Statistical properties of side-channel and fault injection attacks using coding theory

Claude Carlet, Sylvain Guilley

Statistical properties of side-channel and fault injection attacks using coding theory

Claude Carlet[1] and Sylvain Guilley[2,3,4]

[1] LAGA, University of Paris 8
(and Paris 13 and CNRS), Saint-Denis Cedex 02, France.
ORCID: 0000-0002-6118-7927
E-mail: claude.carlet@univ-paris8.fr

[2] Secure-IC S.A.S., 15 Rue Claude Chappe, Bât. B,
35 510 Cesson-Sévigné, France.
ORCID: 0000-0002-5044-3534
E-mail: sylvain.guilley@secure-ic.com

[3] LTCI, Télécom ParisTech, Université Paris-Saclay,
75 013 Paris, France.

[4] École Normale Supérieure, Département d'Informatique,
75 005 Paris, France.

Abstract. Naïve implementation of block ciphers are subject to side-channel and fault injection attacks. To deceive side-channel attacks and to detect fault injection attacks, the designer inserts specially crafted error correcting codes in the implementation. The impact of codes on protection against fault injection attacks is well studied: the number of detected faults relates to their minimum distance. However, regarding side-channel attacks, the link between codes and protection efficiency is blurred. In this paper, we relate statistical properties of code-based countermeasures against side-channel attacks to their efficiency in terms of security, against uni- and multi-variate attacks.

Key words: detection of faults, masking countermeasure, statistics of leakage, uni- and multi-variate side-channel attacks, high-order attacks, probing security model, bounded moment security model, inner product masking, leakage squeezing masking.

1 Protection as a coding problem

Cryptographic algorithms are subject to attacks aiming at extracting their keys. When the adversary has access to the device, he is able to target the implementation of the cryptographic algorithm. Two attack paths customarily encountered are side-channel attacks (where the attacker reads some leakage from the implementation when it is running), and fault injection attacks (where the attacker modifies some intermediate variables inside of the implementation).

In this article, we analyze algorithmic protections combining both side-channel prevention and fault injection detection. We survey security models for a given set of security parameters. In general, several such models can be defined, each addressing a particular kind of attacker. The equivalence or even the reduction of security models is hard and is currently at the core of intensive researches. However, when one focuses on one specific implementation operated in a given context, then security notions can be clarified (e.g., be shown equivalent).

In this paper, we focus on a protection against side-channel and fault injection attacks where the state of the cryptographic algorithm is encoded. From the security model and its parameters, we can thus derive desirable protection properties. Those result from a statistical analysis of the leakage in the presence of countermeasures.

Contributions. Regarding side-channel analysis protections, we identify that the *inner product masking scheme* [3,57] is an instance of the *leakage squeezing* (see [37,36,36,17,18] for 2 shares, and [15] for strictly more than 2 shares) protection using *linear bijections* (Sec. 6.4). The papers about *inner product masking scheme* explain the engineering aspects related to secure computation of finite field laws (addition and multiplication), whereas papers about *leakage squeezing* highlight the accurate security level of the data representation. In this article, we bridge the gap by showing how to design inner product masking schemes with quantifiable security level against bit-level side-channel attacks. For the first time, we relate the dual distance of the code used in the countermeasure, the mutual information between sensitive variable and leakage, and the attack success rate.

A second contribution of this paper is to analyze joint side-channel and fault attacks protections. Specifically, we emit a warning: fault protections and side-channel protections can happen to combine nicely, provided a careful analysis of their combined implementation is carried out (Sec. 6.3). Without such analysis, the combination can be destructive security-wise.

Eventually, we expose a novel method to derive Boolean codes from codes over $\mathbb{F}_{2^k}$ (Sec. 7).

Outline. The rest of the paper is structured as follows. We start in Sec. 2 by explaining how error correcting codes can provide a protection against both side-channel and fault injection attacks. Then, we review in Sec. 3 existing security models, and select some of them. Relevant security parameters are given in Sec. 4. The impact on the protections architecturing is then analyzed in Sec. 5. Known constructions are revisited in Sec. 6, and a new one is given in Sec. 7. Our contributions beyond the state-of-the-art are in Sec. 6 and 7. Eventually, conclusions are in Sec. 8.

2 Introduction

2.1 Principle of coding

One purpose of codes is to detect (and correct) errors. Another purpose is to allow multiple users to use the same channel without interference, while maximizing the use of its capacity. In the context of protections against side-channel attacks, one user will be the cryptographic computation, and the other ones are noisy sources, aiming at making the leakage passing through the channel as difficult to interpret as possible for an eavesdropper. Clearly, this dual use of codes allows to kill two birds with the same stone, which makes it appealing.

Let us insist more in detail on the protection against side-channel attacks. We denote by $X \in \mathbb{F}_2^k$ (where $\mathbb{F}_2^k$ is $\{0,1\}^k$ equipped with an additive group law, denoted by "$\oplus$") a sensitive variable, we intend to protect. It is usually a word, of bit length k. As usual in statistics, we shall use capital letters (such as X) for random variables, and small letters (such as x) for their realizations. The AES [44] block cipher will be our running example, because it is very widespread in the field and is well studied in academic papers. As AES is byte-oriented, we will consider that every variable can be represented by one or more bytes, hence $k = 8$ bits. In a cryptographic implementation, such variable is leaking some *non-injective* and *noisy* information. The non-injective function is denoted as $\varphi : \mathbb{F}_2^k \to \mathbb{R}$, and N denotes the additive noise. Both are represented in Fig. 1, as well as the leakage $X \rightsquigarrow \varphi(X) + N$.

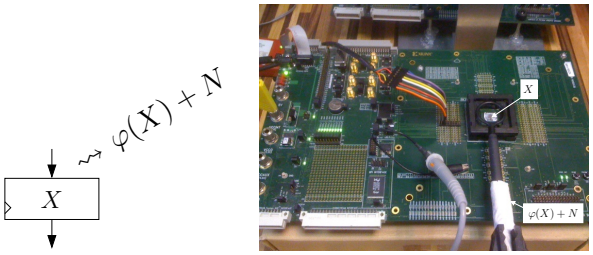

Fig. 1. Leakage arising from the manipulation of variable X

Typically, φ is an *extensive* function (that is, it is the weighted sum over $\mathbb{R}$ of its coordinates), such as the *Hamming weight* (denoted as w_H). This model is attested in many devices, such as smartcards, whose leakage is analyzed in Fig. 2 [30].

To protect against straightforward analysis of leakage, *masking* countermeasure has been initially presented (by Thomas S. Messerges [39]) as a two-step process:

1. the algorithmic parameters (e.g., substitution boxes) are recomputed for a given mask (randomly chosen) by replacing each *sensitive* data X by $(X \oplus$

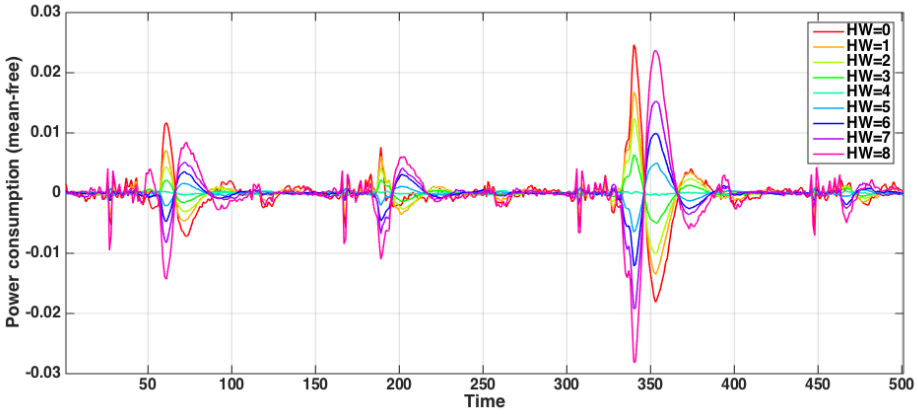

Fig. 2. Decomposition of the leakage per value of $\varphi(X) = w_H(X) \in \{0, \ldots, 8\}$

$\bigoplus_{i=1}^{t} Y_i, Y_1, Y_2, \ldots, Y_t)$, where t is some security parameter and where the Y_i's are chosen randomly independently in the same additive group $\mathbb{F}_2^k$ as X, and then

2. the masked algorithm is executed with masked plaintext and masked key as inputs.

While this strategy works well from a theoretical point of view, some criticisms have emerged over time:

- from a security point of view, it has been noted that the recomputation stage algorithm (which does not depend on the key) leaks a lot of information which can be combined in a constructive way with the algorithm execution (where *masked* sensitive data, that is data which depend on the key and on inputs/outputs known by the attacker, are used), and that such attack path is hard to counter [46,54,13],
- from a performance point of view, the recomputation takes a longer time[1] than the execution of the recomputed algorithm, which obviously limits the advantage of such solution.

Therefore, solutions which are free from the preliminary recomputation stage are favored in practice in many applications (except low-cost smartcard, which do not have enough resources to get rid of the security-wise weak table recomputation stage). Historically, the data and the masking material are processed together during the execution of the algorithm. For instance, in the case where

[1] For instance, in the AES block cipher, the substitution box has 256 entries, hence recomputation requires 256 memory accesses. The number of substitution box calls in the algorithm is 16 (resp. 4) per round for the datapath (resp. key schedule), hence a total of $(16 + 4) \times 10 = 200$ calls, which is indeed less than 256.

