Connecting and Improving Direct Sum Masking and Inner Product Masking

Romain Poussier[1], Qian Guo[1], François-Xavier Standaert[1],
Claude Carlet[2], Sylvain Guilley[3]

[1] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
[2] LAGA, CNRS, Univ. Paris VIII and Paris XIII, France.
[3] TELECOM-ParisTech, Crypto Group, Paris-Saclay University, CNRS, France.

Abstract. Direct Sum Masking (DSM) and Inner Product (IP) masking are two types of countermeasures that have been introduced as alternatives to simpler (e.g., additive) masking schemes to protect cryptographic implementations against side-channel analysis. In this paper, we first show that IP masking can be written as a particular case of DSM. We then analyze the improved security properties that these (more complex) encodings can provide over Boolean masking. For this purpose, we introduce a slight variation of the probing model, which allows us to provide a simple explanation to the "security order amplification" for such masking schemes that was put forward at CARDIS 2016. We then use our model to search for new instances of masking schemes that optimize this security order amplification. We finally discuss the relevance of this security order amplification (and its underlying assumption of linear leakages) based on an experimental case study.

1 Introduction

Masking is among the most investigated countermeasures against side-channel analysis. It aims at performing cryptographic computations on encoded (aka secret shared) data in order to amplify the impact of the noise in the adversary's observations [10, 31, 13, 14]. For example, in the context of block ciphers, a lot of attention has been paid to the efficient exploitation of simple encodings such as additive (e.g., Boolean ones in [32, 12, 25]) or multiplicative ones (e.g., [20, 19]). Very summarized, the main advantage of these simple encodings is that they enable efficient implementations [22].

In parallel, an alternative trend has investigated the potential advantages of slightly more complex encodings. Typical examples include polynomial masking (e.g., [18, 21, 33]), Inner Product (IP) masking [17, 2, 1] and code-based masking (e.g., [6, 5, 8, 7, 9]). While computing over these encodings is generally more expensive [25], the recent literature has shown that their elaborate algebraic structure also leads to improved security properties. For example, it can decrease the information leakages observed in "low noise conditions" [18, 21, 33, 2, 1]. Also, it can improve the "statistical security order" (or security order in the bounded moment leakage model [3]) in case of linear leakage functions [26, 8, 38]. So while it is an open problem to find out which masking scheme offers the best security vs. efficiency tradeoff for complete implementations in actual devices, the better understanding and connection of simple and complex encoding functions appears as a necessary first step in this direction.

For this purpose, and as a starting point, we note that it has already been shown in [2] that IP masking can be viewed as a generalization of simpler encodings (Boolean, multiplicative, affine and polynomial). So our focus in this paper will be on the connection between IP masking and the Direct Sum Masking (DSM) [5, 9], which is a quite general instance of code-based masking. Our contributions in this respect are as follows:

First, we connect IP masking and DSM by showing that the first one can be seen as a particular case of the latter one. Second, we analyze the security properties of these masking schemes. In particular, we show that the "security order amplification" put forward in previous works can be easily explained thanks to (a variation of) the probing model [27], by considering bit-level security, rather than larger (field-element-level) security. Thanks to this connection, we then express how to best optimize the security order amplification (i.e., the bit-level security) based on the dual distance of a binary code. We further perform an informed search on code instances which allows us to improve the state-of-the-art parameters for IP encodings. We finally propose experiments discussing the interest and limitations of security order amplification in practice (i.e., the relevance of the linear leakage assumption).

Cautionary note. Our focus in this paper is on encodings. Admittedly, an even more important issue is to compute (in particular, multiply) efficiently over encodings. In this respect, while the literature on IP masking provides solutions to this problem [2, 1], it remains an open challenge to describe efficient multiplication algorithms for DSM.

2 Connecting DSM and IP masking

In this section we first introduce the two masking schemes that we will analyze, namely IP masking and DSM. We then show how these two methods are connected, focusing only on their functional description (security will be investigated in Section 3).

2.1 Notations

We use capital letters for random variables and small caps for their realizations. We denote the conditional probability of a random variable A given B with $\mathrm{P}[A|B]$. We use sans serif font for functions (e.g., F) and calligraphic fonts for sets (e.g., $\mathcal{A}$). Given a field $\mathbb{K}$, we denote by $a \cdot b$ the field multiplication between two elements a and b. We denote by $[a]_2$ the binary vector representation of some element $a \in \mathbb{F}_{2^k}$ for some k. We use capital bold letters for matrices (e.g., $\mathbf{M}$) and small bold caps for raw vectors (e.g., $\mathbf{v}$). We denote by $\mathbf{v}(i)$ the i-th element of a vector $\mathbf{v}$. We denote by $\mathbf{M}^T$ (resp. $\mathbf{v}^T$) the transpose of a matrix M (resp. a vector $\mathbf{v}$). The inner product between two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ is noted $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$. In the rest of the paper, x denotes a k-bit secret value that we wish to mask and $[x]_2$ its binary vector representation.

2.2 Inner Product masking

IP masking was introduced in [17, 1, 2] as a generalization of Boolean masking. Instead of simply splitting a secret value as the sum of random shares, it decomposes the secret as the inner product between random values and a public vector. More formally, the first step of

the IP encoding is to select a public vector $\mathbf{l} = (l_0, ..., l_{n-1}) \in \mathbb{F}_{2^k}^n \backslash \{0\}$ (with l_0 generally set as $l_0 = 1$ for performance reasons), where n is the number of shares. A sensitive variable $x \in \mathbb{F}_{2^k}$ is then encoded as the vector $\mathbf{s}_{IP} = (s_0, ..., s_{n-1}) \in \mathbb{F}_{2^k}^n$ such that $x = \langle \mathbf{l}, \mathbf{s}_{IP} \rangle$. Algorithm 1 describes the masking initialization procedure, where the function $\mathsf{rand}(\mathbb{F}_{2^k})$ returns a random element uniformly from $\mathbb{F}_{2^k}$. Boolean masking is the particular case of IP masking where $l_i = 1$ for $i \in [0, n-1]$.

Algorithm 1 IPMask.

