On the Practical Security of a Leakage Resilient Masking Scheme

Emmanuel Prouff[1], Matthieu Rivain[2], and Thomas Roche[1]

[1] ANSSI, 51, Bd de la Tour-Maubourg, 75700 Paris 07 SP, France
firstname.name@ssi.gouv.fr
[2] CryptoExperts, 41, Bd des Capucines, 75002 Paris, France
matthieu.rivain@cryptoexperts.com

1 Introduction

At TCC 2012, Dziembowski and Faust show how to construct leakage resilient circuits using secret sharing based on the inner product [2]. At Asiacrypt 2012, Ballash *et al.* turned the latter construction into an efficient masking scheme and they apply it to protect an implementation of AES against side-channel attacks [1]. The so-called *Inner-Product masking* (IP masking for short) was claimed to be secure with respect to two different security models: the λ-limited security model[3] (Section 4 of [1]), and the d^{th}-order security model (see definitions p.8 of [1]). In the former model, the security proof makes sense for a sharing dimension $n > 130$ which is acknowledged impractical by the authors. In the latter model, the scheme is claimed secure up to the order $d = n - 1$.

In this note, we contradict the d^{th}-order security claim by exhibiting a 1^{st}-order flaw in the masking algorithm for any chosen sharing dimension n.

2 Inner Product Masking Scheme

Let us first recall the basic principle of IP masking. In the following, $\mathbb{F}_q$ will denote some field of characteristic 2 (*i.e.* $q = 2^m$ for some $m \geqslant 1$), and let $\oplus$ and $\otimes$ denote respectively the addition and the multiplication over $\mathbb{F}_q$. The inner product between two vectors $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ and $\boldsymbol{Y} = (Y_1, Y_2, \dots, Y_n)$ from $\mathbb{F}_q^n$ is denoted by:

$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = (X_1 \otimes Y_1) \oplus (X_2 \otimes Y_2) \oplus \cdots \oplus (X_n \otimes Y_n) \ .$$

The principle of the IP masking scheme is to manipulate every sensitive variable V as a sharing composed of $2n$ elements, namely the coordinates of two vectors $\boldsymbol{L} = (L_1, L_2, \dots, L_n)$ and $\boldsymbol{R} = (R_1, R_2, \dots, R_n)$ such that $V = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$. In order to prevent a direct 1^{st}-order flaw, the coordinates of $\boldsymbol{L}$ are randomly drawn from $\mathbb{F}_q^* = \mathbb{F}_q \backslash \{0\}$.

In order to perform computation in the masked domain, the authors of [1] define some addition and multiplication functions (`IPAdd` and `IPMult`) processing masked variables. Both of them are based on two building blocks: the `IPHalfMask` and `IPRefresh` procedures, which are recalled hereafter.[4]

The `IPHalfMask` procedure (see Algorithm 1) takes a variable $V \in \mathbb{F}_q$ and a half sharing $\boldsymbol{L} \in (\mathbb{F}_q^*)^n$ and it outputs random half sharing $\boldsymbol{R} \in \mathbb{F}_q^n$ satisfying $V = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$.

[3] Often referred to as the *continuous bounded-range leakage model.*

[4] We do not use the algorithmic presentation from [1] involving two different processors as it is useless for the d^{th}-order security model.

Algorithm 1 Half-Masking a variable: $(\boldsymbol{L}, \boldsymbol{R}) \leftarrow \texttt{IPHalfMask}(V, \boldsymbol{L})$

INPUT: variable $V \in \mathbb{F}_q$ and the vector $\boldsymbol{L}$ of the non-zero shares

OUTPUT: masked variable $(\boldsymbol{L}, \boldsymbol{R})$ such that $Z = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$

```
for $i = 2$ to $n$ do
    $R_i \leftarrow \texttt{rand()}$
$R_1 \leftarrow (V \oplus \bigoplus_{i=2}^{n} L_i \otimes R_i) \otimes L_1^{-1}$
return $(\boldsymbol{L}, \boldsymbol{R})$
```

The `IPRefresh` procedure (see Algorithm 2), takes a sharing $(\boldsymbol{L}, \boldsymbol{R})$ and computes a new fresh sharing $(\boldsymbol{L}', \boldsymbol{R}')$ such that $\langle \boldsymbol{L}', \boldsymbol{R}' \rangle = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$.

