On the Practical Security of a Leakage Resilient Masking Scheme

T. Roche
thomas.roche@ssi.gouv.fr
Joint work with E. Prouff and M. Rivain

CT-RSA 2014 – Feb. 2014

Side Channel Analysis

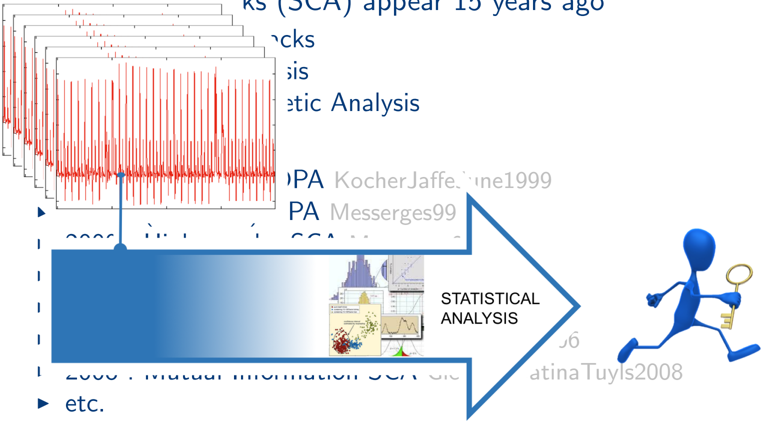
- Side Channel Attacks (SCA) appear 15 years ago
  - 1996 : Timing Attacks
  - 1998 : Power Analysis
  - 2000 : Electromagnetic Analysis
- Numerous attacks
  - 1998 : (single-bit) DPA KocherJaffeJune1999
  - 1999 : (multi-bit) DPA Messerges99
  - 2000 : Higher-order SCA Messerges2000
  - 2002 : Template SCA ChariRaoRohatgi2002
  - 2004 : CPA BrierClavierOlivier2004
  - 2005 : Stochastic SCA SchindlerLemkePaar2006
  - 2008 : Mutual Information SCA GierlichsBatinaTuyls2008
  - etc.

Side Channel Analysis

- Side Channel Attacks (SCA) appear 15 years ago
  - 1996 : Timing Attacks
  - 1998 : Power Analysis
  - 2000 : Electromagnetic Analysis
- Numerous attacks
  - 1998 : (single-bit) DPA KocherJaffeJune1999
  - 1999 : (multi-bit) DPA Messerges99
  - 2000 : Higher-order SCA Messerges2000
  - 2002 : Template SCA ChariRaoRohatgi2002
  - 2004 : CPA BrierClavierOlivier2004
  - 2005 : Stochastic SCA SchindlerLemkePaar2006
  - 2008 : Mutual Information SCA GierlichsBatinaTuyls2008
  - etc.

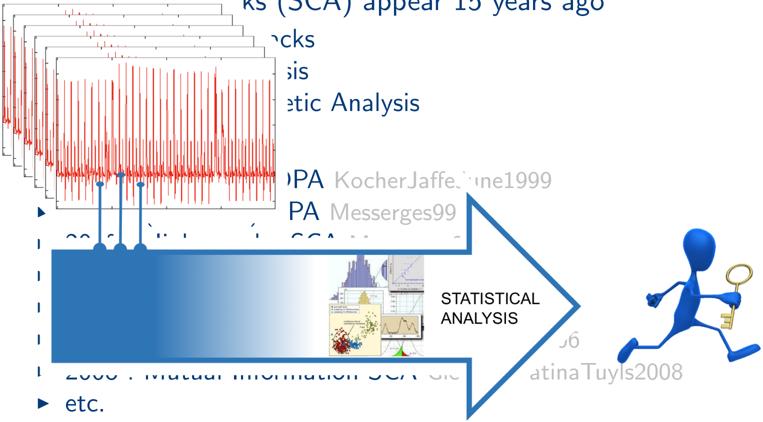
Side Channel Analysis

- ks (SCA) appear 15 years ago
  - cks
  - sis
  - etic Analysis
-
  - PA KocherJaffe une1999
  - PA Messerges99
  - 6
  - atinaTuyls2008
  - etc.