Require: $x, \mathbf{l}, n$
Ensure: $\mathbf{s}_{IP}$ such that $x = \langle \mathbf{l}, \mathbf{s}_{IP} \rangle$
  for $i = 1$ to $n - 1$ **do**
    $s_i \leftarrow \mathsf{rand}(\mathbb{F}_{2^k})$
  end for
  $s_0 = x + \sum_{i=1}^{n-1} l_i \cdot s_i$
  return $\mathbf{s}_{IP}$

2.3 Direct Sum masking

DSM [5, 9] describes masking from an error correcting code viewpoint. As opposed to IP masking, this scheme works on the bit level. That is, a sensitive variable x is viewed as belonging to $\mathbb{F}_2^k$ instead of $\mathbb{F}_{2^k}$ and is thus represented as $[x]_2$. It allows adding an arbitrary amount m of bits of randomness to the encoding of $[x]_2$ (i.e., not necessarily a multiple of k as in IP masking). As a result, the final encoding $\mathbf{s}_{DSM}$ of $[x]_2$ lies in $\mathbb{F}_2^{k+m}$. As a first step, the vector space $\mathbb{F}_2^{k+m}$ is decomposed in two subspaces C and D of dimensions k and m:

$$\mathbb{F}_2^{k+m} = C \oplus D,$$

where C and D respectively represent the spaces where the sensitive variable and the mask lie. That is, the sensitive variables (resp., the mask) are the code words of C (resp., D) of length of $k+m$. We denote by $\mathbf{G}$ and $\mathbf{H}$ the generator matrices of C and D. The encoding of $[x]_2$ is the vector $\mathbf{s}_{DSM} = [x]_2\mathbf{G} + \mathbf{y}\mathbf{H}$, where $\mathbf{y} \in \mathbb{F}_2^m$ is a random binary vector. Recovering $[x]_2$ (i.e., decoding) or $\mathbf{y}$ from $\mathbf{s}_{DSM}$ can then be achieved thanks to a projection on their respective space. We stress the fact that while this scheme has been designed to thwart both side-channel and fault attacks (if C and D are orthogonal), we only focus on the side-channel part.

2.4 Unifying DSM and IP masking

From the previous description of IP masking and DSM, we now show how these two schemes are connected. We recall that the IP encoding of a sensitive variable x using n shares is the vector $\mathbf{s}_{IP} = (s_0 = x + l_1 \cdot s_1 + ... + l_{n-1} \cdot s_{n-1}, s_1, ..., s_{n-1}) \in \mathbb{F}_{2^k}^n$. In order to make the connection with DSM, we first have to move its base field from $\mathbb{F}_2$ to $\mathbb{F}_{2^k}$. That is, we want the final DSM encoding to belong to $\mathbb{F}_{2^k}^n$. We next decompose $\mathbb{F}_{2^k}^n$ using two supplementary subspaces C and D such that $\mathbb{F}_{2^k}^n = C \oplus D$, where the dimension of C is 1 and the dimension

of D is $n-1$. As in the previous subsection, we denote by $\mathbf{G}$ and $\mathbf{H}$ the generator matrices of C and D that we define as follow (where each element belong to $\mathbb{F}_{2^k}$):

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} \qquad \mathbf{H} = \begin{pmatrix} l_1 & 1 & 0 & \dots & 0 \\ l_2 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n-1} & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Equation 1 then shows the encoding vector $\mathbf{s}_{MIX}$ of a secret $x \in \mathbb{F}_{2^k}$ using a randomness vector $\mathbf{y} = (y_1, ..., y_{n-1}) \in \mathbb{F}_{2^k}^{n-1}$:

$$\begin{aligned} \mathbf{s}_{MIX} &= x\mathbf{G} + \mathbf{y}\mathbf{H}, \\ &= (x, 0, ..., 0) + (l_1 \cdot y_1 + ... + l_{n-1} \cdot y_{n-1}, y_1, ..., y_{n-1}), \\ &= (x + l_1 \cdot y_1 + ... + l_{n-1} \cdot y_{n-1}, y_1, ..., y_{n-1}), \\ &= \mathbf{s}_{IP}. \end{aligned} \tag{1}$$

The encoding of an IP masking can thus be written by adapting the DSM scheme base field and choosing the generating matrices accordingly. However, this modification discards one property of the original DSM scheme. Namely, the number of bits of randomness added to the encoding cannot be arbitrarily chosen as it has to be a multiple of k. Besides, we note that the discussions in [5] additionally required the codes C and D to be orthogonal. Yet, this additional property is not required in our discussions that focus only on side-channel security, and where the secret x can be recovered using a projection: $x = \langle \mathbf{l}, \mathbf{s}_{MIX} \rangle$.

3 Probing security and bit probing security

In this section, we discuss the side-channel resistance of the IP masking and DSM in the probing model [27]. For each method, we look at the security of the encoding. In the case of IP masking, we assume that the size k of the base field corresponds to the word size of the implementation and the probes allow the adversary to observe such field elements. As the DSM works on the bit level, we additionally introduce the bit-probing model, where each probe can only look at one bit of the encoding. We finally make the link between the (general, ie.g., field-level) probing security and the bit-probing security of the inner product masking. This connection will be used in the next section in order to explain the security order amplification of the IP masking.

3.1 Probing security

The probing model introduced in [27] formalizes the security improvement obtained with the masking countermeasure. Informally, dth-order probing security ensures that the distribution of any d or less intermediate variables manipulated during the algorithm execution

is independent of any secret value. From an attacker point-of-view, it implies that only the combination of at least $d+1$ intermediate variables can give information on the secret. As a result, the practical security increases exponentially in the number of shares, which is intuitively explained by the fact that the adversary has to estimate higher-order statistical moments, a task of which the sampling complexity grows exponentially in the order [10, 31, 13, 14]. In the case of IP masking with n shares, previous works showed that the encoding has a probing security of order $d = n - 1$ [2, 1].

3.2 Bit-probing security

Thanks to the link exhibited in the previous section, we naturally have that DSM is secure in the probing model as well, which also follows the analysis in [5, 9]. However, since DSM works at the bit level, we additionally define the bit-probing security as the security in a tweaked probing model, where each probe can evaluate only one bit of the encoding (even if this encoding is defined for larger fields). In this model, the security order is thus the maximum number d' such that any combination of d' bits of $\mathbf{s}_{DSM}$ is independent of the secret $[x]_2$. More formally, the bit-probing security of the DSM scheme is given by:

Proposition 1 *let C and D two codes of generator matrices $\mathbf{G}$ and $\mathbf{H}$ define a DSM encoding. Let k and m respectively be the dimensions of C and D. The bit-probing security of the DSM encoding defined by C and D is equal to the distance of the dual code (called the dual distance) of D minus 1.*