Algorithm 2 Refresh Vector: $(\boldsymbol{L}', \boldsymbol{R}') \leftarrow \texttt{IPRefresh}(\boldsymbol{L}, \boldsymbol{R})$

INPUT: Masked variable $(\boldsymbol{L}, \boldsymbol{R})$

OUTPUT: New masked variable $(\boldsymbol{L}', \boldsymbol{R}')$ such that $\langle \boldsymbol{L}, \boldsymbol{R} \rangle = \langle \boldsymbol{L}', \boldsymbol{R}' \rangle$

```
1. $\boldsymbol{L}' \leftarrow (\texttt{randNonZero()})^n$
2. for $i = 1$ to $n$ do                                [$\boldsymbol{A} \leftarrow \boldsymbol{L} \oplus \boldsymbol{L}'$]
3.     $A_i \leftarrow L_i \oplus L'_i$
4. $X \leftarrow \langle \boldsymbol{A}, \boldsymbol{R} \rangle$
5. $\boldsymbol{B} \leftarrow \texttt{IPHalfMask}(X, \boldsymbol{L}')$
6. $\boldsymbol{R}' \leftarrow \boldsymbol{R} \oplus \boldsymbol{B}$
7. return $(\boldsymbol{L}', \boldsymbol{R}')$
```

Remark 1. In Algorithm 2, the steps (1-3) for generating $\boldsymbol{A}$ does not correspond to what is described in [1]. We chose this algorithm for simplicity, this has no incidence whatsoever on the following.

We now recall the masked addition `IPAdd` and the masked multiplication `IPMult` in the two following algorithms.

Algorithm 3 Masked Addition: $(\boldsymbol{X}, \boldsymbol{Y}) \leftarrow \texttt{IPAdd}((\boldsymbol{L}, \boldsymbol{R}), (\boldsymbol{K}, \boldsymbol{Q}))$

INPUT: Two Masked variables $(\boldsymbol{L}, \boldsymbol{R})$ and $(\boldsymbol{K}, \boldsymbol{Q})$

OUTPUT: New masked variable $(\boldsymbol{X}, \boldsymbol{Y})$ such that $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \langle \boldsymbol{L}, \boldsymbol{R} \rangle \oplus \langle \boldsymbol{K}, \boldsymbol{Q} \rangle$

```
$(\boldsymbol{A}, \boldsymbol{B}) \leftarrow \texttt{IPRefresh}(\boldsymbol{K}, \boldsymbol{Q} \oplus \boldsymbol{R})$
$(\boldsymbol{C}, \boldsymbol{D}) \leftarrow \texttt{IPRefresh}(\boldsymbol{L} \oplus \boldsymbol{K}, \boldsymbol{R})$
$Z \leftarrow \langle \boldsymbol{C}, \boldsymbol{D} \rangle$
$\boldsymbol{Y} \leftarrow \texttt{IPHalfMask}(Z, \boldsymbol{A})$
$\boldsymbol{X} \leftarrow \boldsymbol{A}$
$\boldsymbol{Y} \leftarrow \boldsymbol{Y} \oplus \boldsymbol{B}$
return $(\boldsymbol{X}, \boldsymbol{Y})$
```

Algorithm 4 Masked Multiplication: $(\boldsymbol{X},\boldsymbol{Y}) \leftarrow \texttt{IPMult}((\boldsymbol{L},\boldsymbol{R}),(\boldsymbol{K},\boldsymbol{Q}))$

INPUT: Two Masked variables $(\boldsymbol{L},\boldsymbol{R})$ and $(\boldsymbol{K},\boldsymbol{Q})$