$t = 1$ above, the computation is organized by duplicating the state: one half contains the masking material $Y \in \mathbb{F}_2^k$, whereas the second half contains the masked data $X \oplus Y \in \mathbb{F}_2^k$. This is illustrated in Fig. 3. It shall be noted that the leakage is

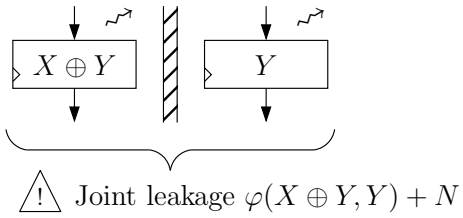

Fig. 3. Leakage arising from the manipulation of the masked variable $X \oplus Y$ and of its (single) mask Y, here of same size $k = 8$ as that of X

now bi-variate, hence harder to exploit by the attacker, because the latter must combine two values to recover useful information. However, some implementations manage to handle $X \oplus Y$ and Y side-by-side; when the non-injective leakage function φ is extensive, we thus have $\varphi(X \oplus Y, Y) = \varphi(X \oplus Y) + \varphi(Y)$, hence it is convenient to describe the masking as an encoding of (X, Y). Namely, the sensitive variable X is encoded by a linear code of generating matrix $(I \parallel 0)$, the mask is encoded using the repetition code of generating matrix $(I \parallel I)$, where I is the identity matrix in $\mathbb{F}_2^k$, and these two codewords are added together.

Thus, we see that protection against side-channel attack can also be expressed in terms of codes. In the former example, the two binary codes are:

1. C, of parameters $[n = 2k, k, 1]$, of generating matrix $(I \parallel 0)$, and
2. D, of parameters $[n = 2k, k, 2]$, of generating matrix $(I \parallel I)$,

such that any element $Z = (X \oplus Y, Y) \in \mathbb{F}_2^n$ is the direct sum of the encoding of X through C and of Y though D.

This approach of coding is well suited to the physical leakage as represented in Fig. 1, since side-channel analysis can be reinterpreted as a decoding problem: the aim of the attacker is the recovery of X after its encoding with masks, and transformation through the non-injective (owing to φ) and noisy (owing to N) leakage function. Notice that high-order masking schemes are detailed in greater details under the view of coding theory in Sec. 6.1.

However, we stress that the attacker has other means to recover information on Z (X after coding):

– with a **probing station**, the attacker is able to read and/or write selected bits,

– on multicore platforms running an operating system, **cache hit/miss**[2] **probing** can be used as an attack, especially if data are used as addresses to memories.

2.2 Design choice

In the previous section, we took the example (recall Fig. 3) of mask (Y) and information (X) of same bitwidth. However, we have already seen that this can be more general, with a value of t larger than 1 in traditional masking. Typically, Y can be made smaller (as small as 1 bit, e.g., in [53,8,39]—for instance, in [39], a 1-bit masking is used to perform a Boolean to arithmetic transform.) But also, for enhanced security, Y can be larger than X, especially in so-called high-order masking schemes [51][3]. The general encoding using linear codes of X is as follows:

$$Z = XG \oplus YH, \tag{1}$$

where:

– G is the generating matrix of a code of length n and of dimension k, and
– H is the generating matrix of a code of length n and of dimension $(n-k)$.

Typically, $k = 8$ bits for AES. For high-order protections, the masks are used as multiple k bit words. Therefore, a typical study will be that $(n-k)$ is a multiple of k.

However, probing attacks do target individual bits.

Therefore, we will consider two kinds of codes: codes on $\mathbb{F}_{2^k}$ and codes on $\mathbb{F}_2$. Notice that a code on $\mathbb{F}_{2^k}$ can be *expanded* on $\mathbb{F}_2$. In MAGMA [55], this operation can be realized on a code `C` easily using command

```
C_expanded := SubfieldRepresentationCode(C, GF(2));.
```

If `C` has parameters $[n, k, d]_{2^m}$, then `C_expanded` has parameters $[mn, mk, d']_2$, where $d' \geq d$. A concrete example will be given in Sec. 7.

3 Security models

3.1 Side-channel analysis

Masking consists in adding some randomness in the computations, which forces the attacker to perform a *high-order* attack, process during which several leakage sources are combined. In turn, if the leakage samples are noisy, the combination results in a so-called *noise amplification*.

There are mainly two security models:

[2] Systems with multiple processors speed up memory accesses using data and instruction memory caches, which are shared by the processors; if a data which is not in the cache memory is fetched, then there is a cache miss (which takes a long time) otherwise, there is a cache hit (which is fast). Thus the hit/miss patterns betray the memory access sequence.

[3] Beware that the high-order implementation in this publication is flawed. For fixes, please refer to [20].

- **Probing model** (cf. Sec. 3.1), as in [31] (and many other papers [9,49,35,22] which stem from this seminal publication).
- **Bounded moment model** (cf. Sec. 3.1), initially defined in [§4][43], and then reintroduced in [6].

Probing model The probing model states the following:

Definition 1 (Probing model). *A masking scheme is secure at order t in the probing model if no tuple of t intermediate variables depends on the secret.*

An unprotected implementation is secure at order $t = 0$ (recall Fig. 2). A protected implementation is secure at order $t > 1$.

When the algorithm handles bitvectors (elements of $\mathbb{F}_2^k$), there is an ambiguity whether the definition 1 refers to intermediate variables as *bitvectors* or as *individual bits*. Thus, in the sequel, we shall clarify this point when talking about the probing model.

An automated method to test for the security of an algorithm with respect to this model at bitvector-level is given in [5,4].

Bounded moment model The bounded moment model states the following:

Definition 2 (Bounded moment model). *A masking scheme is secure at order t in the bounded moment model if no moment of degree t in the intermediate variables depends on the secret.*

With this definition, we also have that an unprotected implementation is secure at order $t = 0$, while a protected implementation is secure at order $t > 1$.

The definition 2 has initially been introduced in the context of low entropy masking schemes (LEMS [43,8]). The concept has been recovered independently [41,42] by noting that attacks at many orders are possible, but that in usual situations (see exception in [13]), the lowest order is the most successful.

Reductions between leakage security models are studied in [6]. When probing model and bounded moment models are considered at the *bit level*, then *they are equivalent* (see Theorems 9 and 10 of [29]).

3.2 Fault injection analysis

Protection of block ciphers with codes is a topic which has been studied for a long time [34,2]. Basically, the security metric relates to the code error detection probability. However, we notice that few constructs have been tackling simultaneously protection against *both* side-channel *and* fault injection analyses.

3.3 Combination of side-channel and fault injection

The ODSM countermeasure (to be analyzed at Sec. 6.3) is the first joint protection against side-channel and fault injection analyses. Carlet et al. noticed

that masks are not sensitive by themselves (in that they do not leak information "standalone"); thus faults can be detected by verifying that masks have not been altered. This strategy is all the more relevant in first-order masking schemes, where the security can be attained by reusing the same mask throughout the algorithm to protect, hence the possibility to perform the integrity check at any arbitrarily chosen time while the algorithm unfolds. A careful warning is nonetheless formulated in Sec. 6.3.

4 Security parameters

Security at order *one* is nowadays considered insufficient for most practical operational environments. Indeed, many attacks at first order (such as second-order correlation power analysis [40], collision-correlation [25], MIA [7], etc.) are known and well mastered by most adversaries.

Regarding fault injection attacks, it is known that very powerful exploitation techniques exist for block ciphers [32]. Thus, once again, detecting a *single* fault is insufficient.

However, it shall be noted that some *palliative* countermeasures are usually implemented in addition to the two abovementioned *curative* countermeasures. Palliative countermeasures consist typically in artificial insertion of horizontal noise (desynchronized start date, random interrupts, dummy decoil operations, etc.), which makes the step for succeeding higher-order attacks drastically high.

Concluding, second-order resistance ($t \geq 2$) to both side-channel analysis and fault injection resistance is, in most case, sufficient if well complemented by other protection means, in a construction denoted by *defense in depth*.

5 Architectural options for protection

Protecting against both side-channel and fault injection attacks can resort to the *masks verification* strategy of ODSM. But more generally, it can be imagined to implement orthogonal protections one of top of each other. Both approaches have pros and cons:

- *encode then mask* suffers no security issue. Indeed, encoding does increase the data bitwidth while making the encoded data redundant, thus reducing the density of the new sensitive variable. However, this does not cause any security issue as masking remains secure even if the variable to protect is not uniformly distributed (which is the case because the sensitive variable here belongs to a codebook). The "encode then mask" suffers more performance than security issues: the sensitive variable, after encoding, encounters a blow-up in size corresponding to the inverse of the code rate. After application of the side-channel protection, this overhead is multiplied by the order of the masking scheme.
- *mask then encode* is thus more efficient in terms of variable size growth. But care must be taken on the way the redundancy is applied. Indeed, linear

codes consist in computing some redundancy on top of the masked data, and this redundancy is a linear transformation. It is well known that some linear transformations destructively combine with the masking: e.g., the addition of all shares clearly completely unmasks the masked data. Besides, it becomes non-obvious to compute on an encoded state. The only proposal in this direction is paper [50], which handles security at bit-level.

In addition to those considerations, it shall be noticed that verification can be achieved both at word or at bit levels. Further investigations are left for future considerations.

6 Some known constructions revisited

In this section, we present several masking schemes under the prism of coding theory. We highlight the links between their definition and their security level. The perfect additive masking (Sec. 6.1) is typically word-oriented.

6.1 Perfect additive masking scheme [9]

In this section, we answer the question "*why is masking an encoding?*". Actually, it is straightforward to show that share-based masking schemes (e.g. [49,26]) consist in encodings. We denote by t the order of the masking, and by $d = t + 1$ the number of shares, that are elements of $\mathbb{F}_2^k$. The protection rationale is as follows:

- $x \in \mathbb{F}_2^k$ the clear data,
- $y = (y_1, y_2, \ldots, y_t) \in (\mathbb{F}_2^k)^t$ are the masks, and the protected data is:
- $z = (x \oplus \bigoplus_{i=1}^{t} y_i, y_1, y_2 \ldots, y_t) \in (\mathbb{F}_2^k)^d$.