Proof. Let $\mathbf{s}$ be the encoding of some value $[x]_2$. We have $\mathbf{s}_{DSM} = [x]_2\mathbf{G} + \mathbf{y}\mathbf{H}$, a vector of $k+m$ bits. The bit-probing security is the number d' such that at least $d'+1$ elements of $\mathbf{s}_{DSM}$ are required to recover at least one bit of $[x]_2$. If we consider the system given by Equation 2:

$$([x]_2\mathbf{G} + \mathbf{y}\mathbf{H})^T = \mathbf{s}_{DSM}^T, \tag{2}$$

where only the right part is known, $d'+1$ is equal to the smallest number of sub-equations of this system that allows recovering at least one bit of $[x]_2$. We assume that the dual distance of D is equal to $d+1$, which is the minimum number of columns of $\mathbf{H}$ that can be linearly dependent. This means that at least $d+1$ sub-equations of the system in Equation 2 are required to suppress the influence of $(\mathbf{y}\mathbf{H})^T$, and thus get information on $[x]_2$. As a result, the bit-probing security of the DSM encoding is equal to d. □

Note that as will be clear next, the bit-probing security order (which can be higher than the probing security order) does not always guarantee a higher practical security order (i.e., in the bounded moment or noisy leakage models [31, 3]) than predicted by the (field-level) probing security order. Yet, it will be instrumental in explaining the security order amplification for certain types of leakage functions put forward in [38].

3.3 Inner product and bit-probing security

We now consider the bit-probing security of the IP masking encoding. In Section 2.4, we showed how IP masking and DSM are linked by changing the base field of the DSM scheme from $\mathbb{F}_2$ to $\mathbb{F}_{2^k}$. In order to assess the bit-probing security of the IP masking by using Proposition 1, we have changed the base field back from $\mathbb{F}_{2^k}$ to $\mathbb{F}_2$. As a result, we want a new

encoding $\mathbf{s}_{MIX2}$ that belongs to $\mathbb{F}_2^{kn}$ such that each bit of $\mathbf{s}_{MIX}$ and $\mathbf{s}_{MIX2}$ are the same: $[\mathbf{s}_{MIX}(i)]_2 = (\mathbf{s}_{MIX2}(ki), ..., \mathbf{s}_{MIX2}(k(i+1)-1))$.

As a preliminary, we first define by $\mathbf{L}_i$ (with $i \in [1, n-1]$) the $k \times k$ binary matrices that represent the multiplication by l_i in $\mathbb{F}_{2^k}$. That is, given some value $x \in \mathbb{F}_{2^k}$, we define $\mathbf{L_i}$ such that $[l_i \cdot x]_2 = (\mathbf{L}_i \times {[x]_2}^T)^T$. The matrix $\mathbf{L}_i$ can be constructed such that its j-th column is equal to $[\alpha^j \cdot l_i]_2$, where α is a root of the polynomial used to create $\mathbb{F}_{2^k}$.

We now define two codes C and D such that $\mathbb{F}_2^{kn} = C \oplus D$, the dimension of C is k, and the dimension of D is $k(n-1)$, with their generator matrix $\mathbf{G}$ and $\mathbf{H}$ specified as follow:

$$\mathbf{G} = \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{pmatrix}, \qquad \mathbf{H} = \begin{pmatrix} \mathbf{L}_1 & \mathbf{1}_k & \mathbf{0}_k & \dots & 0_k \\ \mathbf{L}_2 & 0_k & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_k \\ \mathbf{L}_{n-1} & 0_k & \dots & 0_k & \mathbf{1}_k \end{pmatrix},$$

such that the first k columns of $\mathbf{G}$ are 1 and the next $k(n-1)$ are 0. Here, $\mathbf{1}_k$ denotes the $k \times k$ identity matrix and $\mathbf{0}_k$ denotes the $k \times k$ zero matrix. Equation 3 then shows the encoding vector $\mathbf{s}_{MIX2}$ of a secret $[x]_2 \in \mathbb{F}_2^k$ using a randomness vector $\mathbf{y} = (y_1, ..., y_{k(n-1)}) \in \mathbb{F}_2^{k(n-1)}$:

$$\begin{aligned} \mathbf{s}_{MIX2} &= [x]_2\mathbf{G} + \mathbf{y}\mathbf{H}, \\ &= ([x]_2, 0, ..., 0) + ([l_1 \cdot y_1]_2 + ... + [l_{n-1} \cdot y_{n-1}]_2, [y_1]_2, ..., [y_{n-1}]_2), \\ &= ([\mathbf{s}_{MIX}(0)]_2, ..., [\mathbf{s}_{MIX}(n-1)]_2)\cdot \end{aligned} \tag{3}$$

From Proposition 1, we know that the bit-probing security of $\mathbf{s}_{MIX2}$ corresponds to the dual distance of $\mathbf{H}$ minus 1 (which depends on the selection of the $\mathbf{l} = (l_1, ..., l_{n-1})$ vector of the IP masking, as already hinted in [38]). As a result, the best bit-probing security using n shares can be achieved by selecting $\mathbf{l}$ such that the dual distance of $\mathbf{H}$ is maximized.

4 Security order amplification

Under some physical assumption that will be discussed later, it has been observed that the concrete security order of the IP encoding (in the bounded moment or noisy leakage models [31, 3]) can be higher than the one given by its probing security [38]. In this section, we provide a formal explanation of this phenomenon that we so-far denoted as security order amplification. We first introduce the bounded moment model that we will use for this purpose [3]. We then apply this model to the IP masking, and explain its link with security order amplification.

4.1 Bounded moment model

The bounded moment leakage model has been introduced in [3], mainly in order to formalize the security of parallel implementations and to connect probing security with current (moment-based) evaluation practices such as [35].

For our following discussions, we will consider a n-share masked implementation with encoding $\mathbf{s} = (s_0, ..., s_{n-1})$ of a secret x that manipulates all the shares within N cycles. As in [3], we denote by $\mathcal{Y}_c$ the set of shares that are manipulated during the cycle c ($0 \leq c \leq N-1$) and by n_c the cardinal of $\mathcal{Y}_c$. We assume that the random variable L_c that represents the leakage associated to the computation during the cycle c follows a linear model:

$$L_c = \alpha_c^0 \mathsf{L}_c^0(\mathcal{Y}_c(0)) + ... + \alpha_c^{n_c-1} \mathsf{L}_c^{n_c-1}(\mathcal{Y}_c(n_c-1)) + R_c, \tag{4}$$

where L_c^i denotes the deterministic leakage part associated to the manipulation of the share $\mathcal{Y}_c(i)$ and R_c a random noise variable. Note that by linear model, we mean that the different L_c's are summed in Equation 4, which is needed to ensure that the leakages corresponding to different shares are independent (otherwise even the probing security order will not be reflected in the bounded moment or noisy leakage models). By contrast, so far we do not assume that the L_c's are linear (this will be only needed for security order amplification).