OUTPUT: New masked variable $(\boldsymbol{X},\boldsymbol{Y})$ such that $\langle \boldsymbol{X},\boldsymbol{Y}\rangle = \langle \boldsymbol{L},\boldsymbol{R}\rangle \otimes \langle \boldsymbol{K},\boldsymbol{Q}\rangle$

```
for $i = 0$ to $n-1$ do
    for $j = 1$ to $n$ do
        $\tilde{U}_{i*n+j} \leftarrow L_{i+1} \otimes K_j$
        $\tilde{V}_{i*n+j} \leftarrow R_{i+1} \otimes Q_j$
$(\boldsymbol{U},\boldsymbol{V}) \leftarrow \texttt{IPRefresh}(\tilde{\boldsymbol{U}},\tilde{\boldsymbol{V}})$
$\boldsymbol{A} \leftarrow (U_1,\cdots,U_n)$;   $\boldsymbol{C} \leftarrow (U_{n+1},\cdots,U_{n^2})$
$\boldsymbol{B} \leftarrow (V_1,\cdots,V_n)$;   $\boldsymbol{D} \leftarrow (V_{n+1},\cdots,V_{n^2})$
$Z \leftarrow \langle \boldsymbol{C},\boldsymbol{D}\rangle$
$\boldsymbol{Y} \leftarrow \texttt{IPHalfMask}(Z,\boldsymbol{A})$
$\boldsymbol{X} \leftarrow \boldsymbol{A}$
$\boldsymbol{Y} \leftarrow \boldsymbol{Y} \oplus \boldsymbol{B}$
return $(\boldsymbol{X},\boldsymbol{Y})$
```

3 A First-Order Flaw

Balasch *et al.* claim that the above IP masking scheme is secure against any side-channel attack of order $d = n-1$, or equivalently, that any family of $n-1$ intermediate variables is independent of any sensitive variable. We contradict this claim hereafter by showing that for any fixed parameter n, there always exists a first-order side-channel attack on the IP masking scheme. To this end, we will exhibit an intermediate variable that is statistically dependent on some sensitive variable in both the `IPRefresh` and `IPAdd` procedures (Algorithms 2 and 3 above).

Let $\boldsymbol{A} = (A_1, A_2, \ldots, A_n)$ and $\boldsymbol{B} = (B_1, B_2, \ldots, B_n)$ be random vectors uniformly distributed over $(\mathbb{F}_q^*)^n$, and let $\boldsymbol{R} = (R_1, R_2, \ldots, R_n)$ be a random vector uniformly distributed over $\mathbb{F}_q^n$, such that $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{R}$ are mutually independent. Consider the function f_n defined by:

$$f_n(a,b) = \Pr[\langle \boldsymbol{A},\boldsymbol{R}\rangle = a \wedge \langle \boldsymbol{B},\boldsymbol{R}\rangle = b] \ . \tag{1}$$

We first study f_n with respect to n before exhibiting the IP masking flaw.

3.1 Study of f_n

The study of f_n developed in this section is recursive. First, in Lemma 1, we give an explicit expression to f_1. Then, in Lemma 2, we exhibit a recursive relationship for f_n. Both lemmas are eventually involved to provide an explicit expression to f_n (Theorem 1).

Lemma 1. *The function f_1 satisfies*

$$f_1(a,b) = \begin{cases} \frac{1}{q} & \text{if } (a,b) = (0,0) \\ 0 & \text{if } (a,b) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ \frac{1}{q(q-1)} & \text{if } (a,b) \in \mathbb{F}_q^* \times \mathbb{F}_q^* \end{cases}$$

Proof. First, since both A_1 and B_1 are non-zero, we have

$$f_1(0,0) \ = \ \Pr[A_1 \otimes R_1 = 0 \wedge B_1 \otimes R_1 = 0] \ = \ \Pr[R_1 = 0] \ = \ \frac{1}{q} \ .$$

Moreover, for any $a \neq 0$, we have

$$f_1(a,0) \;=\; \Pr[R_1 = a \otimes A_1^{-1} \wedge R_1 = 0] \;=\; 0 \; .$$

Similarly, we also have $f(0,b) = 0$ if $b \neq 0$.
Eventually, the total probability law together with the mutual independence between A_1, B_1 and R_1, imply

$$f_1(a,b) \;=\; \sum_{a_1 \in \mathbb{F}_q^*} \Pr[A_1 = a_1] \times \Pr[R_1 = a \otimes a_1^{-1} \wedge B_1 \otimes R_1 = b] \; ,$$

which for $a \neq 0$ and $b \neq 0$ gives

$$f_1(a,b) \;=\; \sum_{a_1 \in \mathbb{F}_q^*} \Pr[A_1 = a_1] \times \Pr[R_1 = a \otimes a_1^{-1} \wedge B_1 = b\,(a^{-1} \otimes a_1)] \;=\; \frac{1}{q(q-1)} \; .$$