So we have $n = d \times k = (t + 1) \times k$, and $z = xG \oplus yH$, where

$$G = \begin{pmatrix} I_k & 0 & 0 & \cdots & 0 \end{pmatrix} \qquad \text{and} \qquad H = \begin{pmatrix} I_k & I_k & 0 & \cdots & 0 \\ I_k & 0 & I_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_k & 0 & 0 & \cdots & I_k \end{pmatrix} . \tag{2}$$

Notice that $GH^\intercal \neq 0$, thus the codes generated by G and H are **not** complementary dual [21].

6.2 Inner Product (IP [3]) masking scheme

The perfect masking scheme depicted in Eqn. (2) presents intrinsic weaknesses[4]: for instance, it does not correspond to individual bit masking, but rather word-

[4] The *perfect* masking scheme introduced in 2001 [9] is *perfect* in that it ensures *perfect* independence at word-level between tuples of intermediate variable missing at least one share. However, it is not *perfect* in the sense of bit-level security. Hence the later introduction in 2011 of leakage squeezing masking scheme [37] and in 2015 of inner product masking scheme [3].

wise. Individual bits in the shares can be attacked independently one of the others, thereby enabling k parallel divide-and-conquer mono-bit strategies. Hence there is a need for a *secondary security objective* which is *bit-oriented*. The publications dealing with *inner product masking* [3,57] therefore attempt to shuffle bits within one share. However, both the choice to focus on one share and the code selection method are currently not discussed mathematically in the published literature. Still, there is a way to select linear functions in line with a security objective. This will be made clear in Sec. 6.4 devoted to leakage squeezing countermeasure.

6.3 Orthogonal Direct Sum Masking (ODSM [10])

ODSM refers to the masking scheme where the data to protect is represented as in Eqn. (1).

Implementation An example of implementation of ODSM for AES ($n = 2k = 16$) is given in the appendix B of [11]. These indications shall suffice to reproduce a protected design. The only practical detail to be precised in the implementation is the computation of the linear transformation $\mathbb{F}_2^{16} \to \mathbb{F}_2^{16}$. It can be implemented as a vector-matrix product, as explained in Algorithm 2 of [11]. Thus, there is no need to save a $2^{16} \times 16$ table corresponding to all the values of xL' for $x \in \mathbb{F}_2^{16}$. Besides, if it is wanted all the same to resort to a table-based implementation, it is possible to split the $2^{16} \times 16$ table into tables of size $2^{16} \times 8$ (see Alg. 1 in [29]).

Security against fault injection attacks In ODSM, the transformations (e.g., the call to substitution boxes) are presented as operating in parallel on the whole state $z \in \mathbb{F}_2^n$. It is described in [10] how linear and non-linear operations can be tabulated. When $k|n$, the ODSM scheme can be interpreted as a computation which can be carried out on k-bit words. In this case, one knows that linear operations can be safely implemented as the parallel composition of the linear operation on each of the $d = n/k$ shares. However, this should not be understood as the fact that arbitrary linear operations can be securely implemented on the whole state. Indeed, for instance, the projection of z on x is linear and is clearly insecure. Therefore, care must be taken when implementing (linear) operations between shares. For instance, it is secure to project z on the code of generating matrix H in parallel to the code of generating matrix G, and to get then y, but the projection algorithm shall be scrutinized. Indeed, the following method:

Step 1: z is projected on the code of generating matrix G in parallel to the code of generating matrix H to retrieve x,
Step 2: then y is retrieved as the subtraction $z \oplus xG$ on $\mathbb{F}_2$,

is not desirable from a security standpoint, owing to the demasking of the variable at Step 1.

6.4 Leakage squeezing

Background about Leakage Squeezing countermeasure The leakage squeezing idea [27,37] is based on masking but additionally applies some bijective functions (linear or non-linear) to the shares. A quantitative analysis of the gain in terms of *bounded moment leakage* security model is carried out in [36], where it is found that the best bijections can be non-linear in relation with non-linear codes (e.g., the Nordstrom-Robinson code for $k = 8$, $n = 16$). A comprehensive search of functions / codes suitable in the *bounded moment leakage* model is carried out in [19]. The suitable codes are nicknamed *Complementary Information Set* (CIS). A survey of usage of codes in the field of side-channel analysis is conducted by the first author [14]. In parallel, an approach using cellular automata to build codes is proposed in [33]. Following from [36], the conditions for building better codes are precised in [17]. Also, this journal paper shows that the leakage squeezing countermeasure resists model imperfections. The mutual information between the sensitive data and the leakage is computed empirically in [36,17]. In [18], it is demonstrated mathematically that this mutual information vanishes exponentially with the noise variance, at a rate which is proportional to the countermeasure first non-constant moment (known as the *HCI* or *High-order Correlation Immunity*). In this section, we relate bounded moments, mutual information, and attack success rate. That is, we show that the attacks are all the more difficult as the first non-constant moment of the leakage is high, and that this behaviour tracks that of the mutual information.

Eventually, notice that leakage squeezing with more than two shares has already been studied, from a security perspective in [15] and from the codes construction point of view in [24] (where HO-CIS codes are introduced as a generalization of CIS codes). The most recent survey on codes in side-channel analysis is available in [29]. In the rest of this section on leakage squeezing, we do detail only leakage squeezing with two shares.

Definition and use-case Leakage squeezing (LS) consists in masking $X \in \mathbb{F}_2^k$ using representation

$$(X \oplus Y, F(Y)), \tag{3}$$

where F is a bijective function from $\mathbb{F}_2^k$. The security order of LS is studied in [36]. We compare here-after LS at various orders (and we use indices, e.g., F_t, for $t = 0, 1, \ldots$, to make a difference between the different functions F):

- 0 (no protection); the leakage has only one share, that is X (plain).
- 1: $F_1 = Id$, i.e., Eqn. (3) represents perfect masking ($F_1(y) = y$).
- 2: F_2 is a linear function, where the matrix of F_2 is:

$$M_2 = \begin{pmatrix} 0\,0\,1\,1 \\ 1\,1\,0\,1 \\ 0\,1\,1\,1 \\ 1\,1\,0\,0 \end{pmatrix}.$$

– 3: F_3 is a linear function (which is optimal—cf. $F3'$ in [17, Sec. 5.2]), of matrix:

$$M_3 = \begin{pmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}.$$

Alternatively, the truth tables of F_t are (using hexadecimal notations):

– $\{F_1(y), 0 \leq y < 2^4\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f\}$,
– $\{F_2(y), 0 \leq y < 2^4\} = \{0, a, e, 4, 5, f, b, 1, 7, d, 9, 3, 2, 8, c, 6\}$,
– $\{F_3(y), 0 \leq y < 2^4\} = \{0, e, d, 3, b, 5, 6, 8, 7, 9, a, 4, c, 2, 1, f\}$.

When the bijective functions F_t are linear, the leakage squeezing is a special instance of ODSM, with generating matrices G and H defined in Eqn. (1) equal to $G = (I_k||0)$ and $H = (I_k||M_t)$.

Leakage distributions for Leakage Squeezing The resulting (uni-variate) distributions in Hamming weight, when $X \oplus Y$ and $F_t(Y)$ are manipulated in parallel, are represented in Fig. 4(a), when the noise has variance $\sigma^2 = 1$. The versions resisting attacks at orders 1, 2 and 3 are represented in Fig. 5(a), Fig. 6(a), and Fig. 7(a). The scale is the same for all plots. It can be seen that:

– distributions in Fig. 4(a) do not have the same mean,
– distributions in Fig. 5(a) have the same mean ($= n = 4$), but not the same variance (informally, some distributions are *larger* than others),
– distributions in Fig. 6(a) have the same mean ($= n = 4$), the same variance ($= n/2 + \sigma^2$), but not the same skewness (informally, some distributions are bending to the *right*, other to the *left*, while the others are *straight*),
– distributions in Fig. 7(a) have same mean ($= n = 4$), same variance ($= n/2 + \sigma^2$), no skewness, but different kurtosis (informally, some distributions have *smaller tails* than others).

Remark 3. The distributions represented in Fig. 4(a), Fig. 5(a), Fig. 6(a), and Fig. 7(a) are the convolution of the 2^n cosets of the weight distribution of the graph of functions F_t, for $0 \leq t \leq 3$.

The bi-variate distributions, that is:

$$(w_H(X \oplus Y) + N, w_H(F_t(Y)) + N') \in \mathbb{R}^2, \quad \text{where } N, N' \sim \mathcal{N}(0, \sigma^2),$$

which represent the word-oriented case, are represented in Fig. 4(b), Fig. 5(b), Fig. 6(b), and Fig. 7(b). It can be seen that Fig. 4(a) is merely the value at abscissa of the corresponding bi-variate distribution (somehow artificial, since this implementation uses only one share—however, the representation allows to contrast leakage of unprotected and protected implementations) represented in Fig. 4(b). Besides, Fig. 5(a), Fig. 6(a), and Fig. 7(a) are merely the diagonal

of corresponding bi-variate distributions represented in Fig. 5(b), Fig. 6(b), and Fig. 7(b).