A fully serial implementation corresponds to the case $N = n$ and $n_c = 1, c \in [0, N-1]$. On the opposite, a fully parallel implementation would be $N = 1$ and $n_0 = n$. In the later case, higher-order probing security can never be achieved as a single variable contains the information on all shares. Hence, the bounded moment model has been introduced to characterize the security of such fully parallel implementations. Basically, having a bounded moment security of order d means that any statistical moment up to the order d of the leakage distribution $\{\mathsf{L}_c\}_{c=0}^{N-1}$ is independent of the secret. More formally, Definition 1 describes the bounded moment security.

Definition 1. *Let $\{L_c\}_{c=0}^{N-1}$ be the leakages of a N cycles parallel masked implementation that manipulates a secret x. We denote by E the expectation operation. This implementation is security at order d if, for all N-tuples $d_i \in \mathbb{N}^N$ such that $\sum_{i=0}^{N-1} d_i \leq d$, we have that $\mathsf{E}(\mathsf{L}_0^{d_0} \times ... \times \mathsf{L}_{N-1}^{d_{N-1}})$ is independent of x.*

Interestingly, it has been shown in [3] that proving security at order o in the probing model (for a serial n-cycle implementation) implies security at order o in the bounded moment model for the corresponding parallel (1-cycle) implementation. We will use this theorem in the next subsection to prove the security order amplification of the inner product masking.

4.2 Security order amplification for IP masking

We now assume an implementation of the IP masking with n shares on $\mathbb{F}_2^k$, where one share corresponds to one variable. That is, we consider the encoding $\mathbf{s}_{MIX} = (s_0, ..., s_{n-1})$ such that each s_i is manipulated independently. We denote by L_i the random variable that represents the leakage on s_i. We assume that the different L_i's are independent and are the sum of a deterministic and random part: $L_i = \mathsf{L}_i(s_i) + R_i$, where L_i denotes the deterministic part of the leakage and R_i denotes a random noise variable. As stated in Section 3.1, this encoding has a probing security of order $d = n - 1$. However, it has been noticed in [38] that the actual security of the encoding can be higher than d if the leakage function is linear in the bits of the variable. That is, in this case, information on the secret can be only obtained by estimating a statistical moment d' of the leakage distribution, with $d' \geq d$. This can be

intuitively explained by the public vector $\mathbf{l}$ that mixes the bits of the different shares, as opposed to Boolean masking where knowing the first bit of each share directly reveals the first bit of the secret. More formally, the security amplification property of the inner product masking is given by Proposition 2.

Proposition 2 *Let $\mathbf{s}_{MIX} = (s_0, ..., s_{n-1})$ be the n shares of the IP encoding vector defined by the public vector $\mathbf{l} = (1, l_1, ..., l_{n-1})$. If the functions L_i manipulating these shares are linear in the bits of s_i, the bounded moment security order d' of the IP encoding given by $\mathbf{s}_{MIX}$ is equal to the bit-probing security of its equivalent encoding $\mathbf{s}_{MIX2}$.*

Proof. As we assume that the L_i's are linear in the bits of s_i, the leakage L_i can be represented as follows:

$$\begin{aligned} L_i &= \mathsf{L}_i(s_i) + R_i, \\ &= \alpha_i + \alpha_i^0 \times [s_i]_2(0) + ... + \alpha_i^{k-1} \times [s_i]_2(k-1) + R_i, \\ &= \alpha_i^0 \times ([s_i]_2(0) + \frac{\alpha_i}{\alpha_i^0}) + ... + \alpha_i^{k-1} \times [s_i]_2(k-1) + R_i, \\ &= \alpha_i^j \times \mathsf{F}_i^0([s_i]_2(0)) + ... + \alpha_i^{k-1} \times \mathsf{F}_i^k([s_k]_2(0)) + R_i, \end{aligned} \tag{5}$$

with $\mathsf{F}_i^j, j \in [0, k-1]$ a deterministic function in the bit j of s_i and $(\alpha_i, \alpha_i^0, ..., \alpha_i^{k-1}) \in \mathbb{R}^{k+1}$. We can see that the last line of Equation 5 has the same form as Equation 4. As we have n different leakages L_i, each one linearly manipulating k bits, the full leakages $\{L_i\}_{i=0}^{n-1}$ of $\mathbf{s}_{MIX}$ can be viewed as the leakages of a parallel implementation of $\mathbf{s}_{MIX2}$ with $N = n$ cycles, each one manipulating $n_c = k$ single-bit variables. As a result, the bounded security of a serial implementation (called A) of $\mathbf{s}_{MIX}$ is the same as a parallel implementation (called B) of $\mathbf{s}_{MIX2}$ with $N = n$ and $N_c = k$. From [3], we know that proving the bounded security of B is equivalent to proving the probing security of its serial implementation. As the probing security of the serial implementation of B corresponds to the case where one probe can only evaluate one bit, it corresponds to the bit-probing security of $\mathbf{s}_{MIX2}$, which conclude the proof. □

Intuitively, this result simply corresponds to the observation that while probing security at order d implies bounded moment security at order d in case the leakages of the shares are independent (with each share a field element), bit-security at order d' implies bounded moment security at order d' in the case where not only the leakages of the shares are independent (with each share being a field element), but also the different bits of each field element is manipulated independently (which is ensured by the linear leakage assumption). This result implies that, under the linear leakage assumption, maximizing the bounded moment security of the IP encoding $\mathbf{s}_{MIX}$ is the same as maximizing the bit-probing security of $\mathbf{s}_{MIX2}$. The next step is thus to find the best public vectors $\mathbf{l}$ so that the bit-probing security of $\mathbf{s}_{MIX2}$ is maximized.

5 Searching for good codes

As shown in Proposition 1, the key parameter characterizing the bit-probing security is the dual distance of the linear code $\mathcal{D}$ with generator matrix:

$$\mathbf{H} = \begin{pmatrix} \mathbf{L}_1 & \mathbf{I}_k & \mathbf{0}_k & \dots & 0_k \\ \mathbf{L}_2 & 0_k & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_k \\ \mathbf{L}_{n-1} & 0_k & \dots & 0_k & \mathbf{I}_k \end{pmatrix}, \tag{6}$$

where $\mathbf{I}_k$ is an identity matrix with dimension k and $\mathbf{L}_i$ is the matrix representation of a finite field element l_i. Therefore, we have the following proposition.