□

Lemma 2. *For every $n \geqslant 1$, there exist $f_n^{00}, f_n^{01}, f_n^{11} \in \mathbb{R}$ such that*

$$f_n(a,b) = \begin{cases} f_n^{00} & \textit{if } (a,b) = (0,0) \\ f_n^{01} & \textit{if } (a,b) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ f_n^{11} & \textit{if } (a,b) \in \mathbb{F}_q^* \times \mathbb{F}_q^* \end{cases}$$

Moreover, we have

$$\begin{aligned} f_{n+1}^{00} &= \frac{1}{q} f_n^{00} + \frac{q-1}{q} f_n^{11} \; , \\ f_{n+1}^{01} &= \frac{2}{q} f_n^{01} + \frac{q-2}{q} f_n^{11} \; , \\ f_{n+1}^{11} &= \frac{1}{q(q-1)} f_n^{00} + \frac{2(q-2)}{q(q-1)} f_n^{01} + \frac{(q-1)+(q-2)^2}{q(q-1)} f_n^{11} \; . \end{aligned}$$

Proof. The first statement is true for $n = 1$ by Lemma 1. It is then implied by recurrence from the second statement. Therefore, we only need to show the latter statement.

For every $n > 1$, the total probability law implies

$$f_{n+1}(a,b) = \sum_{(a_0,b_0) \in \mathbb{F}_q^2} f_n(a \oplus a_0, b \oplus b_0) f_1(a_0,b_0) \; . \tag{2}$$

1. For $(a,b) = (0,0)$, the terms in the sum (2) are of the form $f_n(a_0,b_0) f_1(a_0,b_0)$. Then by Lemma 1, we get

$$f_n(a_0,b_0) f_1(a_0,b_0) = \begin{cases} \frac{1}{q} f_n(0,0) & \text{if } (a_0,b_0) = (0,0) \\ 0 & \text{if } (a_0,b_0) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ \frac{1}{q(q-1)} f_n(a_0,b_0) & \text{if } (a_0,b_0) \in \mathbb{F}_q^* \times \mathbb{F}_q^* \end{cases}$$

   We deduce

$$f_{n+1}(a,b) = \frac{1}{q} f_n^{00} + (q-1)^2 \frac{1}{q(q-1)} f_n^{11} \; . \tag{3}$$

2. For $(a,b) \in \{0\} \times \mathbb{F}_q^*$, the terms in the sum (2) are of the form $f_n(a_0, b \oplus b_0) f_1(a_0, b_0)$, with $b \neq 0$. Then by Lemma 1, we get

$$f_n(a_0, b \oplus b_0) f_1(a_0, b_0) = \begin{cases} \frac{1}{q} f_n(0,b) & \text{if } (a_0,b_0) = (0,0) \\ 0 & \text{if } (a_0,b_0) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ \frac{1}{q(q-1)} f_n(a_0,0) & \text{if } (a_0,b_0) \in \mathbb{F}_q^* \times \{b\} \\ \frac{1}{q(q-1)} f_n(a_0,b_0) & \text{if } (a_0,b_0) \in \mathbb{F}_q^* \times (\mathbb{F}_q^* \backslash \{b\}) \end{cases}$$

We deduce

$$f_{n+1}(a,b) = \frac{1}{q} f_n^{01} + (q-1) \frac{1}{q(q-1)} f_n^{01} + (q-1)(q-2) \frac{1}{q(q-1)} f_n^{11} \ . \tag{4}$$

For $(a,b) \in \mathbb{F}_q^* \times \{0\}$, we have the same equality by symmetry of the function f_n.