Side Channel Analysis

- …ks (SCA) appear 15 years ago
  - …cks
  - …sis
  - …etic Analysis
-
  - …PA KocherJaffe…une1999
  - PA Messerges99
  - …6
  - …atinaTuyls2008
  - etc.



Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

[Chari-Jutla-Rao-Rohatgi CRYPTO'99]

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $((L_i)_i | x = 0) = ((L_i)_i | x = 1)$:

$$q \geq O(1)\sigma^d$$

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

[Chari-Jutla-Rao-Rohatgi CRYPTO'99]

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $\left((L_i)_i | x = 0\right) = \left((L_i)_i | x = 1\right)$:

$$q \geq O(1)\sigma^d$$

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

[Chari-Jutla-Rao-Rohatgi CRYPTO'99]

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $\big((L_i)_i | x = 0\big) = \big((L_i)_i | x = 1\big)$:

$$q \geq O(1)\sigma^d$$

Probing Adversary

- Notion introduced in IshaiSahaiWagner, CRYPTO 2003
- A d^{th}-order probing adversary is allowed to observe **at most** d intermediate results during the overall algorithm processing.
  - Hardware interpretation : d is the maximum of wires observed in the circuit.
  - Software interpretation : d is the maximum of different timings during the processing.
- d^{th}-order probing adversary $=$ d^{th}-order SCA as introduced in Messerges99.
- Countermeasures proved to be secure against a d^{th}-order probing adv. :
  - $d = 1$: KocherJaffeJune99, BlömerGuajardoKrummel04, ProuffRivain07.
  - $d = 2$: RivainDottaxProuff08.
  - $d \geq 1$: IshaiSahaiWagner03, ProuffRoche11, GenelleProuffQuisquater11, CarletGoubinProuffQuisquaterRivain12.

Probing Adversary

- Notion introduced in IshaiSahaiWagner, CRYPTO 2003
- A d^{th}-order probing adversary is allowed to observe **at most** d intermediate results during the overall algorithm processing.
  - Hardware interpretation : d is the maximum of wires observed in the circuit.
  - Software interpretation : d is the maximum of different timings during the processing.
- d^{th}-order probing adversary $=$ d^{th}-order SCA as introduced in Messerges99.
- Countermeasures proved to be secure against a d^{th}-order probing adv. :
  - $d = 1$: KocherJaffeJune99, BlömerGuajardoKrummel04, ProuffRivain07.
  - $d = 2$: RivainDottaxProuff08.
  - $d \geq 1$: IshaiSahaiWagner03, ProuffRoche11, GenelleProuffQuisquater11, CarletGoubinProuffQuisquaterRivain12.

Higher-Order Masking Schemes

Achieving security in the probing adversary model

Definition

A *dth-order masking scheme* for an encryption algorithm $c \leftarrow \mathcal{E}(m, k)$ is an algorithm

$$(c_0, c_1, \ldots, c_d) \leftarrow \mathcal{E}'\big((m_0, m_1, \ldots, m_d), (k_0, k_1, \ldots, k_d)\big)$$

- Completeness : there exists R s.t. :

$$R(c_0, \cdots, c_d) = \mathcal{E}(m, k)$$

- Security : $\forall \{iv_1, iv_2, \ldots, iv_d\} \subseteq \{\text{intermediate var. of } \mathcal{E}'\}$:

$$\Pr\big(k \mid iv_1, iv_2, \ldots, iv_d\big) = \Pr\big(k\big)$$

State Of The Art

dth-order masking schemes

- Boolean Masking $n = 2d + 1, O(d^2)$
  [Ishai *et al.03]* *(hardware oriented)*
  $\hookrightarrow$ *[Rivain-Prouff 10] [Kim* et al.11]
- Multiplicative Masking $n = d + 1, O(d^2)$
  [Genelle *et al.11]*
  (alternating Boolean and Multiplicative Masking)
- Polynomial Masking $\widetilde{O}(d^2)$
  [Prouff-Roche 11] ($n = 2d + 1$, Glitches Resitance)
- Inner-Product Masking $O(d^2)$
  [Balasch *et al.12*] ($n = 2(d + 1)$, Glitches Resistance)

