

# IronMask

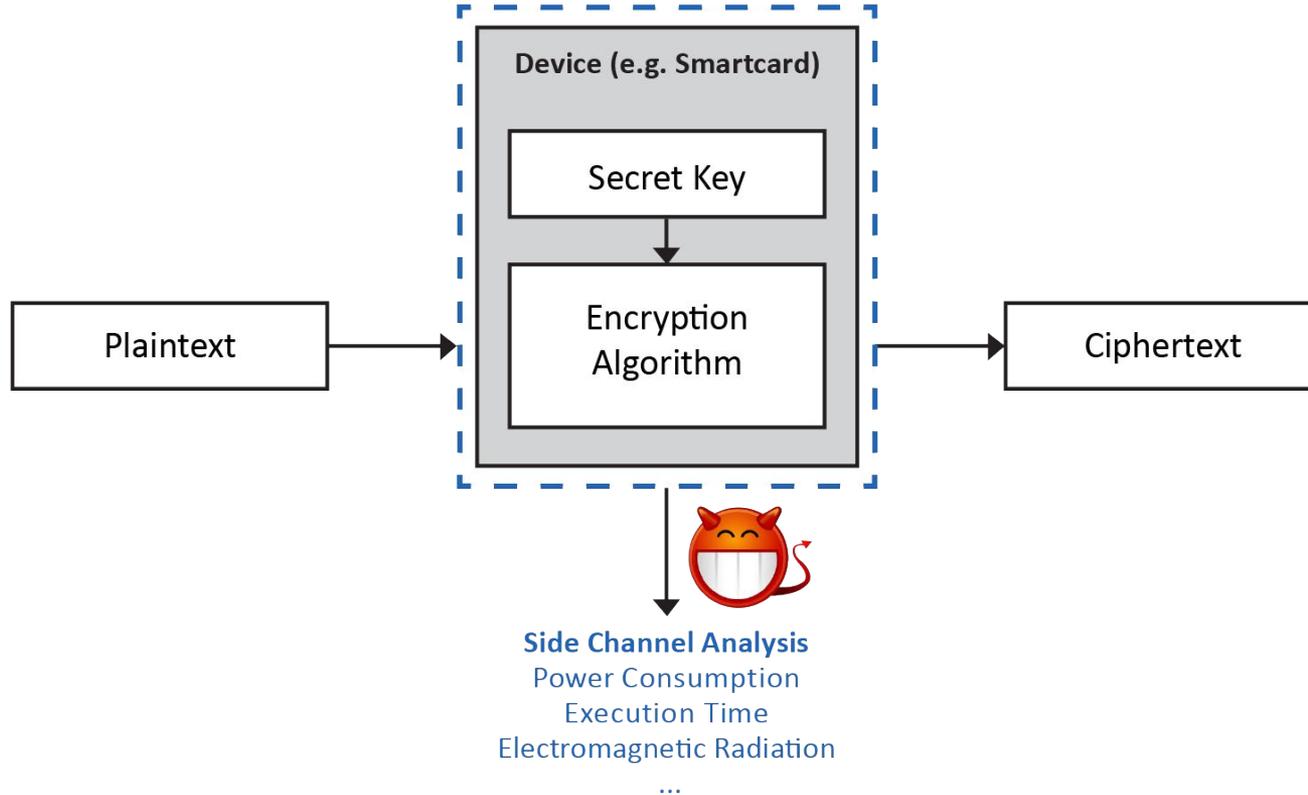
## Versatile Verification of Masking Security

Sonia Belaïd     Darius Mercadier  
Matthieu Rivain     Abdel Taleb

CryptoExperts, France

43rd IEEE Symposium on Security and Privacy  
June 24, 2022

# Introduction - Side-Channel Attacks



# Introduction - Masking

Sensitive variable  $\mathcal{X}$

$$\mathcal{X} = x_0 \oplus \dots \oplus x_{n-1}$$

All  $n$  shares are needed to recover the value of  $\mathcal{X}$

# Leakage Models

- $t$ -Probing model

$t$  variables leak during an execution

- Random probing model

each variable leaks with fixed probability  $p$

- Noisy leakage model

each variable leaks a noisy function of its value

# Automatic Verification Tools

- Check security/composition properties for relatively small circuits/gadgets
- Determine which variable reveals information about the secret shares
- Conclude depending on the constraints of the properties (probing-like, random probing-like, ...)