It is interesting to see that some distributions are identical for some values of x. We group identical distributions by classes, labelled in lexicographical order. In the uni-variate case (recall Eqn. (5)), the number of classes is respectively 5, 5, 6 and 3 (for bijection F_0, F_1, F_2 and F_3), as represented in Tab. 1. The bi-variate case (recall Eqn. (4)) is represented in the bottom line of Tab. 1. In these tables, the layout is as given below:

$$\begin{bmatrix} \texttt{x=0x0} & \texttt{x=0x1} & \texttt{x=0x2} & \texttt{x=0x3} \\ \texttt{x=0x4} & \texttt{x=0x5} & \texttt{x=0x6} & \texttt{x=0x7} \\ \texttt{x=0x8} & \texttt{x=0x9} & \texttt{x=0xa} & \texttt{x=0xb} \\ \texttt{x=0xc} & \texttt{x=0xd} & \texttt{x=0xe} & \texttt{x=0xf} \end{bmatrix}.$$

Table 1. Classes of identical uni-variate and bi-variate distributions, in leakage squeezing with functions F_t, for $0 \leq t \leq 3$

	$t=0$	$t=1$	$t=2$	$t=3$
uni-variate distribution	0 1 1 2 0 1 1 2 0 1 2 2 0 1 1 2 1 2 2 3 1 2 2 3 1 3 4 4 1 2 2 1 1 2 2 3 1 2 2 3 2 4 5 1 1 2 2 1 2 3 3 4 2 3 3 4 2 4 1 5 2 1 1 2			
bi-variate distribution	0 1 1 2 0 1 1 2 0 1 2 3 0 1 1 2 1 2 2 3 1 2 2 3 1 4 5 6 1 2 2 3 1 2 2 3 1 2 2 3 2 5 7 8 1 2 2 3 2 3 3 4 2 3 3 4 3 6 8 9 2 3 3 4			

Uni- and Bi-variate Attacks on Leakage Squeezing Attacks have been simulated, both in uni- and bi-variate settings. In the *bi-variate setting*, the attacker gets the leakages $L^{(1)}$ and $L^{(2)}$ corresponding to masked data and mask (with bijection $F_t : \mathbb{F}_2^k \to \mathbb{F}_2^k$ applied on it):

$$\begin{cases} L^{(1)} = w_H(T \oplus k^* \oplus Y) + N \\ L^{(2)} = w_H(F_t(Y)) + N' \end{cases}, \tag{4}$$

where $X = T \oplus k^*$ is the sensitive variable (known text $T \in \mathbb{F}_2^k$ and secret key $k^* \in \mathbb{F}_2^k$). The equation (4) is the application of the Hamming weight leakage model on the two shares of Eqn. (3), and in the addition of noise. In the *uni-variate setting*, the attackers gets only one leakage sample:

$$L = L^{(1)} + L^{(2)}. \tag{5}$$

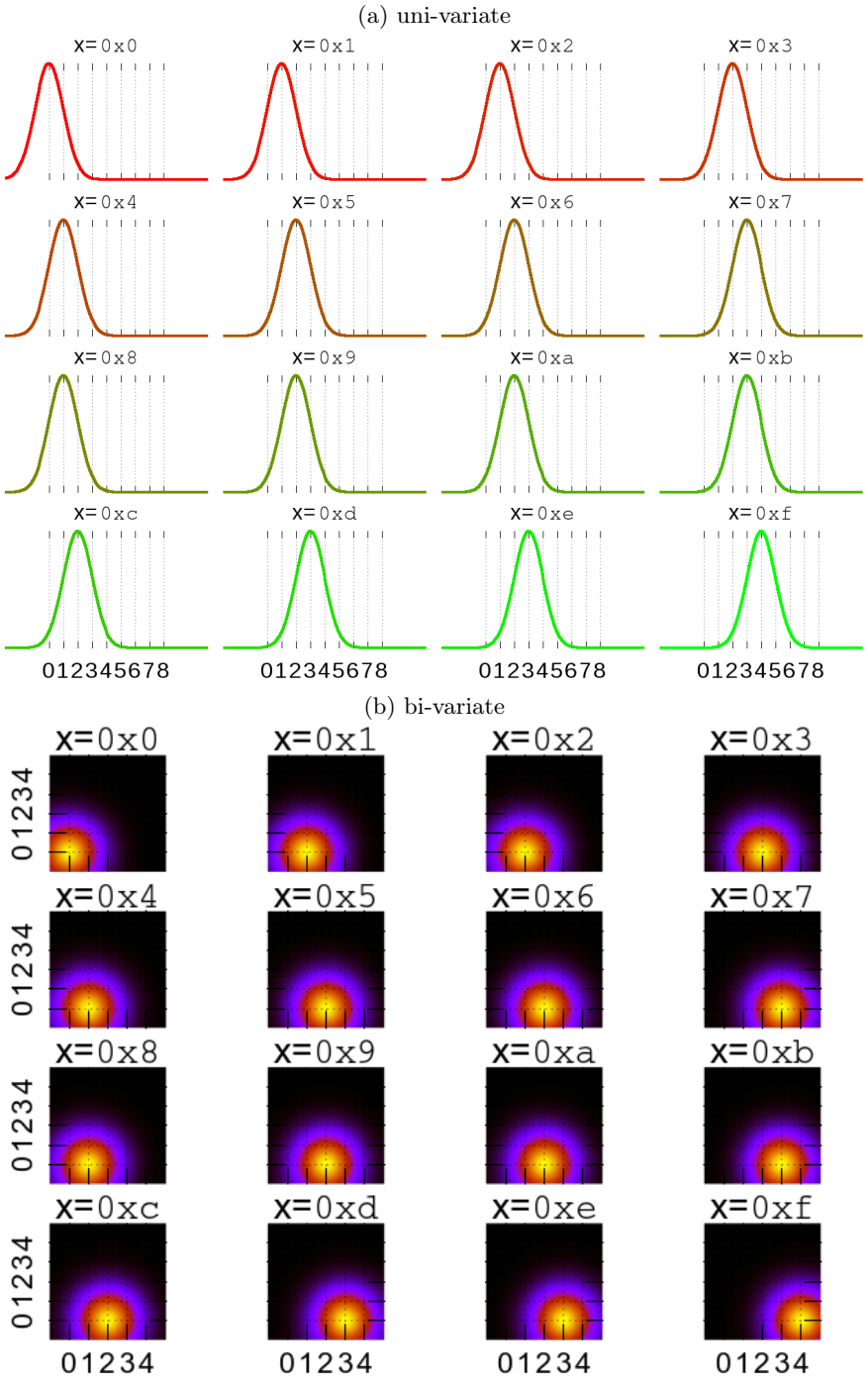

Fig. 4. Leakage distribution without countermeasure ($n = 4$). Due to absence of masking, the leakage traces consist in Gaussian functions, centered at $0, 1, \ldots, k$

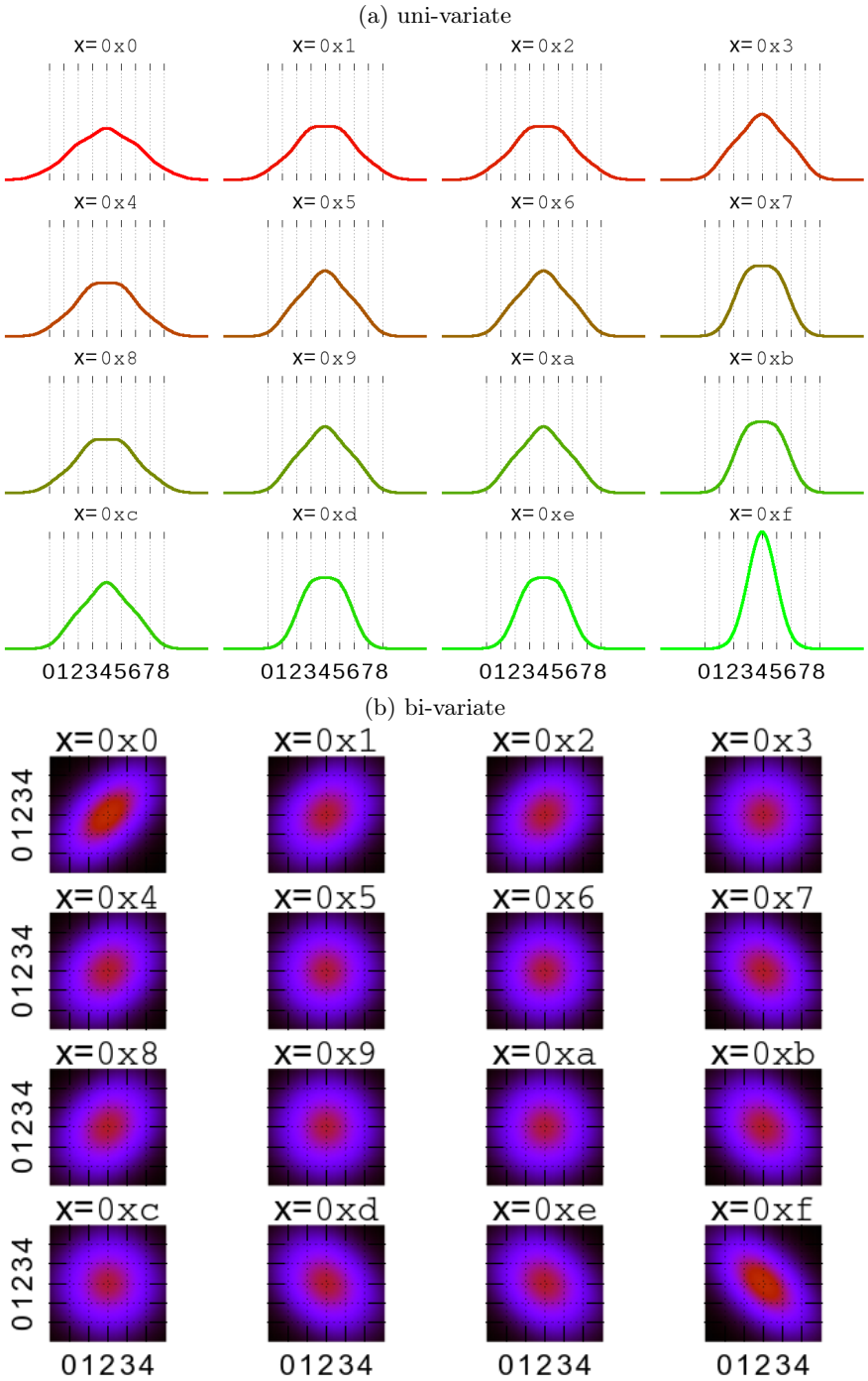

Fig. 5. Leakage distribution with leakage squeezing countermeasure at order 1 ($k = 4$, see Sec. 6.4)