State Of The Art

dth-order masking schemes

- Boolean Masking $n = 2d + 1, O(d^2)$
  [Ishai *et al.03]* *(hardware oriented)*
  $\hookrightarrow$ *[Rivain-Prouff 10] [Kim* et al.11]
- Multiplicative Masking $n = d + 1, O(d^2)$
  [Genelle *et al.11]*
  (alternating Boolean and Multiplicative Masking)
- Polynomial Masking $\widetilde{O}(d^2)$
  [Prouff-Roche 11] ($n = 2d + 1$, Glitches Resitance)
- Inner-Product Masking $O(d^2)$
  [Balasch *et al.12*] ($n = 2(d + 1)$, Glitches Resistance)

Mutual Information Evaluation

Hamming Weight Model and Additive Gaussian Noise

$$\mathcal{O}(X) = HW(X) + \mathcal{B}$$

$\mathcal{B} \leftarrow \mathcal{N}(0, \sigma)$

In this idealized model, the success rate of an optimal multi-query (HO-)SCA targeting $(Z_0, \cdots, Z_d)$ is a monotonously increasing function of

$$\mathcal{I}(\mathcal{O}(Z_0), \cdots, \mathcal{O}(Z_d); Z)$$

[Standaert *et al.* 09]

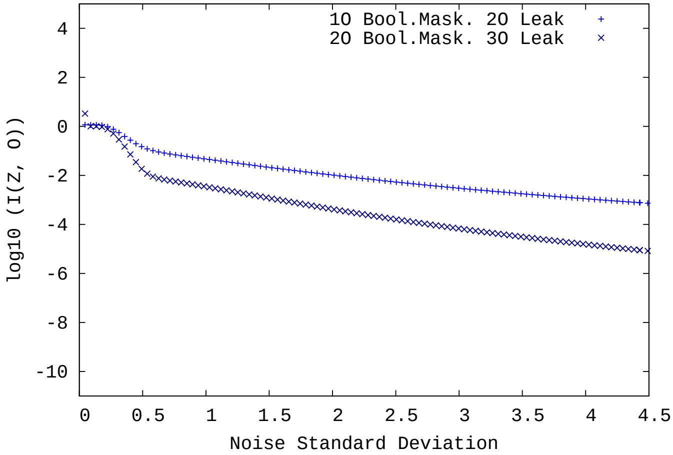
Boolean Sharing

Manipulation of randomized variable

$$z \xrightarrow{\$} (z \oplus r_1 \oplus \cdots \oplus r_d, r_1, \cdots, r_d) \ ,$$

where r_i are randomly generated in $\mathrm{GF}(2^\ell)$.

Information Leaked by a d^{th}-order Boolean Sharing



Multiplicative Sharing

Manipulation of randomized variable

$$z \xrightarrow{\$} (z \mathbf{x} r_1 \mathbf{x} \cdots \mathbf{x} r_d, r_1, \cdots, r_d) \ ,$$

where r_i are randomly generated in $\mathrm{GF}^*(2^\ell)$.