# Contributions

- Formalisation of all of the probing and random probing properties from a single common building block.
- Exact verification method for almost all circuits/gadgets in the state-of-the-art, using an extension of the algebraic characterization introduced in [EC:BBPPTV16,C:BBPPTV17] of gadgets with linear randomness (all random values are additive) to more general gadgets with non-linear randomness (e.g multiplication gadgets with input refreshing) like [CHES:BCPZ16,EC:BelRivTal21].
- IronMask: an **exact** verification tool for **all** probing and random probing properties

# Building Block for Security Properties

Simulation based definitions of all (random) probing properties

Given a set of probes  $P$  + a set of output shares  $O$ , determine the set of input shares  $I$  necessary for a perfect simulation

Example: set of variables  $\{a_0 + r_0, a_1 + r_1, a_2\}$ , uniform random values  $r_0, r_1$

Input share  $a_2$  is necessary for a perfect distribution simulation

# Building Block for Security Properties

Simulation based definitions of all (random) probing properties

Given a set of probes  $P$  + a set of output shares  $O$ , determine the set of input shares  $I$  necessary for a perfect simulation

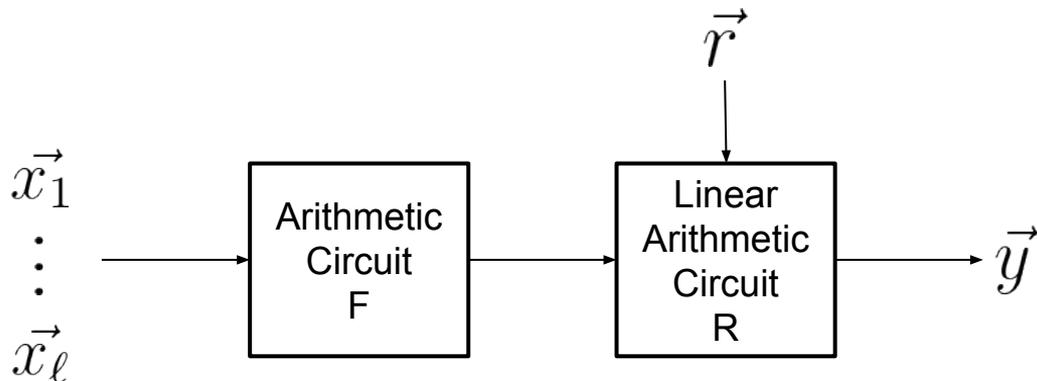
**SIS:** a function which outputs the smallest of the sets  $I$

For a  $n$ -share  $\ell$ -input 1-output gadget  $G$

$$SIS_G(P, O) = (I_1, \dots, I_\ell)$$

# Algebraic Characterization of Gadgets

Gadgets with linear randomness



Probe  $p$  on such gadgets

$$p = f_p(\vec{x}_1, \dots, \vec{x}_\ell) + \vec{r}^T \cdot \vec{s}_p$$

# Algebraic Characterization of Gadgets

Gadgets with linear randomness: exact verification

$$p = f_p(\vec{x}_1, \dots, \vec{x}_\ell) + \vec{r}^T \cdot \vec{s}_p$$

Set of probes  $\vec{P} = (p_1, \dots, p_d)$

$$S = \begin{pmatrix} \vec{s}_{p_1} \\ \vdots \\ \vec{s}_{p_d} \end{pmatrix} \xrightarrow{\text{row reduction}} S' = N \cdot S = \begin{pmatrix} 0_{m,d-m} & 0_{m,R-d+m} \\ I_{d-m} & S'' \end{pmatrix}$$

$$\vec{P}' = N \cdot \vec{P} = (p'_1, \dots, p'_m, p'_{m+1}, \dots, p'_d)$$

# Algebraic Characterization of Gadgets

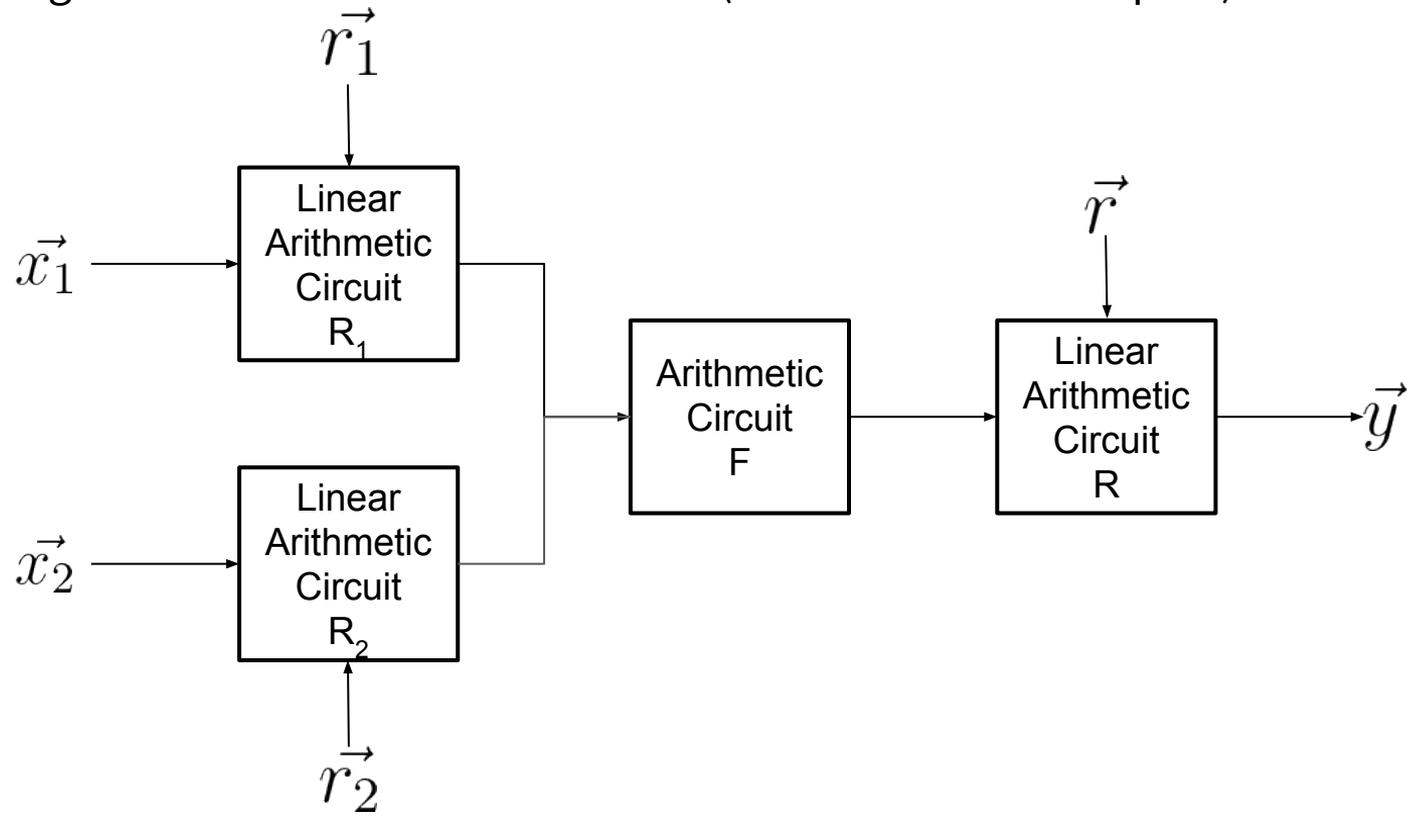
$$\vec{P}' = N \cdot \vec{P} = (p'_1, \dots, p'_m, p'_{m+1}, \dots, p'_d)$$