Proposition 3 *The problem of searching for instantiations of an IP masking scheme with good bit-probing security is equivalent to that of searching for an $[nk, k]$ linear code $\mathcal{C}_g$ over $\mathbb{F}_2$ with large minimal distance and generator matrix:*

$$\mathbf{G}_g = \begin{pmatrix} \mathbf{I}_k & \mathbf{L}_1^T & \mathbf{L}_2^T & \dots & \mathbf{L}_{n-1}^T \end{pmatrix}. \tag{7}$$

The best possible linear codes with a small dimension k (e.g., $k \leq 8$) are well-studied in literature, see [23, 37, 4]. Therefore, the minimal distance of $\mathcal{C}_g$ can be upper-bounded. Moreover, the sub-matrix $\mathbf{L}_i^T$ in $\mathbf{G}_g$ is connected to the underlying irreducible polynomial $g(x) \in \mathbb{F}_2[X]$, which defines the field extension from $\mathbb{F}_2$ to $\mathbb{F}_{2^k}$.

We now consider the practically-relevant case study of implementing the AES securely, i.e., when $k = 8$ and we use the AES polynomial $x^8 + x^4 + x^3 + x + 1$. In the following subsection, we show that one can choose the l_i's to form an IP masking scheme with bit-probing security that is close to the upper bound defined by the best possible eight dimensional binary linear codes.

5.1 Application: 8-bit implementation of the AES

The problem of determining the largest possible minimum distance of an eight dimensional binary linear code is settled in [4], i.e., it is equal to or slightly smaller than the distance defined by the Griesmer Bound [24].

The companion matrix (see [28]) of $g(x) = x^8 + x^4 + x^3 + x + 1$ is defined as:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & \dots & 0 & 0 \end{pmatrix}, \tag{8}$$

whose last row is of the form (1 1 0 1 1 0 0 0). Thus, all the possible field elements of $\mathbb{F}_{2^8}$ can be enumerated as:

$$\sum_{j=0}^{7} a_j \mathbf{A}^j,$$

for all $\mathbf{a} = (a_0, a_1, \ldots, a_7) \in \mathbb{F}_2^8$. We next use three approaches for finding good linear codes with generator matrix satisfying the constraint in Equation 7.

Exhaustive search: When n is small, i.e., less than 4, we can do a brute-force search. That is, we enumerate all possible generator matrices $\mathbf{G}_g$ with the same form as that in Equation (7), and then test its minimum distance.

Random search: We choose $\mathbf{L}_i$ at random to construct $\mathbf{G}_g$ and then test its minimum distance.

Inductive search: We construct good $[8n, 8]$ linear codes satisfying Equation (7) from good $[8(n-n_0), 8]$ linear codes satisfying Equation (7), where n_0 is a small positive integer (e.g., $1, 2, 3$, or 4). That is, we fix the first $8(n-n_0)$ columns of $\mathbf{G}_g$ as the found generator matrix of a good code with length $8(n-n_0)$, exhaust all possible $\mathbf{L}_i$'s, for $i = n-n_0, \ldots, n-1$, and then test its minimum distance.

The numerical results by running Magma are shown in Table 1, where the column n is the number of shares, the column $n \cdot k$ is the code length, the column d_{best}^{IP} is the best minimum distance found from linear codes with a generator matrix satisfying Equation (7), the column d_{best}^{U} is the upper bound derived from the best achievable minimum distance for any $[8n, 8]$ linear codes, and the last column Δ is the difference between the prior two columns (i.e., the gap between IP masking and DSM). It is clear from this table that IP masking can achieve near-optimal bit-probing security if the number of shares is relatively small. Actually, most of the interesting choices of n in practice are covered in this table (since, due to performance reasons, state-of-the-art implementations of IP masking so far did not go beyond 2 or 3 shares). The constructed good codes also show an approach to instantiate IP masking with good bit-probing security. That is, one can determine the finite field elements l_i's from the found generator matrix $\mathbf{G}_g$ corresponding to a good linear code. We did exhaustive search for $n = 2, 3, 4$, random search for $n = 5, 6$, and inductive search for $n = 7$. Therefore, we cannot rule out the possibility of finding a linear code to reach the upper bound when $n \geq 6$ with more computational efforts.[1]

Concretely and as an example, this table shows that when considering IP masking with three shares, the standard probing model guarantees a security order 2. In case the shares only give rise to linear leakages, the bit-level probing model guarantees a security of order 7.

6 Experimental validation

The previous positive results admittedly (highly) depend on a physical assumption (i.e., linear leakages) that may not be perfectly respected. So in order to validate the theoretical results,

[1] Note that the inductive search allows us to find good linear codes with relatively large minimum distance rather quickly. For instance, we can easily obtain a desired $[72, 8, 30]$ linear code by the inductive search with the code length $8n$ increasing by 16 gradually from 24 to 72. The gap Δ here is only -2. This task takes less than 2 minutes when using the online Magma calculator, while it is almost intractable by the other two approaches even running on a powerful local Magma server.

Table 1: The best linear codes corresponding to an IP masking scheme found by Magma. The field extension from $\mathbb{F}_2$ to $\mathbb{F}_{2^k}$ is defined by the AES polynomial.

n	$n \cdot k$	d^{IP}_{best}	d^{U}_{best}	Δ
2	16	4	5	-1
3	24	8	8	0
4	32	12	13	-1
5	40	16	16	0
6	48	21	22	-1
7	56	23	24	-1

we now consider a practical security evaluation of an IP encoding implementation. In this respect, this case study comes with the (usual) cautionary note that the only thing we show next is that there exist implementations for which (essentially) linear leakages are observed for certain samples. This does not imply that the security order amplification can be observed for full implementations (which, as mentioned in the introduction, is left as an important scope for further research). Yet, it shows that this security order amplification can at least be used to reduce the amount of leaky samples in an implementation and/or their informativeness.

6.1 Target implementation

Our experiments are performed on a 32 bits ARM Cortex-M4 microcontroller using the Atmel SAM4C-EK evaluation kit running at 100 MHz.[2] We implemented the IP encoding using two shares. We performed the trace acquisition using a Lecroy WaveRunner HRO 66 ZI oscilloscope running at 500 megasamples per second. We monitored the voltage variation using a 4.7 Ω resistor set in the supply circuit of the chip. For each execution and a given value of l_1, we select a random secret x, a random value s_1 and compute the encoding $\mathbf{s} = (s_0 = x + l_1 \cdot s_1, s_1)$. We acquire the leakages by triggering the measurement prior to the successive load of these two shares s_0 and s_1 into the memory.