Multiplicative Sharing

Manipulation of randomized variable

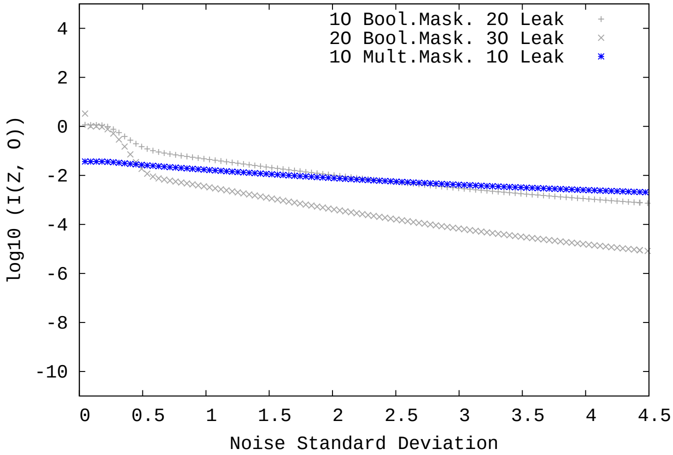
$$z \xrightarrow{\$} (z \mathbf{x} r_1 \mathbf{x} \cdots \mathbf{x} r_d, r_1, \cdots, r_d) \ ,$$

where r_i are randomly generated in $\mathrm{GF}^*(2^\ell)$.

1st-order Flaw

$$\Pr(z \mathbf{x} r_1 \mathbf{x} \cdots \mathbf{x} r_d = 0 | z = 0) = 1$$

Information Leaked by a d^{th}-order Multiplicative Sharing



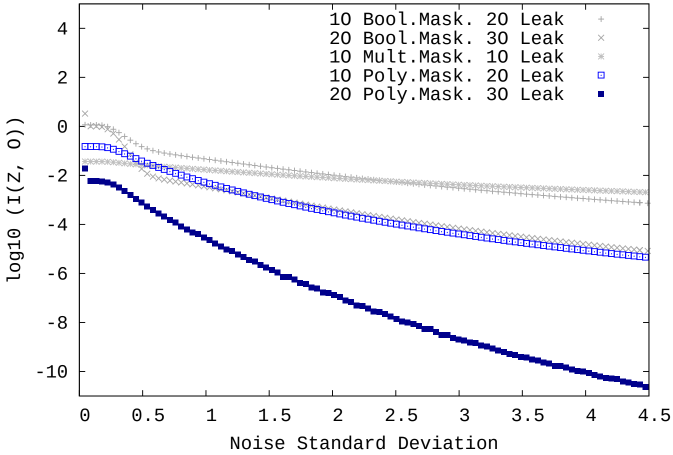
Shamir's Secret Sharing

Manipulation of randomized variable

$$\begin{array}{rcl} z & \xrightarrow{\$} & P_z[X] : z + a_1 X + \cdots + a_d X^d \\ P_z, \alpha_1, \cdots, \alpha_n & \rightarrow & (P_z(\alpha_1), \cdots, P_z(\alpha_n)) \end{array}$$

where a_i are randomly generated in $\mathrm{GF}(2^\ell)$
and α_i are distinct public value of $\mathrm{GF}(2^\ell)$.

Information Leaked by a d^{th}-order Shamir's Secret Sharing



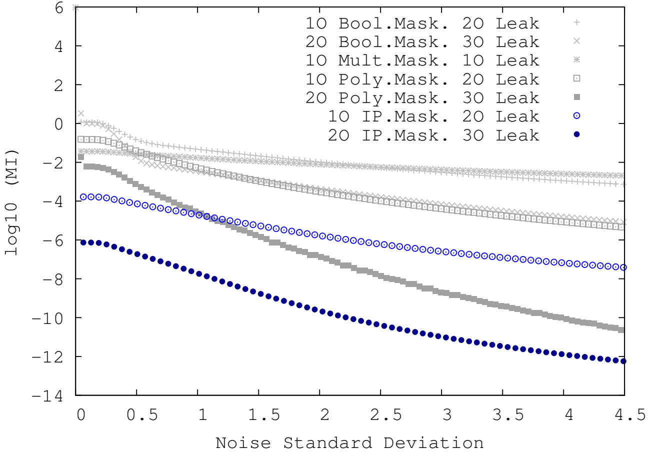
IP-masking [Dziembowski-Faust TCC 2012]

Manipulation of randomized variable

$$z \xrightarrow{\$} \left(\frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n, L_1, \cdots, L_n\right)$$

where R_i are randomly generated in $\mathrm{GF}(2^\ell)$
and L_i are randomly generated in $\mathrm{GF}(2^\ell)^\star$.