$(p'_{m+1}, \dots, p'_d)$  can be simulated from uniform random values

Input shares necessary to simulate  $\vec{P}$  are the same to simulate  $(p'_1, \dots, p'_m)$

# Algebraic Characterization of Gadgets

Gadgets with non-linear randomness (illustration with 2 inputs)



# Algebraic Characterization of Gadgets

Gadgets with non-linear randomness

Probe  $p$  on such gadgets

$$p = f_p(R_1(\vec{x}_1, \vec{r}_1), R_2(\vec{x}_2, \vec{r}_2)) + \vec{r}^T \cdot \vec{s}_p$$

Perform three Gaussian Eliminations

- First with respect to  $\vec{r}$
- Then with respect to  $\vec{r}_1$  and  $\vec{r}_2$

**Proven Result:** the strategy is an exact verification method for such gadgets

# IronMask

- C Implementation of the new exact verification technique
- Implements state-of-the-art verification optimizations
- Represents variable dependence as integer arrays to accelerate operations

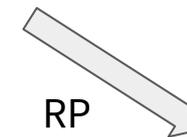
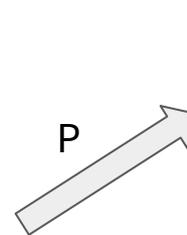
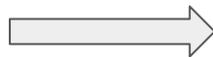
# IronMask - overview

```
#shares 2
#in a b
#randoms r0
#out c
```

```
m0 = a0 * b1
t0 = r0 + m0
m1 = a1 * b0
t1 = t0 + m1
```

```
m2 = a0 * b0
c0 = m2 + r0
```

```
m3 = a1 * b1
c1 = m3 + t1
```



$f(p)$

Counter-example

$t\text{-NI} / t\text{-SNI} / t\text{-PINI}$

```
$ ./ironmask isw-2-shares.sage SNI -t 1
```

# Evaluation - Scope

✓ handled  
✓ handled but inexact  
✗ not handled  
P: Probing  
RP: Random Probing

Tool	Properties	Supported gadgets			Fast verification
		Linear randomness	Non-linear randomness with 2 inputs	Non-linear randomness with > 2 inputs	
SILVER	P	✓	✓	✓	✗
Maskverif	P	✓	✓	✓	✓
Matverif	P	✓	✗	✗	✓
VRAPS	RP	✓	✓	✓	✗
IronMask	P, RP	✓	✓	✗	✓

# Performance - Probing

Competitive with the fastest verification tools for probing-like properties

# Performance - Random Probing

<b>Gadget</b>	<b>Verification time</b>	
	<b>IronMask</b>	<b>VRAPS</b>
ISW mult 5 shares	3sec	1h 15min
ISW mult 6 shares	17sec	24h
ISW mult 7 shares	24sec	24h

# Conclusion

- Formalization of all (random) probing properties in the state-of-the-art
- Algebraic Characterization of gadgets with (non-)linear randomness
- Exact proven verification methods
- IronMask
  - Exact for linear and non-linear randomness with 2 inputs
  - Probing: similar performance as other fast verification tools
  - Random probing: much faster than VRAPS