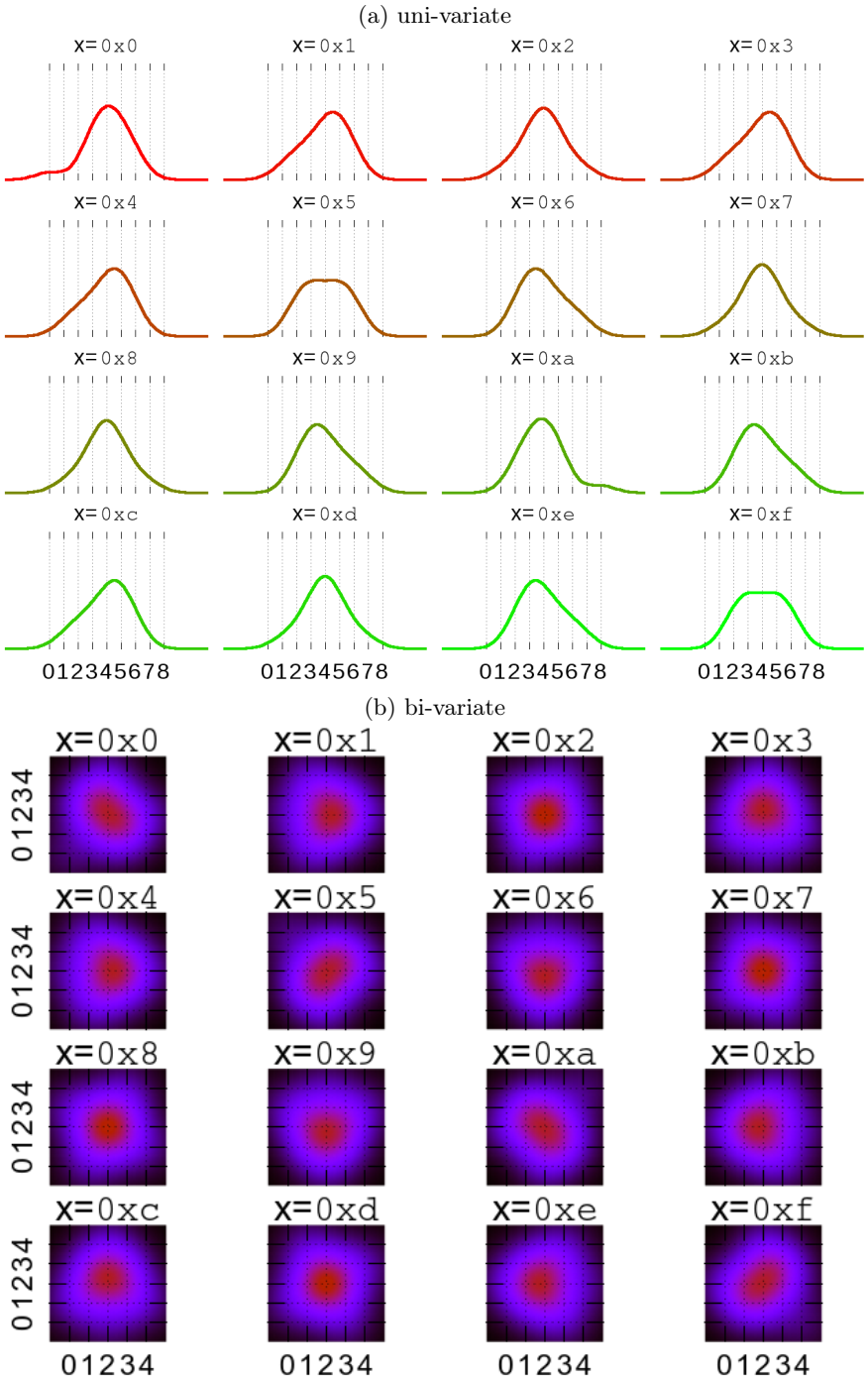

Fig. 6. Leakage distribution with leakage squeezing countermeasure at order 2 ($k = 4$, see Sec. 6.4)

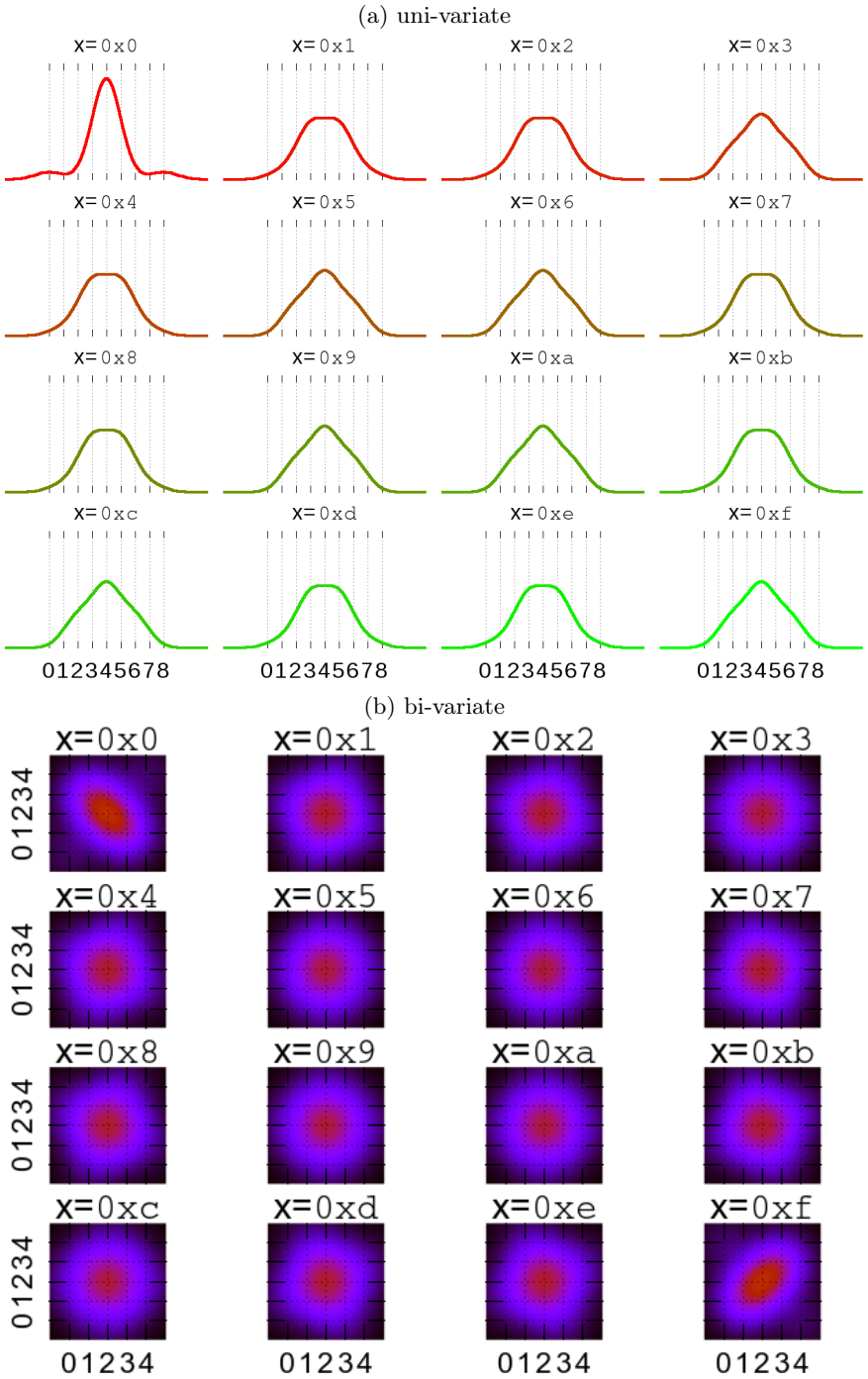

Fig. 7. Leakage distribution with leakage squeezing countermeasure at order 3 ($k = 4$, see Sec. 6.4)

For the sake of fair comparison, we focus on the *optimal attack* [12], that is, the attack which maximizes the probability of success in secret key recovery. Notice that the bijections used in leakage squeezing countermeasure are supposed public information.

– The uni-variate attack measures the sum $l_q^{(1)} + l_q^{(2)}$ of leakage for each trace q $(1 \leq q \leq Q)$, hence the optimal attack estimates the correct key k^* as:

$$\hat{k^*} = \underset{k \in \mathbb{F}_2^k}{\operatorname{argmax}} \sum_{q=1}^{Q} \log \sum_{y \in \mathbb{F}_2^k} \setminus \exp -\frac{1}{4\sigma^2} \left\{ \left(l_q^{(1)} + l_q^{(2)} - w_H(t_q \oplus k \oplus y, F_t(y)) \right)^2 \right\}. \quad (6)$$

– The bi-variate attacks measures each share $l_q^{(1)}$ and $l_q^{(2)}$ independently, hence the optimal attack estimates the correct key k^* as:

$$\hat{k^*} = \underset{k \in \mathbb{F}_2^k}{\operatorname{argmax}} \sum_{q=1}^{Q} \log \sum_{y \in \mathbb{F}_2^k} \setminus \exp -\frac{1}{2\sigma^2} \left\{ \left(l_q^{(1)} - w_H(t_q \oplus k \oplus y) \right)^2 + \left(l_q^{(2)} - w_H(F_t(y)) \right)^2 \right\}. \quad (7)$$

Notice that the noise in uni-variate case is $N + N' \sim \mathcal{N}\left(0, 2\sigma^2\right)$, whereas in the bi-variate case, it is $(N, N') \sim \mathcal{N}\left(\left(\begin{smallmatrix} 0 & 0 \end{smallmatrix}\right), \sigma^2 \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)\right)$; this explains the different factors in the exponential for expressions (6) and (7). Results in terms of success rate ($\mathsf{SR} = \mathbb{P}(\hat{h^*} = k^*)$) are shown in Fig. 8 for $\sigma = 1$. The success rates are obtained after 100 independent attacks, and the estimation error of the curves are superimposed (they correspond to $\pm$ the standard deviation of the SR estimator; refer to [38] for their calculation).

