressible tuple

Constructive Approach - Incompressible tuple



Constructive approach - Example

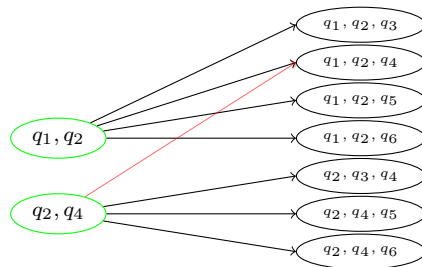
Example:

- ▶ 6 wires.
- ▶ 3 incompressible failure tuples : (q_1, q_2) , (q_2, q_4) and (q_4, q_5, q_6)
- ▶ $c_3 = ?$

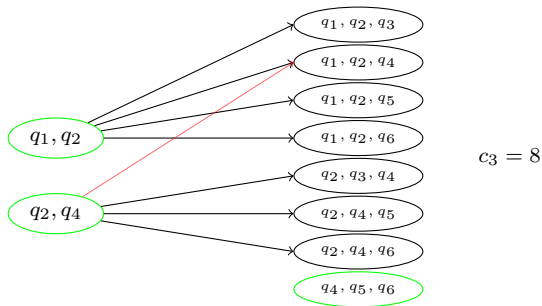
q_1, q_2

q_2, q_4

Constructive approach - Example



Constructive approach - Example



Timing performance

	IronMask (Car 2)	IronMaskArithmetic(Car 3329)
Add_3_shares RP -c 4	0s	0s
Add_7_shares SNI -t 6	8s	4s
Copy_6_shares RPC -c 5 -t 2	1min 20s	1 min 9s
Mult_6_shares NI -t 5	1s	4min 7s
Mult-ref_5_shares RP -c 4	1s	10s

Thanks for your attention