Information Leaked by a d^{th}-order IP sharing



IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

- $2n$ shares for $(n-1)$ probing security
- (HO-)Glitches Attack resistant masking scheme
- Weak information leakage assuming standard Leakage Functions *e.g. HW*
- Complexity $O(n^2)$
- Proofs in the continuous bounded-range leakage model only if $n \geq 130$
  - *Practical Leakage Resilient Masking Scheme*
  - $\mathcal{O}() : \{0,1\}^{\ell} \mapsto \{0,1\}^{\lambda}$ $\lambda << \ell$

IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Inner-Product Sharing Scheme

$$z \xrightarrow{\$} \left(\frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n, L_1, \cdots, L_n\right)$$

R_i in $\mathrm{GF}(2^\ell)$, L_i in $\mathrm{GF}(2^\ell)^\star$.

IP-Masking Scheme [Balasch *et al.* 12]

$$\text{inputs} : \{(\mathbf{L_A}, \mathbf{R_A}), (\mathbf{L_B}, \mathbf{R_B})\}$$

- $\texttt{RefreshMasks}(A)$: $O(n)$
- $A + B$: $O(n)$
- $xA + y$: $O(n)$
- $A \times B$: $O(n^2)$

IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Inner-Product Sharing Scheme

$$z \xrightarrow{\$} \left(\frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n, L_1, \cdots, L_n\right)$$

R_i in $\mathrm{GF}(2^\ell)$, L_i in $\mathrm{GF}(2^\ell)^\star$.

IP-Masking Scheme [Balasch *et al.* 12]

$$\text{inputs} : \{(\mathbf{L_A}, \mathbf{R_A}), (\mathbf{L_B}, \mathbf{R_B})\}$$

- `RefreshMasks`(A) : $O(n)$
- $A + B$: $O(n)$
- $xA + y$: $O(n)$
- $A \times B$: $O(n^2)$

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

`/* Refresh Masks */`

$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
    $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{B} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{B}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                                    */
L* ← (randNonZero())^n;
for i = 1 to n do
    A_i ← L_i ⊕ L*_i;
X ← ⟨A, R⟩;
B ← IPHalfMask(X, L*);
R* ← R ⊕ B;
return (L*, R*);
```

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

`/* Refresh Masks */`

$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
    $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{B} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{B}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

`/* Refresh Masks */`

$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
  $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{B} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{B}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

For $n = 2$,

$$\begin{aligned} V &= L_1 R_1 \oplus L_2 R_2 \\ X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2 \end{aligned}$$

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

`/* Refresh Masks */`

$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
  $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{B} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{B}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

For $n = 2$,

$$\begin{aligned} V &= L_1 R_1 \oplus L_2 R_2 \\ X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2 \end{aligned}$$

Algorithm Refresh Mask description

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of V.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                              */
L* ← (randNonZero())^n;
for i = 1 to n do
    A_i ← L_i ⊕ L*_i;
X ← ⟨A, R⟩;
B ← IPHalfMask(X, L*);
R* ← R ⊕ B;
return (L*, R*);
```

For $n = 2$,

$$
\begin{aligned}
V &= L_1 R_1 \oplus L_2 R_2 \\
X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2
\end{aligned}
$$

A 1st-order Flaw for any d

$$\Pr[X = x \mid V = v] = \begin{cases} \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell - 1)^{n-2}} & \text{if } x = 0 \\ \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell - 1)^{n-1}} & \text{if } x \neq 0 \end{cases}$$

for $v = 0$, and

$$\Pr[X = x \mid V = v] = \begin{cases} \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell - 1)^{n-1}} & \text{if } x = v \\ \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell - 1)^{n}} & \text{if } x \neq v \end{cases},$$

otherwise.