3. For $(a,b) \in \mathbb{F}_q^* \times \mathbb{F}_q^*$, the terms in the sum (2) are of the form $f_n(a \oplus a_0, b \oplus b_0) f_1(a_0, b_0)$, with $a \neq 0$ and $b \neq 0$. Then by Lemma 1, we get

$$f_n(a \oplus a_0, b \oplus b_0) f_1(a_0, b_0) = \begin{cases} \frac{1}{q} f_n(a,b) & \text{if } (a_0,b_0) = (0,0) \\ \frac{1}{q(q-1)} f_n(0,0) & \text{if } (a_0,b_0) = (a,b) \\ 0 & \text{if } (a_0,b_0) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ \frac{1}{q(q-1)} f_n(a \oplus a_0, 0) & \text{if } (a_0,b_0) \in (\mathbb{F}_q^* \backslash \{a\}) \times \{b\} \\ \frac{1}{q(q-1)} f_n(0, b \oplus b_0) & \text{if } (a_0,b_0) \in \{a\} \times (\mathbb{F}_q^* \backslash \{b\}) \\ \frac{1}{q(q-1)} f_n(a \oplus a_0, b \oplus b_0) & \text{if } (a_0,b_0) \in (\mathbb{F}_q^* \backslash \{a\}) \times (\mathbb{F}_q^* \backslash \{b\}) \end{cases}$$

We deduce

$$f_{n+1}(a,b) = \frac{1}{q} f_n^{11} + \frac{1}{q(q-1)} f_n^{00} + 2\Big((q-2) \frac{1}{q(q-1)} f_n^{01}\Big) + (q-2)^2 \frac{1}{q(q-1)} f_n^{11} \ . \tag{5}$$

Equations (3), (4) and (5) directly yield the second statement. □

Theorem 1. *For every $n \geqslant 1$ we have*

$$f_n(a,b) = \begin{cases} \frac{1}{q^2} + \frac{1}{q^2(q-1)^{n-2}} & \textit{if } (a,b) = (0,0) \\ \frac{1}{q^2} - \frac{1}{q^2(q-1)^{n-1}} & \textit{if } (a,b) \in (\{0\} \times \mathbb{F}_q^*) \cup (\mathbb{F}_q^* \times \{0\}) \\ \frac{1}{q^2} + \frac{1}{q^2(q-1)^{n}} & \textit{if } (a,b) \in \mathbb{F}_q^* \times \mathbb{F}_q^* \end{cases}$$

Proof. From Lemma 2, we have

$$\begin{pmatrix} f_{n+1}^{00} \\ f_{n+1}^{01} \\ f_{n+1}^{11} \end{pmatrix} = \begin{pmatrix} \frac{1}{q} & 0 & \frac{q-1}{q} \\ 0 & \frac{2}{q} & \frac{q-2}{q} \\ \frac{1}{q(q-1)} & \frac{2(q-2)}{q(q-1)} & \frac{(q-1)+(q-2)^2}{q(q-1)} \end{pmatrix} \cdot \begin{pmatrix} f_n^{00} \\ f_n^{01} \\ f_n^{11} \end{pmatrix} = P \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{q-1} \end{pmatrix} \cdot P^{-1} \cdot \begin{pmatrix} f_n^{00} \\ f_n^{01} \\ f_n^{11} \end{pmatrix} \tag{6}$$

where P is the matrix of eigenvectors which satisfies

$$P = \begin{pmatrix} 1 & 1-q & q^2-2q+1 \\ 1 & \frac{1}{2}(2-q) & 1-q \\ 1 & 1 & 1 \end{pmatrix}$$

By recursively applying (6), we can express $(f_n^{00}, f_n^{01}, f_n^{11})$ with respect to $(f_1^{00}, f_1^{01}, f_1^{11})$ as

$$\begin{pmatrix} f_n^{00} \\ f_n^{01} \\ f_n^{11} \end{pmatrix} = P \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{(q-1)^{n-1}} \end{pmatrix} \cdot P^{-1} \cdot \begin{pmatrix} f_1^{00} \\ f_1^{01} \\ f_1^{11} \end{pmatrix}$$

Finally, by Lemma 1 we have $(f_1^{00}, f_1^{01}, f_1^{11}) = \left(\frac{1}{q}, 0, \frac{1}{q(q-1)}\right)$, which together with the above equation yields the theorem statement. □

3.2 Application to the IP Masking Scheme

The flaw occurs in the mask-refreshing procedure `IPRefresh` and in the addition procedure `IPAdd` (see in Algorithm 2 and Algorithm 3). For the sake of clarity, we first detail it in the `IPRefresh` setting and then show it occurs as well in the `IPAdd` procedure.