6.2 Analysing the leakages

Leakage detection. Our first experiments aim at analyzing how the device leaks. As a preliminary, we start by identifying the points of interest that corresponds to the manipulation of s_0 and s_1 (in an evaluator-friendly setting where we know these values). Figure 1 shows the result of the correlation between the different time samples with the Hamming weight of s_0 (left) and s_1 (right) using 40,000 traces. Our following analyses only focus on the two samples giving the maximum correlation for both shares. We refer to the time sample corresponding to the manipulation of s_0 (resp. s_1) as t_0 (resp. t_1).

Linear regression. The theoretical results on the security order amplification of Section 4 rely on the assumption that the leakage function is linear. As this assumption is hardware-dependent, we first evaluated the linearity of the leakages produced by our target. For a given

[2] http://www.atmel.com/tools/SAM4C-EK.aspx.

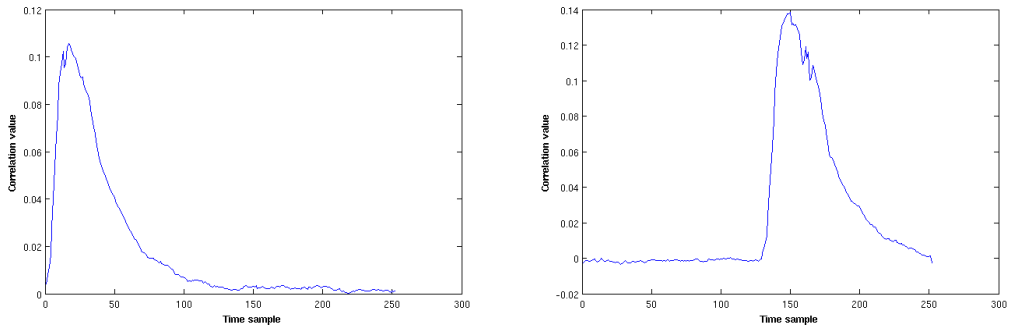

Fig. 1: Detection of points of interest.

time sample, linear regression is perfectly suited for this purpose [34]: it allows estimating how the manipulated data is leaked at the bit level. Denoting by $\mathsf{L} : \mathbb{F}_2^8 \rightarrow \mathbb{R}$ the deterministic part of the actual leakage function, a linear regression of degree q gives the function $\hat{\mathsf{L}}_q$ that approximate L by using bit combinations of degree up to q. As a results, it is a suitable tool to estimate the linearity of the leakages, by just comparing regressions of degree 1 and 2. The description of the resulting $\hat{\mathsf{L}}_1$ and $\hat{\mathsf{L}}_2$ approximations are given by Equation 9. The coefficients α, α_i and $\alpha_{i,j}$ belong to $\mathbb{R}$ and are the results of the linear regression:

$$\begin{aligned} \hat{\mathsf{L}}_1(x) &= \alpha + \sum_{i=0}^{7} \alpha_i \times [x]_2(i) \\ \hat{\mathsf{L}}_2(x) &= \alpha + \sum_{i=0}^{7} \alpha_i \times [x]_2(i) + \sum_{i=0}^{6} \sum_{j=i+1}^{7} \alpha_{i,j} \times [x]_2(i) \times [x]_2(j) \end{aligned} \tag{9}$$

Using the same traces as for the points of interest detection, we computed the linear regression at t_1 using both a linear ($\hat{\mathsf{L}}_1$) and a quadratic ($\hat{\mathsf{L}}_2$) basis (t_0 gave same results and is thus omitted). The left (resp., right) part of Figure 2 shows the resulting coefficients for the linear (resp., quadratic) basis. The first value indexed by 0 corresponds to the offset α. The next 8 values indexed from 1 to 8 are the linear coefficient α_i. Finally, the quadratic coefficients $\alpha_{i,j}$ are indexed from 9 to 36 (only in the right figure). We can see that the linear terms are significantly more dominant than the quadratic ones. As a result, it provides some confidence that our target (samples) are good candidates for the linear leakages assumption.

Mutual information. Evaluating the linearity of the leakage function by only looking at the coefficients of $\hat{\mathsf{L}}_1$ and $\hat{\mathsf{L}}_2$ has two drawbacks. First, it is hard to judge if the models have converged. Secondly, we cannot know if the small values given by the quadratic coefficients are significant or come from estimation errors. In order to get rid of these two problems and push the analysis one step further, we compute the perceived information introduced in [16] arising from $\hat{\mathsf{L}}_1$ and $\hat{\mathsf{L}}_2$. The latter metric essentially captures the amount of information that can

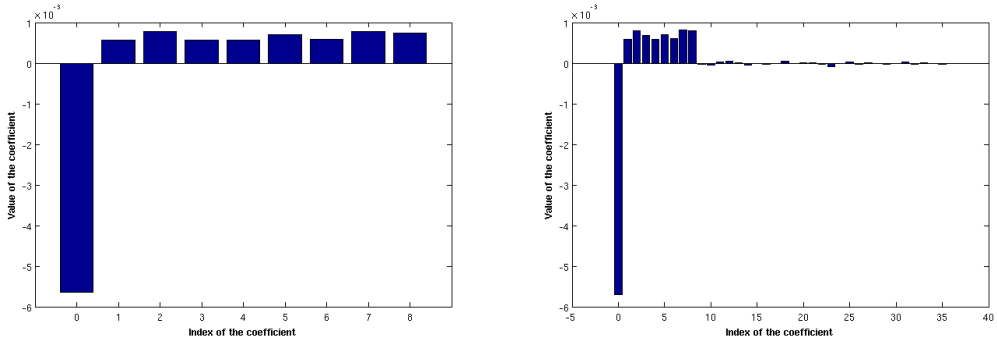

Fig. 2: Linear regression results.

be extracted from a model, possibly biased by estimation and assumption errors. (Because of place constraints, we refer to this previous work for the details).

Figure 3 shows this perceived information for the linear model $\hat{\mathsf{L}}_1$ and the quadratic model $\hat{\mathsf{L}}_2$ in function of the number of traces used for the estimation of the model. As expected, the quadratic model needs more samples to converge as it is more complex. Interestingly, we can see that both models converge towards approximately the same value. This now formally confirms that the quadratic model does not bring significantly more information than the linear one in our setting. As a consequence, we deduce that the true leakages of our target are close to linear (and therefore that it is a good candidate to benefit from security order amplification).

6.3 Concrete security assessment

We now present additional results of concrete security analyses performed on our implementation. For this purpose, and in order to directly evaluate whether the security order of our IP encoding was amplified, we aim at detecting the lowest statistical moment in the leakages that reveals information on the secret. To do so, we apply the ρ-test with K-fold cross-validation as described in [15]. Note that in order to limit the (high) data requirements for this last experiment, we used the trick proposed in [36], Section 3.2 and performed a preliminary averaging of our traces (assuming mask knowledge) before trying to detect higher-order statistical dependencies. Namely, we used a 60x averaging for the second-order detections and 100x averaging for the third-order ones.

