Side-Channel Indistinguishability

Claude Carlet, Sylvain Guilley

Side-Channel Indistinguishability*

Claude Carlet
University of Paris 8, LAGA, (CNRS, UMR 7539), University of Paris 13,
Sorbonne Paris Cité, 2 rue de la liberté, F-93 526 Saint-Denis,
Cedex, France.
claude.carlet@gmail.com

Sylvain Guilley
Télécom-ParisTech (CNRS, UMR 5141),
37/39 rue Dareau, 75 014 Paris, France.
Secure-IC S.A.S., 80 avenue des Buttes de Coësmes, 35 700, Rennes, France.
sylvain.guilley@TELECOM-ParisTech.fr

Abstract

We introduce a masking strategy for hardware that prevents any side-channel attacker from recovering *uniquely* the secret key of a cryptographic device. In this masking scheme, termed *homomorphic*, the sensitive data is exclusive-ored with a random value that belongs to a given set. We show that if this masking set is concealed, then no information about the cryptographic key leaks. If the masking set is public (or disclosed), then any (high-order) attack reveals a group of *equiprobable* keys. Those results are applied to the case of the AES, where sensitive variables are bytes. To any mask corresponds a masked substitution box. We prove that there exists a homomorphic masking with 16 masks (hence a number of substitution boxes equal to that of the same algorithm without masking) that resists mono-variate first-, second-, and third-order side-channel attacks. Furthermore, even if the masking set is public, each byte of the correct key is found only *ex æquo* with 15 incorrect ones, making the side-channel analysis insufficient alone – the remaining key space shall be explored by other means (typically exhaustive search). Thus, our homomorphic masking strategy allows both to increase the number of side-channel measurements and to demand for a final non negligible brute-forcing (of complexity $16^{N_B} = 2^{64}$ for AES, that has $N_B = 16$ substitution boxes). The hardware implementation of the *Rotating Substitution boxes Masking* (RSM) is a practical instantiation of our homomorphic masking countermeasure.

Keywords: Brute-force, hardware implementation, high-order correlation power attacks (HO-CPA), homomorphic masking, indistinguishability, resilience, rotating substitution boxes masking (RSM), side-channel analysis, symmetric ciphers.

1 Introduction

Embedded systems that contain cryptography must be protected against side-channel analyses, to prevent attackers from exploiting information that leaks out of the system. Typical side-channels are the (instant) power consumption and the radiated electromagnetic field. The side-channel conveys a noisy image of internal variables right to the attacker. Important variables are those termed *sensitive*, that depend both on a varying input known by the attacker and on a constant inner data. In the context of block ciphers, they can typically be a part (*e.g.* one byte) of the plaintext (or of the ciphertext) and of the key (or of the key schedule). In addition, a sensitive variable shall depend on small parts of cryptographic key, *e.g.* bytes of it. This allows for a *divide-and-conquer* approach, where the key is recovered byte by byte (as in the case of

*Extended version of [3].

AES [14]). To prevent those attacks, countermeasures are applied. *Masking* (refer to [9, Chap. 9]) is a large class of them; it consists in randomizing the computations so that the leakage becomes less dependent on the sensitive variables. Masking in general applies as well to *software* and *hardware* designs.

This article brings two major contributions to the topic of masking, with application to hardware implementations. We introduce a masking strategy that is *leakage resilient*, while at the same time being applicable to any block cipher. The first contribution is the proof of its absence of leakage if the used masks are kept secret (*e.g.* alongside with the cryptographic key). The second contribution is a tradeoff between the size w of the masks set (assumed to be public this time) and the complexity of a brute-force search to finish the attack (since the correct key is recovered *ex æquo* with other incorrect keys).

The rest of the paper is organized as follows. Sec. 2 describes the studied homomorphic masking scheme. Its vulnerability evaluation is covered in Sec. 3. In Sec. 4, the secrecy of the homomorphic masking scheme is studied. Indistinguishability is demonstrated when the masks set is unknown. Otherwise, we prove that the hypotheses on the *key bytes* converge into an affine space of size less than or equal to $1 \leqslant w \leqslant 2^n$ (where n is the bitwidth of the sensitive variable, *e.g.* $n = 8$ for AES). Despite this burden, high-order attacks remain possible. They are discussed in Sec. 5, with the help of simulations. Several protections using only as many masks as identical substitution boxes in the algorithm (*e.g.* 16 in AES and in PRESENT) are proposed. In the case of AES, it is possible to find a mask set of cardinality $w = 16$ that thwarts zero-offset attacks of order 1 to $d = 3$ including. The number of *ex æquo* key bytes is maximal, *i.e.* equal to $w = 16$. The proposed countermeasure is compared in terms of 16th-order success rate with other less educated choices for the masks. This study allows to prove "empirically" that our countermeasure both presents the smallest side-channel leakage and leads to an exhaustive search of maximal difficulty. Eventually, Sec. 6 gives the conclusions and draws some perspectives. The appendices A and B end the paper: they respectively recall known results, for the article to be self-contained, and give interesting properties about the leakage for the optimal mask set in the case $n = 8$ bit.