It can be seen that the security increases (i.e., more and more traces are needed to recover the key) with the resistance order t, $0 \leq t \leq 3$. Said differently, the larger the dual distance of the code generated by H, the more difficult the attack. Moreover, it appears clearly that bi-variate attacks are more successful than uni-variate attacks, since information is lost while the two leakages are summed up (recall that in Tab. 1, there are less classes in the uni-variate case than in the bi-variate case). This notice settles a quantitative assessment why so-called zero-offset *uni-variate* attacks [56] are less efficient than truly *multi-variate* counterparts. The two functions F_2 and F_3 seem to yield similar security level, at least for low noise $\sigma = 1$. However, when the noise increases, F_3 clearly increases more than F_2 the resistance of the implementation against attacks, as illustrated in Fig. 9 for $\sigma = 2$. One can see the "*staggering*" of the number of traces to succeed for a given order: the success rate curve without protection (F_0) is squared to obtain that with 1st-order protection (F_1). This fact has already been reported in [28].

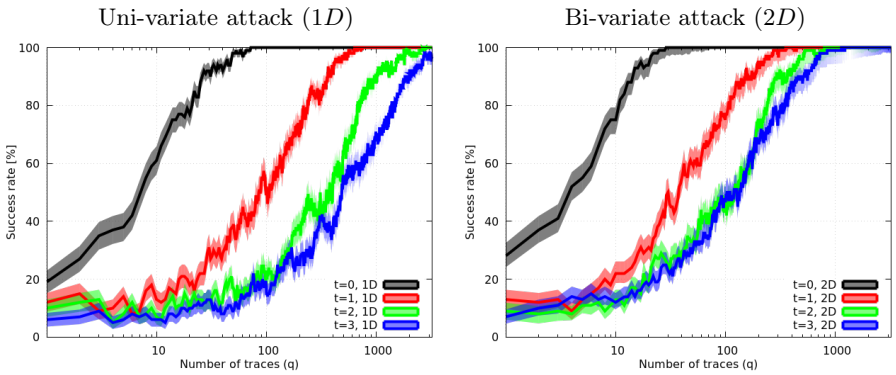

Fig. 8. Attack result for $\sigma = 1$

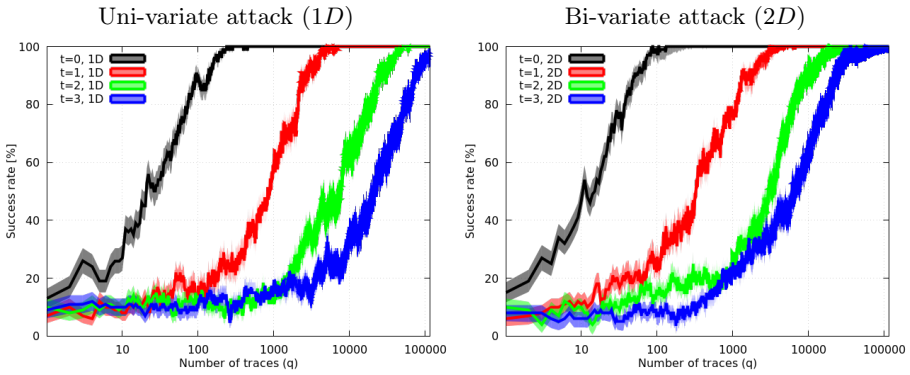

Fig. 9. Attack result for $\sigma = 2$

Information Leakage under Leakage Squeezing Protection Besides, we also evaluate the information leakage of the four levels of protections. We compute $I(L;X)$, where $X = T \oplus k^*$ is uniformly distributed over $\mathbb{F}_2^k$, and where the leakage L is the uni- or bi-variate leakage function.

- In the uni-variate case, L is

$$L^{(1)} + L^{(2)} = w_H(T \oplus k^* \oplus Y) + N + w_H(F_t(Y)) + N' \in \mathbb{R};$$

- In the bi-variate case, L is

$$(L^{(1)}, L^{(2)}) = (w_H(T \oplus k^* \oplus Y) + N, w_H(F_t(Y)) + N') \in \mathbb{R}^2,$$

where:

- Y is uniformly distributed over $\mathbb{F}_2^{n-k}$ (here $= \mathbb{F}_2^k$ since $n = 2k$) and
- N and N' are two additive (recall Fig. 1) and independent noises of centered normal law with identical standard deviation σ.

The resulting mutual information values are given in Fig. 10 for uni- and bi-variate attacks.

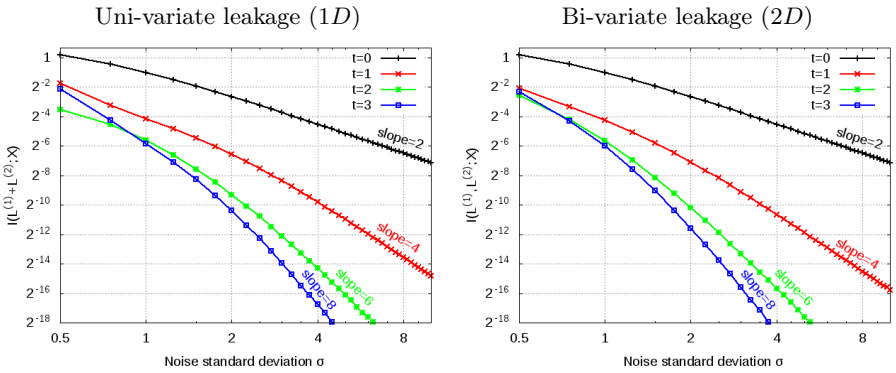

Fig. 10. Mutual information analysis for uni- and bi-variate leakage

Interestingly, in presence of large noise, the mutual information decreases affinely with σ (in log-log scale), with a slope $-2(t+1) = -2d$, where:

- t is the protection order, and
- $d = t+1$ is the minimum order of a successful attack (also denoted *High-order Correlation Immunity* or *HCI* in [18, Def. 2]).

This noting is demonstrated mathematically in [18, Theorem 1].

It can thus be stated that LS with bijection F_t has the same bit-level security with two shares as perfect masking with $t + 1 = d$ shares.

Link between Attacks and Information Leakage It is demonstrated in [28] that, for *additive distinguishers*, there exists a coefficient E, called *first-order exponent*, such that the number of traces q to extract the key k^* with success probability SR satisfies the property:

$$1 - \mathsf{SR} \approx \exp -q \cdot E, \tag{8}$$

where $\approx$ is an asymptotic equivalence (detailed in [28]).

It is hinted in [52] that such exponent is proportional to the mutual information (as computed in previous section 6.4), provided the distinguisher is the *template attack*. Now, with perfect profiling, the *template attack* [23] coincides with the *optimal distinguisher* [12]. Thus, we aim in this section at validating this finding on the bi-variate (but higher-order secure) LS masking scheme, using the distinguishers (6) and (7) for the optimal attacks.

To validate experimentally that the first-order E involved in (8) is proportional to $I(L;X)$, we extract the number of measurements q to recover the key k^* with probability $\mathsf{SR} = 80\%$. The two figures 8 and 9 allow to extract 16 values of number of traces. The corresponding values (for $\sigma = 1$ and 2) of $I(L;X)$ are extracted from Fig. 10. These data are represented in Fig. 11.

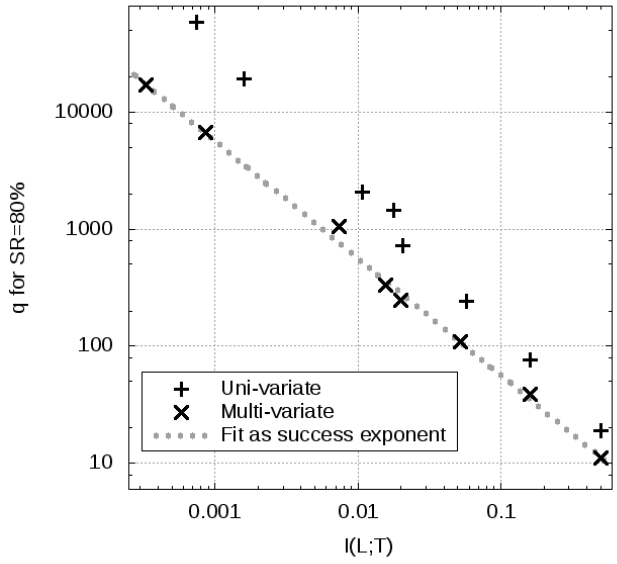

Fig. 11. Number of traces to extract the secret key k^* with probability 80% as a function of the mutual information between the leakage and the sensitive variable $X = T \oplus k^*$

In the case of the bi-variate attack, it is possible to fit these data by linear regression as relationship:

$$\log(q) = \log(-\log(0.80)) - \log(\alpha \cdot I(L;X)),$$

where the estimated parameter α is found to be $\alpha = 0.0396361 \pm 0.0002805$. This good fit with a law where $q \times I(L;X)$ is a constant (curve of slope -1 in Fig. 11) validates that in the case of optimal attack on bi-variate leakage, one has that (8) holds, with first-order exponent equal to:

$$E = \alpha \cdot I(L;X). \tag{9}$$

We underline that this result holds, surprisingly, for 4 different leakage scenarios (corresponding to the use of F_t, $t \in \{0,1,2,3\}$). Therefore, the relationship (9) seems very general. On the contrary, it might explain why the law (8) fits less nicely (the interpolated slope of the curve is > -1), since sum of two leakages is a *ad hoc* operation.