Correlation-test. Given a leakage L, the ρ-test allows detecting a mean dependency between L and the secret x. The first step is to estimate a model from a profiling set $\mathcal{L}_{prof}$ of N_{prof} leakage samples on L. This model corresponds to the average leakage on L for each value of the secret x. The next step is to use this model on a test set $\mathcal{L}_{att}$ of N_{test} samples. We compute the correlation r between $\mathcal{L}_{test}$ and our model applied on the secret values used to

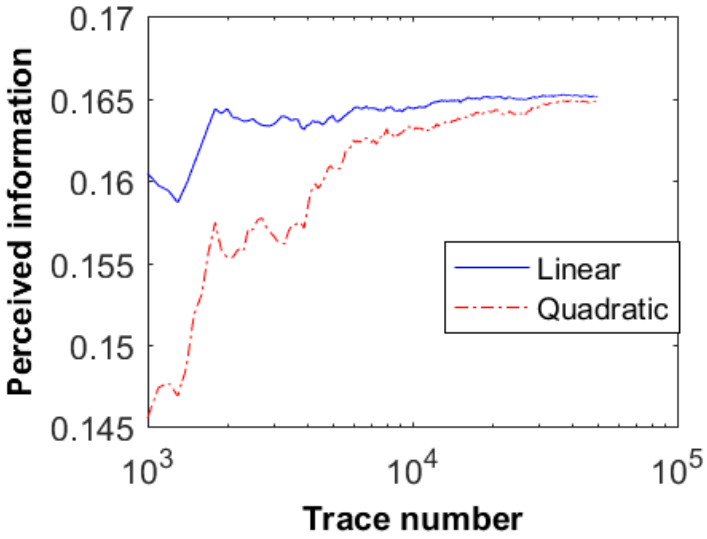

Fig. 3: Perceived information from linear and quadratic leakage models.

generate $\mathcal{L}_{test}$. We then compute the normalized Fisher's z-transformation on r:

$$r_z = \frac{\sqrt{N_{test} - 3}}{2} \times \ln\left(\frac{1+r}{1-r}\right), \tag{10}$$

Where $\ln$ denotes the natural logarithm. The obtained value r_z can be approximately interpreted as following a normal distribution with mean 0 and variance 1. As in [15], we set the confidence threshold of r_z that detects the presence of a dependency to 5.

***K*-fold cross validation.** We use a 4-fold cross validation in order to reduce the variability of the ρ-test. That is, we acquire set $\mathcal{L}$ of N leakages that we partition in 4 independent subsets $\mathcal{L}_i, i \in [1, 4]$ of equal size. We then apply the ρ-test 4 times by using a different test set each time (more precisely, iteration i uses $\mathcal{L}_i$ as a test set and $\cup_{j \neq i} \mathcal{L}_j$ as profiling set).

Evaluation results. In our first (reference) experiment, we used l_1 equal to 1, which is equivalent to a Boolean masking encoding. The corresponding DSM representation is such that the dual distance of D is equal to two. As we expect a second-order dependency, we apply the ρ-test on the center product $L = (L_{t_1} - \bar{L}_{t_1}) \cdot (L_{t_2} - \bar{L}_{t_2})$, where $\bar{L}_{t_1}$ (resp. $\bar{L}_{t_2}$) denotes the sample mean of L_{t_1} (resp. L_{t_2}). Figure 4 shows the result of the ρ-test with 4-fold cross validation. The x-coordinate shows the number of average traces being used, and the y coordinate shows the confidence value. The black curves is the line $y = 5$ that shows the confidence threshold. Each of the remaining 4 curves represents one of the cross validation tests. As expected, we quickly detect a second order leakage after roughly 5,000 average traces.

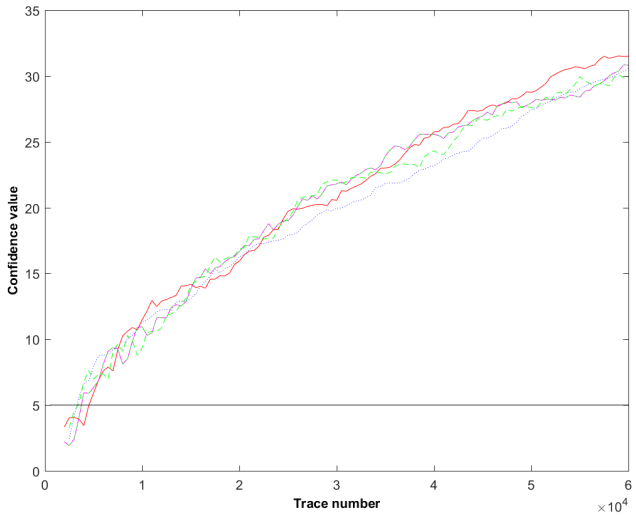

Fig. 4: Results of the ρ-test for IP masking with $l_1 = 1$.

As a second experiment, we set $l_1 = 3$, which is the hexadecimal representation of the polynomial $x + 1$. The corresponding ODSM representation is such that the dual distance of D is equal to three. That is, we expect the lowest statistical moment that gives information on the secret to be equal to three, thus having a security order amplification. We verify this in two steps. First, we apply the ρ-test on the center product as in the previous experiment to detect if any second-order dependency can be seen. Secondly, we apply the ρ-test on a new center product $L = (L_{t_1} - \bar{L}_{t_1}) \cdot (L_{t_2} - \bar{L}_{t_2}) \cdot (L_{t_2} - \bar{L}_{t_2})$ to detect the presence of a third-order dependency. The left (resp., right) part of Figure 5 shows the results of the ρ-test with 4-fold cross validation for the second-order (resp., third-order) test. Again, the x-coordinate represents the number of average traces, the y coordinates the confidence value and the black curve the confidence threshold. As we can see on the left part of the figure, no second-order leakages are detected with up to 100,000 average traces. However, the right part of the figure shows a third-order dependency around 60,000 average traces. This confirms both the high linearity of the leakages of this chip and the relevance of the theoretical investigations in Section 4.