Flaw in mask-refreshing procedure. The `IPRefresh` procedure takes an IP masking $(\boldsymbol{L}, \boldsymbol{R})$ of some sensitive variable V (*i.e.* such that $V = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$), and it returns a fresh masking $(\boldsymbol{L}', \boldsymbol{R}')$ such that $V = \langle \boldsymbol{L}', \boldsymbol{R}' \rangle$. The first step of the procedure consists in randomly picking some vector $\boldsymbol{A} \in \mathbb{F}_q^n$ such that $A_i \neq L_i$ for every i. Then one computes $\boldsymbol{L}' = \boldsymbol{L} \oplus \boldsymbol{A}$ and $X = \langle \boldsymbol{A}, \boldsymbol{R} \rangle$. Note that $\boldsymbol{L}$ and $\boldsymbol{L}'$ are mutually independent and both uniformly distributed over $(\mathbb{F}_q^*)^n$. We show hereafter that X leaks information on the sensitive variable V. Indeed we have

$$\Pr[X = x \mid V = v] = \frac{\Pr[V = v \wedge X = x]}{\Pr[V = v]} = \frac{\Pr[V = v \wedge X \oplus V = x \oplus v]}{\Pr[V = v]} \ .$$

Then from

$$\Pr[V = v \wedge X \oplus V = x \oplus v] = \Pr[\langle \boldsymbol{L}, \boldsymbol{R} \rangle = v \wedge \langle \boldsymbol{L}', \boldsymbol{R} \rangle = x \oplus v] = f_n(v, x \oplus v) \ ,$$

we get

$$\Pr[X = x \mid V = v] = \frac{f_n(v, x \oplus v)}{\Pr[V = v]} \ . \tag{7}$$

By Theorem 1 and given that $\Pr[V = v] = \frac{1}{q}$, (7) gives

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

for $v = 0$, and

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

otherwise.

We see that when the sensitive variable V equals 0, then the intermediate variable X is more likely to equal 0 than another value in $\mathbb{F}_q$. On the other hand, when V does not equal 0, the sensitive variable X is more likely to be any value of $\mathbb{F}_q$ but v. Although the bias is exponentially small in n, for small values of n it may induce a significant information leakage.

Flaw in the addition procedure. The IPAdd procedure is subject to a similar flaw. Indeed at Step 3 of Algorithm 3, one computes

$$Z = \langle \boldsymbol{C}, \boldsymbol{D} \rangle = \langle \boldsymbol{L} \oplus \boldsymbol{K}, \boldsymbol{R} \rangle \ ,$$

where $\boldsymbol{L}$ and $\boldsymbol{K}$ are mutually independent and both uniformly distributed over $(\mathbb{F}_q^*)^n$. Therefore the distribution of Z given $V = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$ suffers the exact same bias as the distribution of X in the IPRefresh procedure.

Remark 2. It can be noted that the IPMult procedure looks more secure. Indeed except for the IPRefresh call, we did not find any flaw in the actual algorithm. Moreover the IPRefresh procedure is called on a sharing of dimension n^2. Hence, even for small values of n, the observed bias quickly becomes very small.

4 Mutual Information Evaluation of the First-Order Flaw

We have seen in Section 3.2 that Balasch *et al.* 's proposal possesses a first-order flaw whatever the masking dimension n of their scheme. To complete our study, we conduct hereafter an information theoretic evaluation of the flaw in a common leakage model (namely Hamming weight leakage with Gaussian noise). We compare the quantity of leaking information from the flaw with that of the natural nth-order leakage from the right half sharing $\boldsymbol{R}$ for $n = 2$ and $n = 3$.