7 A new construction for leakage squeezing and inner product masking

7.1 Rationale of the construction

In this section, we explain how to obtain CIS (and HO-CIS) codes based on code expansion from $\mathbb{F}_{2^k}$ to $\mathbb{F}_2$. The procedure is the following:

1. Decide on a number m of shares of k bit words.
2. Search for a code of parameters $[m,1]_{2^k}$ of minimum distance m; basically, this means that the generating matrix of the code consists in a line of m non-zero values of $\mathbb{F}_{2^k}$.
3. Expand the code on $\mathbb{F}_2$. This code is HO-CIS of order m (see Proposition 2.2 of [1]). The protection order of this code in bit-level security models is equal to its minimum distance minus one.
4. Write its generating matrix as $(M_1||M_2||\ldots||M_m)$, where M_i $(1 \leq i \leq m)$ are $k \times k$ matrices with entries in $\mathbb{F}_2$.
5. As explained in [16, Appendix B, page 21], the linear function to apply to share i $(1 \leq i \leq m)$ is generated by matrix M_i^{-1}.

7.2 Example on a non-optimal code

We detail in Listing 1.1 an example of a masking with $m = 2$ shares of $k = 4$ bits, which has order 1 security at word level and order 2 security at the bit level[5].

[5] Notice that in Listing 1.1 and in the rest of this section, the symbol X denotes the dummy variable for field $\mathbb{F}_2$ extension to $\mathbb{F}_{16}$. Thus, it shall not be confused with X, the sensitive variable (recall Eqn. (1)).

```
F<X> := PolynomialRing(GF(2));
P := F ! 1+X+X^4; // Degree k irreducible polynomial
GF16<X> := ext<GF(2)|P>;
C5:=LinearCode<GF16, 2 | [1,1+X]>;                                    //
    Step 2
//  [2, 1, 2] Constacyclic by X^7 Linear Code over GF(2^4)
//  Generator matrix:
//  [   1  X^4]
C5_expanded := SubfieldRepresentationCode(C5,GF(2));                  //
    Step 3
//  [8, 4, 3] Linear Code over GF(2)
//  Generator matrix:
//  [1 0 0 0 1 1 0 0]
//  [0 1 0 0 0 1 1 0]
//  [0 0 1 0 0 0 1 1]
//  [0 0 0 1 1 1 0 1]
M5_inv:=Submatrix(GeneratorMatrix(C5_expanded),1,5,4,4);              //
    Step 4
M5_inv^-1; // Used as bijection F5                                    //
    Step 5
//  [0 1 1 1]
//  [1 1 1 1]
//  [1 0 1 1]
//  [1 0 0 1]
```

Listing 1.1. Example of program for obtaining codes (in magma [55] language)

The construction for this code needed in leakage squeezing is explained below:

1. we opt for a leakage squeezing with a mask Y of bitwidth equal to that of the data X to protect,
2. the code `C5` in $\mathbb{F}_{2^k}$ is generated by $(1||1+X)$, where $\mathbb{F}_{2^4} = \mathbb{F}_2[X]/1+X+X^4$, hence has parameters $[2,1,2]_{16}$,
3. this code is expanded into `C5_expanded`, which has parameters $[8,4,3]_2$. Therefore, the security at bit level is $3-1=2$, which is one more than that of `C5` at word level,
4. the generating matrix of `C5_expanded` is written in systematic form as $(I_4||\texttt{M5\_inv}) = \begin{pmatrix} 1\,0\,0\,0 & 1\,1\,0\,0 \\ 0\,1\,0\,0 & 0\,1\,1\,0 \\ 0\,0\,1\,0 & 0\,0\,1\,1 \\ 0\,0\,0\,1 & 1\,1\,0\,1 \end{pmatrix}$
5. the researched linear bijection has matrix $M5 = \texttt{M5\_inv}^{-1} = \begin{pmatrix} 0\,1\,1\,1 \\ 1\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,0\,0\,1 \end{pmatrix}$.

The resulting linear function has truth table:

$$\{F_5(y), 0 \leq y < 2^4\} = \{\texttt{0}, \texttt{e}, \texttt{3}, \texttt{d}, \texttt{7}, \texttt{9}, \texttt{4}, \texttt{a}, \texttt{f}, \texttt{1}, \texttt{c}, \texttt{2}, \texttt{8}, \texttt{6}, \texttt{b}, \texttt{5}\}.$$

7.3 Example on an optimal code

In the case $k = 4$ and $n = 2k = 8$, we detail how the (autodual, and unique of type-II [6]) Reed-Muller RM$(1, 3)$ code with parameters $[8, 4, 4]_2$ and generating matrix

$$\begin{pmatrix} 1\,0\,0\,0 & 0\,1\,1\,1 \\ 0\,1\,0\,0 & 1\,0\,1\,1 \\ 0\,0\,1\,0 & 1\,1\,0\,1 \\ 0\,0\,0\,1 & 1\,1\,1\,0 \end{pmatrix} \tag{10}$$

can be derived from a linear code of parameters $[2, 1]_{16}$ over $\mathbb{F}_{2^4}$. We aim to find an irreducible polynomial $P(X)$ such that $\mathbb{F}_{16} = \mathbb{F}_2[X]/P(X)$ and the code over $\mathbb{F}_{16}$ has parameters $[1, X + X^2 + X^3]_2$. Equivalently, this means that:

1. $d^\circ(P(X)) = 4$ and
2. the three following conditions are met:
   - $X(X + X^2 + X^3) = 1 + X^2 + X^3 \bmod P(X)$,
   - $X^2(X + X^2 + X^3) = 1 + X + X^3 \bmod P(X)$,
   - $X^3(X + X^2 + X^3) = 1 + X + X^2 \bmod P(X)$.

The second, third, and forth condition mean that $P(X)$ is a divisor of $\gcd(X(X + X^2 + X^3) + 1 + X^2 + X^3, X^2(X + X^2 + X^3) + 1 + X + X^3, X^3(X + X^2 + X^3) + 1 + X + X^2) = \gcd(1 + X^4, 1 + X + X^4 + X^5, 1 + X + X^2 + X^4 + X^5 + X^6)$.

Now, it happens indeed that $1+X^4|1+X+X^4+X^5, 1+X+X^2+X^4+X^5+X^6$. However, $\mathbb{F}_2[X]/(1 + X^4)$ is a ring, and not a field, because $1 + X^4 = (1 + X)^4$. Thus, the code is defined over a ring, which has never been analyzed this way in masking, and which opens the door to interesting perspectives.

We have tested all 4! permutations of the four last columns in the $[8, 4, 4]_2$ code generating matrix (10), without success to lift this code as a $[2, 1, 2]_{16}$ code in $\mathbb{F}_{16}$. Actually, this code can be obtained as a binary image (through a graymap) of a $[2, 1]$ code on $\mathbb{F}_4[X]/(X^2)$ or on $\mathbb{F}_2[X]/(X^4)$, and of a code of parameters $[4, 2]$ on $\mathbb{F}_2[X]/(X^2)$.

8 Conclusions

In this paper we have studied the statistical distribution of uni- and multi-variate leakage functions of cryptographic implementations when some countermeasures against fault injection and side-channel analyses are applied.

We have observed that the previous studied protection called *leakage squeezing* is a generalization of the variants of perfect masking, including inner product masking (see Tab. 2 for a recap). In this sense, we extend the work [47], which explores the links between inner product masking and direct sum masking. We show that leakage squeezing is all the more secure as its underlying code has a high minimum distance d. Side-channel attacks of orders $1, \ldots, t = d - 1$ are impossible. We relate this value to the slope $-2d$ of the mutual information between sensitive variables and the leakage (represented in log-log scale), and show

[6] See http://www.unilim.fr/pages_perso/philippe.gaborit/SD/GF2/GF2II.htm.

that, in practice, the success rate of attacks is less when d is large. We also reveal that bi-variate mutual information (resp. bi-variate attack probability success) is less than in the uni-variate case.

Table 2. Hierarchy between masking styles

Perfect masking [9]	⊆	**Inner product** [3]	⊆	**Leakage squeezing** [17]	⊆	**ODSM** [10]
$k\|n$, G and H are made up of block matrices of size $k \times k$ being either zero or the identity (cf. Eqn. (2)).		$k\|n$, G and H are made up of $k \times k$ invertible matrices corresponding to $\mathbb{F}_{2^k}$-linear isomorphisms[7].		$k\|n$, G and H are arbitrary matrices — LS can even extend to non-linear codes.		k and $n \geq k$ are free, and G and H must simply generate two complementary codes in $\mathbb{F}_2^n$.

Eventually, we propose a new method to build (HO-)CIS codes based on code expansion, which is promising in the context of leakage squeezing, i.e., when a high level of security is required both at word- and at bit-level.

Acknowledgements

The authors wish to thank Patrick Solé for valuable inputs and suggestions about this article. This work was supported in part by National Natural Science Foundation of China (No. 61632020), and by the ANR CHIST-ERA project SECODE (*Secure Codes to thwart Cyber-physical Attacks*). The authors are also grateful to Félix Ulmer from University of Rennes 1 for inputs about lifting of binary codes on larger structures of size 2^m, where $m > 1$.

References

1. Adel Alahmadi, Cem Güneri, Hatoon Shohaib, and Patrick Solé. Long quasi-polycyclic t-CIS codes. *CoRR*, abs/1703.03109, 2017.
2. Sabine Azzi, Bruno Barras, Maria Christofi, and David Vigilant. Using linear codes as a fault countermeasure for nonlinear operations: application to AES and formal verification. *J. Cryptographic Engineering*, 7(1):75–85, 2017.
3. Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. Inner product masking revisited. In Oswald and Fischlin [45], pages 486–510.
4. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, and Benjamin Grégoire. Compositional Verification of Higher-Order Masking: Application to a Verifying Masking Compiler. *IACR Cryptology ePrint Archive*, 2015:506, 2015.