A 1st-order Flaw for any d

$$\Pr[X = x \mid V = v] = \begin{cases} \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell-1)^{n-2}} & \text{if } x = 0 \\ \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell-1)^{n-1}} & \text{if } x \neq 0 \end{cases}$$

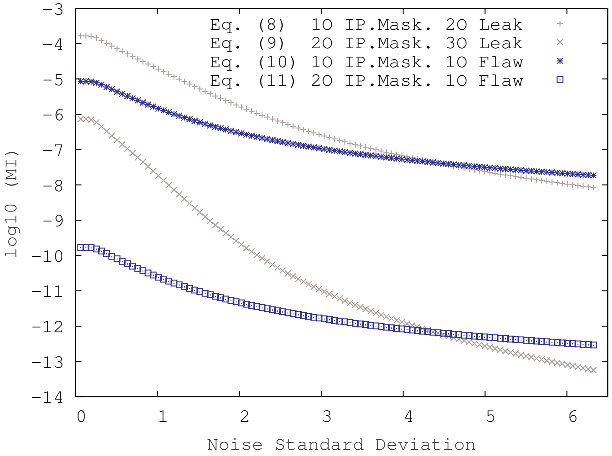
for $v = 0$, and

$$\Pr[X = x \mid V = v] = \begin{cases} \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell-1)^{n-1}} & \text{if } x = v \\ \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell-1)^{n}} & \text{if } x \neq v \end{cases},$$

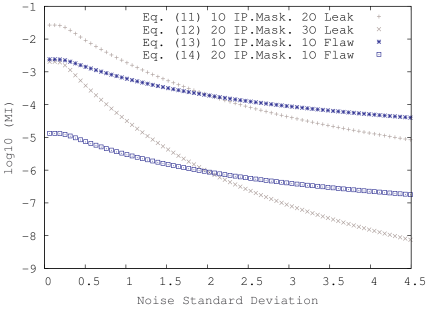
otherwise.

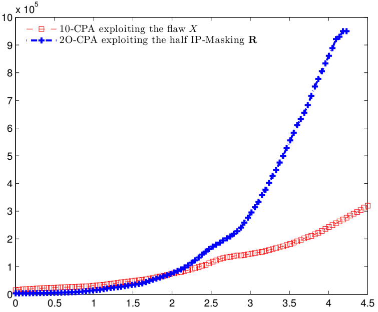
$$\mathcal{I}(V; \mathcal{O}(X)) \neq 0$$

Information Leaked by the 1st-order Flaw



Information Leaked by the 1st-order Flaw on 4-bit variables





A security flaw in Blasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - ↪ in practice much easier to mount than a d-order attack.
  - ↪ noise addition techniques won't help that much.
- proof in the continuous bounded-leakage model is still standing
  - ↪ ways of improving the $n \geq 130$ bound ?

A security flaw in Blasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - $\hookrightarrow$ in practice much easier to mount than a d-order attack.
  - $\hookrightarrow$ noise addition techniques won't help that much.
- proof in the continuous bounded-leakage model is still standing
  - $\hookrightarrow$ ways of improving the $n \geq 130$ bound ?

A security flaw in Blasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - $\hookrightarrow$ in practice much easier to mount than a d-order attack.
  - $\hookrightarrow$ noise addition techniques won't help that much.
- proof in the continuous bounded-leakage model is still standing
  - $\hookrightarrow$ ways of improving the $n \geq 130$ bound ?

IP-Masking Scheme *w.r.t.* to recent results in leakage resilience proofs

- ProufRivain, EUROCRYPT 2013
- security proofs in continuous leakage model

  practical noisy leakage models
- additive masking (Ishai *et al.* scheme)
- improvements and link with probing security

  DucDziembowskiFaust, to appear EUROCRYPT 2014