To quantify the amount of leaking information, we model the relationship between the physical leakage and the manipulated variables as follows. Each tuple of variables $(V_1, V_2, \cdots, V_t)$ is associated with a tuple of leakages $\mathcal{L} = (L_1, L_2, \cdots, L_t)$ s.t. $L_j = \mathrm{HW}(V_j) + \mathcal{N}_j$, where HW denotes the Hamming weight function and $\mathcal{N}_j$ denotes an independent Gaussian variable with mean 0 and standard deviation σ. We use the notation $\mathcal{L} \hookleftarrow (V_1, V_2, \cdots, V_t)$ to refer to this association. To compare the information revealed by the flaw and that revealed by the leakage the right half sharing, we computed the mutual information[5] $I(V, \mathcal{L})$ between the sensitive variable $V = \langle \boldsymbol{L}, \boldsymbol{R} \rangle$ and the leakage $\mathcal{L}$ in the following situations:

$$\text{Right-half leakage for } n = 2\text{:} \quad \mathcal{L} \hookleftarrow \quad (\boldsymbol{R} = (R_1, R_n)) \ . \tag{8}$$

$$\text{Right-half leakage for } n = 3\text{:} \quad \mathcal{L} \hookleftarrow (\boldsymbol{R} = (R_1, R_2, R_3)) \ . \tag{9}$$

$$\text{First-order flaw for } n = 2\text{:} \quad \mathcal{L} \hookleftarrow (X = \langle \boldsymbol{L} \oplus \boldsymbol{L}', \boldsymbol{R} \rangle) \ . \tag{10}$$

$$\text{First-order flaw for } n = 3\text{:} \quad \mathcal{L} \hookleftarrow (X = \langle \boldsymbol{L} \oplus \boldsymbol{L}', \boldsymbol{R} \rangle) \ . \tag{11}$$

Figure 4 summarizes the information theoretic evaluation for each leakage (8) to (11). It can be observed that for each sharing dimension n, there exists a gap value of σ up to which the first-order flaw become more informative than the overall right-half leakage. For instance, for $n = 2$, this gap value is $\sigma \approx 4.5$. This observation is in accordance with the soundness of the d^{th}-order security notion: a security at a greater order implies a smaller asymptotic leakage (with respect to an increasing noise). We also emphasize that the d^{th}-order security notion is relevant towards more practical issues: the resynchronization of leakage traces and

[5] As shown in [3], the number of measurements required to achieve a given success-rate in a maximum likelihood attack is related to the mutual information evaluation and it roughly equals $c \times I(A, \mathcal{L})^{-1}$, where c is a constant related to the chosen success-rate.

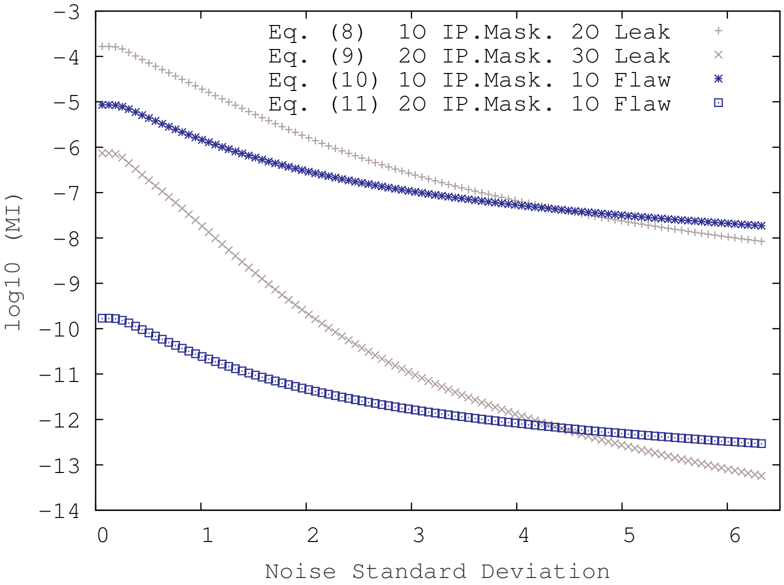

Fig. 1. Mutual information ($\log_{10}$) between the leakage and the sensitive variable over an increasing noise standard deviation (x-axis).

detection of the points of interest. These issues make higher-order attacks much more difficult to mount in practice than first-order ones. As a consequence, the first-order flaw has a greater impact from a practical point of view than suggested in Fig. 4.

References

1. J. Balasch, S. Faust, B. Gierlichs, and I. Verbauwhede. Theory and practice of a leakage resilient masking scheme. In X. Wang and K. Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012.
2. S. Dziembowski and S. Faust. Leakage-resilient circuits without computational assumptions. In R. Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 230–247. Springer, 2012.
3. F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard. The World is not Enough: Another Look on Second-Order DPA. In M. Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.