[7] Also, paper [57] presents a variant of inner product masking [3] in that the $k \times k$ submatrices can take on any value.

5. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. Verified Proofs of Higher-Order Masking. In Oswald and Fischlin [45], pages 457–485.
6. 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 Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.
7. Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptology*, 24(2):269–291, 2011.
8. Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. A low-entropy first-degree secure provable masking scheme for resource-constrained devices. In *Proceedings of the Workshop on Embedded Systems Security, WESS 2013, Montreal, Quebec, Canada, September 29 - October 4, 2013*, pages 7:1–7:10. ACM, 2013.
9. Johannes Blömer, Jorge Guajardo, and Volker Krummel. Provably Secure Masking of AES. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2004.
10. 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 David Naccache and Damien Sauveron, editors, *Information Security Theory and Practice. Securing the Internet of Things - 8th IFIP WG 11.2 International Workshop, WISTP 2014, Heraklion, Crete, Greece, June 30 - July 2, 2014. Proceedings*, volume 8501 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2014.
11. 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. Cryptology ePrint Archive, Report 2014/665, 2014. http://eprint.iacr.org/2014/665/ (extended version of conference paper [**?**]).
12. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks Will Fall Off – Higher-Order Optimal Distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 344–365. Springer, 2014.
13. Nicolas Bruneau, Sylvain Guilley, Zakaria Najm, and Yannick Teglia. "multivariate high-order attacks of shuffled tables recomputation". *Journal of Cryptology*, pages 1–43, 2017.
14. Claude Carlet. Correlation-Immune Boolean Functions for Leakage Squeezing and Rotating S-Box Masking against Side Channel Attacks. In Benedikt Gierlichs, Sylvain Guilley, and Debdeep Mukhopadhyay, editors, *SPACE*, volume 8204 of *Lecture Notes in Computer Science*, pages 70–74. Springer, 2013.
15. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. In Steven D. Galbraith and Mridul Nandi, editors,

Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings, volume 7668 of *Lecture Notes in Computer Science*, pages 120–139. Springer, 2012.

16. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. Cryptology ePrint Archive, Report 2012/567, 2012. http://eprint.iacr.org/2012/567.
17. 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.
18. 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.
19. Claude Carlet, Philippe Gaborit, Jon-Lark Kim, and Patrick Solé. A New Class of Codes for Boolean Masking of Cryptographic Computations. *IEEE Transactions on Information Theory*, 58(9):6000–6011, 2012.
20. Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-Order Masking Schemes for S-Boxes. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 366–384. Springer, 2012.
21. Claude Carlet and Sylvain Guilley. Complementary dual codes for countermeasures to side-channel attacks. *Adv. in Math. of Comm.*, 10(1):131–150, February 2016.
22. Claude Carlet, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Algebraic Decomposition for Probing Security. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 742–763. Springer, 2015.
23. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
24. Yeow Meng Chee, Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, Han Mao Kiah, Jon-Lark Kim, Patrick Solé, and Xiande Zhang. Multiply Constant-Weight Codes and the Reliability of Loop Physically Unclonable Functions. *IEEE Transactions on Information Theory*, 60(11):7026–7034, 2014.
25. Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Improved Collision-Correlation Power Analysis on First Order Protected AES. In Preneel and Takagi [48], pages 49–62.
26. Jean-Sébastien Coron. Higher Order Masking of Look-Up Tables. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 441–458. Springer, 2014.
27. Jean-Luc Danger and Sylvain Guilley. Cryptography circuit protected against observation attacks, in particular of a high order, January 18 2010. International patent, granted as CA2749961, CN102405615, ES2435721, EP2380306, FR2941342, JP2012516068, KR20120026022, SG173111, US2012250854 and WO2010084106.
28. Sylvain Guilley, Annelie Heuser, and Olivier Rioul. A Key to Success - Success Exponents for Side-Channel Distinguishers. In Alex Biryukov and Vipul Goyal,

editors, *Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, volume 9462 of *Lecture Notes in Computer Science*, pages 270–290. Springer, 2015.
29. Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Codes for Side-Channel Attacks and Protections. In Said El Hajji, Abderrahmane Nitaj, and El Mamoun Souidi, editors, *Codes, Cryptology and Information Security - Second International Conference, C2SI 2017, Rabat, Morocco, April 10-12, 2017, Proceedings - In Honor of Claude Carlet*, volume 10194 of *Lecture Notes in Computer Science*, pages 35–55. Springer, 2017.
30. Annelie Heuser. *Distinguishing Distinguisher : A Theoretical Approach to Side-channel Analysis*. PhD thesis, TELECOM-ParisTech, December 18 2015.
31. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, August 17–21 2003. Santa Barbara, California, USA.
32. Marc Joye and Michael Tunstall. *Fault Analysis in Cryptography*. Springer LNCS, March 2011. DOI: 10.1007/978-3-642-29656-7 ; ISBN 978-3-642-29655-0.
33. Sandip Karmakar and Dipanwita Roy Chowdhury. Leakage Squeezing Using Cellular Automata. In Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Automata*, volume 8155 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2013.
34. Mark G. Karpovsky, Konrad J. Kulikowski, and Alexander Taubin. Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard. In Jean-Jacques Quisquater, Pierre Paradinas, Yves Deswarte, and Anas Abou El Kalam, editors, *Smart Card Research and Advanced Applications VI, IFIP 18th World Computer Congress, TC8/WG8.8 & TC11/WG11.2 Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS), 22-27 August 2004, Toulouse, France*, volume 153 of *IFIP*, pages 177–192. Kluwer/Springer, 2004.
35. HeeSeok Kim, Seokhie Hong, and Jongin Lim. A Fast and Provably Secure Higher-Order Masking of AES S-Box. In Preneel and Takagi [48], pages 95–107.
36. Houssem Maghrebi, Claude Carlet, Sylvain Guilley, and Jean-Luc Danger. Optimal First-Order Masking with Linear and Non-linear Bijections. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT*, volume 7374 of *Lecture Notes in Computer Science*, pages 360–377. Springer, 2012.
37. Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage squeezing countermeasure against high-order attacks. In Claudio Agostino Ardagna and Jianying Zhou, editors, *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - 5th IFIP WG 11.2 International Workshop, WISTP 2011, Heraklion, Crete, Greece, June 1-3, 2011. Proceedings*, volume 6633 of *Lecture Notes in Computer Science*, pages 208–223. Springer, 2011.
38. Houssem Maghrebi, Olivier Rioul, Sylvain Guilley, and Jean-Luc Danger. Comparison between Side-Channel Analysis Distinguishers. In Tat Wing Chim and Tsz Hon Yuen, editors, *ICICS*, volume 7618 of *LNCS*, pages 331–340. Springer, 2012.
39. Thomas S. Messerges. Securing the AES finalists against power analysis attacks. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 2000.

40. Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 238–251. Springer-Verlag, August 17-18 2000. Worcester, MA, USA.
41. Amir Moradi. Statistical tools flavor side-channel collision attacks. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 428–445. Springer, 2012.
42. Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. *IACR Cryptology ePrint Archive*, 2014:409, June 2 2014.
43. Maxime Nassar, Sylvain Guilley, and Jean-Luc Danger. Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks. In Daniel J. Bernstein and Sanjit Chatterjee, editors, *Progress in Cryptology - INDOCRYPT 2011 - 12th International Conference on Cryptology in India, Chennai, India, December 11-14, 2011. Proceedings*, volume 7107 of *Lecture Notes in Computer Science*, pages 22–39. Springer, 2011.
44. NIST/ITL/CSD. Advanced Encryption Standard (AES). FIPS PUB 197, Nov 2001. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf (also ISO/IEC 18033-3:2010).
45. Elisabeth Oswald and Marc Fischlin, editors. *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*. Springer, 2015.
46. Jing Pan, Jerry I. den Hartog, and Jiqiang Lu. You cannot hide behind the mask: Power analysis on a provably secure S-box implementation. In Heung Youl Youm and Moti Yung, editors, *Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, volume 5932 of *Lecture Notes in Computer Science*, pages 178–192. Springer, 2009.
47. Romain Poussier, Qian Guo, François-Xavier Standaert, Claude Carlet, and Sylvain Guilley. Connecting and Improving Direct Sum Masking and Inner Product Masking. In Yannick Teglia and Thomas Eisenbarth, editors, *Smart Card Research and Advanced Applications - 16th International Conference, CARDIS 2017, Lugano, Switzerland, November 13-15, 2017, Revised Selected Papers*, Lecture Notes in Computer Science. Springer, 2017.
48. Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 – October 1, 2011. Proceedings*, volume 6917 of *LNCS*. Springer, 2011.
49. Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.
50. Tobias Schneider, Amir Moradi, and Tim Güneysu. Parti - towards combined hardware countermeasures against side-channel and fault-injection attacks. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 302–332. Springer, 2016.
51. Kai Schramm and Christof Paar. Higher Order Masking of the AES. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.
52. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In Masayuki Abe, editor, *Advances*

in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

53. Yannick Teglia, Pierre-Yvan Liardet, and Alain Pomet. Protection of the execution of a DES algorithm, March 27 2012. US Patent 8,144,865.
54. Michael Tunstall, Carolyn Whitnall, and Elisabeth Oswald. Masking Tables – An Underestimated Security Risk. In Shiho Moriai, editor, *FSE*, volume 8424 of *Lecture Notes in Computer Science*, pages 425–444. Springer, 2013.
55. University of Sydney. Magma Computational Algebra System. http://magma.maths.usyd.edu.au/magma/, Accessed on 2014-08-22.
56. Jason Waddle and David A. Wagner. Towards Efficient Second-Order Power Analysis. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2004.
57. 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 Kerstin Lemke-Rust and Michael Tunstall, editors, *Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers*, volume 10146 of *Lecture Notes in Computer Science*, pages 174–191. Springer, 2016.