Discussion. To conclude, we emphasize that the results of the ρ-test experiment for $l_1 = 3$ were based on 100x averaged traces, leading to a Signal-to-Noise Ratio close to 1 (which is out of the noise levels where masking security proofs apply [14]). So this experiment does not formally prove that no second-order dependency could appear for this noise level (without averaging). In this respect, we recall that this choice was motivated by time constraints (without averaging, we could not detect third-order dependencies either). Besides, and in view of the leakage analysis in Section 6.2, we are confident that the security order amplification

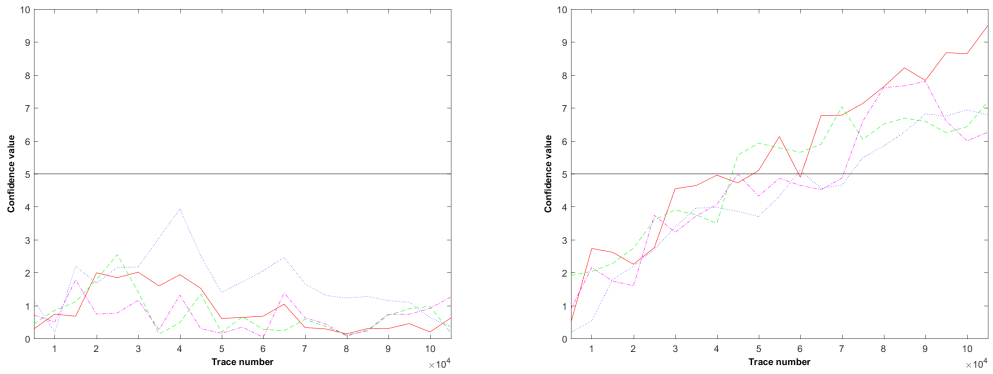

Fig. 5: Results of the ρ-test for IP masking with $l_1 = 3$

put forward in this last section does actually correspond to our theoretical expectations with (close enough to) linear leakages.

Acknowledgements. François-Xavier Standaert is a research associate of the Belgian Fund for Scientific Research. This work has been funded in parts by the European Commission through the H2020 project 731591 (acronym REASSURE), the CHIST-ERA project SECODE and the ERC project 724725 (acronym SWORD).

References

1. Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. Inner product masking revisited. In Oswald and Fischlin [30], pages 486–510.
2. Josep Balasch, Sebastian Faust, Benedikt Gierlichs, and Ingrid Verbauwhede. Theory and practice of a leakage resilient masking scheme. In *ASIACRYPT 2012*, pages 758–775, 2012.
3. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Coron and Nielsen [11], pages 535–566.
4. Iliya Bouyukliev, David B. Jaffe, and Vesselin Vavrek. The smallest length of eight-dimensional binary linear codes with prescribed minimum distance. *IEEE Trans. Information Theory*, 46(4):1539–1544, 2000.
5. Julien Bringer, Claude Carlet, Hervé Chabanne, Sylvain Guilley, and Houssem Maghrebi. Orthogonal direct sum masking - A smartcard friendly computation paradigm in a code, with builtin protection against side-channel and fault attacks. In *WISTP 2014*, pages 40–56, 2014.
6. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage squeezing of order two. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 120–139. Springer, 2012.
7. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage squeezing: Optimal implementation and security evaluation. *J. Mathematical Cryptology*, 8(3):249–295, 2014.

8. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, Houssem Maghrebi, and Emmanuel Prouff. Achieving side-channel high-order correlation immunity with leakage squeezing. *J. Cryptographic Engineering*, 4(2):107–121, 2014.
9. Claude Carlet and Sylvain Guilley. Complementary dual codes for counter-measures to side-channel attacks. *Adv. in Math. of Comm.*, 10(1):131–150, 2016.
10. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO '99*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.
11. Jean-Sébastien Coron and Jesper Buus Nielsen, editors. *EUROCRYPT 2017*, volume 10210 of *LNCS*, 2017.
12. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 410–424. Springer, 2013.
13. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Nguyen and Oswald [29], pages 423–440.
14. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In Oswald and Fischlin [30], pages 401–429.
15. François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In *EUROCRYPT 2016*, pages 240–262, 2016.
16. François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In Nguyen and Oswald [29], pages 459–476.
17. Stefan Dziembowski and Sebastian Faust. Leakage-resilient cryptography from the inner-product extractor. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 702–721. Springer, 2011.
18. Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine masking against higher-order side channel analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *LNCS*, pages 262–280. Springer, 2010.
19. Laurie Genelle, Emmanuel Prouff, and Michaël Quisquater. Thwarting higher-order side channel analysis with additive and multiplicative maskings. In *CHES 2011*, pages 240–255, 2011.
20. Jovan Dj. Golic and Christophe Tymen. Multiplicative masking and power analysis of AES. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 198–212. Springer, 2002.
21. Louis Goubin and Ange Martinelli. Protecting AES with shamir's secret sharing scheme. In *CHES 2011*, pages 79–94, 2011.
22. Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In Coron and Nielsen [11], pages 567–597.
23. Markus Grassl. Tables of linear codes and quantum codes. `http://http://www.codetables.de/`, 2015. [Online; accessed 25-April-2017].
24. James H Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, 1960.
25. Vincent Grosso, Emmanuel Prouff, and François-Xavier Standaert. Efficient masked s-boxes processing - A step forward -. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 2014*, volume 8469 of *LNCS*, pages 251–266. Springer, 2014.
26. Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low entropy masking schemes, revisited. In Aurélien Francillon and Pankaj Rohatgi, editors, *CARDIS 2013*, volume 8419 of *LNCS*, pages 33–43. Springer, 2013.
27. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO 2003*, pages 463–481, 2003.
28. R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of mathematics and its applications. Addison-Wesley Publishing Company, Advanced Book Program/World Science Division, 1983.

29. Phong Q. Nguyen and Elisabeth Oswald, editors. *EUROCRYPT 2014*, volume 8441 of *LNCS*. Springer, 2014.
30. Elisabeth Oswald and Marc Fischlin, editors. *EUROCRYPT 2015*, volume 9056 of *LNCS*. Springer, 2015.
31. Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 142–159. Springer, 2013.
32. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.
33. Thomas Roche and Emmanuel Prouff. Higher-order glitch free implementation of the AES using secure multi-party computation protocols - extended version. *J. Cryptographic Engineering*, 2(2):111–127, 2012.
34. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *CHES 2005*, pages 30–46, 2005.
35. Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.
36. François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. *IACR Cryptology ePrint Archive*, 2017:138, 2017.
37. CA van Tilborg Henk. The smallest length of binary 7–dimensional linear codes with prescribed minimum distance. *Discrete Mathematics*, 33(2):197–207, 1981.
38. Weijia Wang, François-Xavier Standaert, Yu Yu, Sihang Pu, Junrong Liu, Zheng Guo, and Dawu Gu. Inner product masking for bitslice ciphers and security order amplification for linear leakages. In *CARDIS 2016*, pages 174–191, 2016.