2 Homomorphic Masking Scheme

In this article, we are interested in the protection of iterated block ciphers (such as AES). They are made up of linear parts (AddRoundKey, MixColumns and ShiftRows for AES) and non-linear parts (in AES: only SubBytes, an 8 bit to 8 bit bijection). We study the security of a *hardware* masking scheme, that is assumed to leak a sensitive variable Z, equal to the sum (bitwise, in $\mathbb{F}_2^n$) of a plaintext X, *first* masked by a random variable M, and *second* added to a secret key byte K:

$$Z = X \oplus M \oplus K \ . \tag{1}$$

Let us denote by n the common bitwidth of X, M and K. The variable X is a public part of the data involved in the computation (*e.g.* the plaintext or the ciphertext). The variable K is a secret (*e.g.* one byte of the key schedule) to recover, constant over all the side-channel observations. Eventually, M is the mask that implements the countermeasure. It is an unpredictable random variable, redrawn fresh at every encryption, that lives in a subset C of $\mathbb{F}_2^n$. Designs featuring a leakage as per Eqn. (1) are numerous. However, it is customary that, in addition to Z, another *share* leaks M (refer for instance to the setup described in [7] for differential side-channel attacks). Keeping the mask variable M along with the masked variable Z allows for an easy unmasking at the end of the computation (see for instance the architecture with *two paths* described in [21]). Masking schemes that leak the mask M solely through Z are those based on precomputed tables (table S' in lieu of S); one example is *Rotating Sboxes Masking* (or RSM [13]). In RSM, the plaintext X enters masked as $X \oplus M$, then goes through linear operations (such as the key K addition, *viz.* AddRoundKey) with respect to the exclusive-or (*i.e.* homomorphically in the group $(\mathbb{F}_2^n, \oplus)$), and traverses the substitution boxes (abridged *sboxes*) that are masked at the input and at the output with masks that make up for the initial masking. This tactic is referred to as *Global Look-Up Table* (GLUT) [16]. For the overhead of this countermeasure to be acceptable, not

all possible masked sboxes are precomputed. This amounts to reducing the size of the set $C \subseteq \mathbb{F}_2^n$ from 2^n to $w = |C| \leqslant 2^n$: the GLUT equation thus becomes:

$$\begin{aligned} &\forall (Y, M_{\text{in}}, M_{\text{out}}) \in \mathbb{F}_2^n \times C \times C, \\ &(Y, M_{\text{in}}, M_{\text{out}}) \quad \mapsto \quad S'(Y, M_{\text{in}}, M_{\text{out}}) \\ &\qquad\qquad = S(Y \oplus M_{\text{in}}) \oplus M_{\text{out}} \in \mathbb{F}_2^n \ . \end{aligned} \tag{2}$$

Thus the size (in units of memory bits) of the required RAM (or ROM) for the masked sboxes is reduced from: $n \times 2^{3n}$ (when $C = \mathbb{F}_2^n$) to $n \times 2^{n+2\log_2(w)}$. Further, the non-linear sbox is implemented in a look-up table: therefore, the value of the sbox is not computed with logic gates, but accessed from an array (RAM or ROM). This means that its evaluation does not produce *glitches* (additional spurious leakage resulting from incomplete transitions caused by races between signals). Therefore, attacks like [15] do not apply. Thus the exploitable leakage writes as:

$$\mathscr{L} = \mathcal{L}(X \oplus M \oplus K) \ , \tag{3}$$

where $\mathcal{L}$ is a degree-one pseudo Boolean function (*i.e.* a function from $\mathbb{F}_2^n$ to $\mathbb{R}$). In the sequel, without loss of generality, we assume $\mathcal{L}$ is balanced. The fact $\mathcal{L}$ is of degree one reflects the hypothesis that each bit is independent, but can be weighted differently. The mask M, uniformly distributed on C, coincides with the mask M_{in} at masked sbox input (in Eqn. (2)).

Also, in RSM, M_{out} is deterministically known from M_{in} (it is the successor in C); this remains secure because it is not expected that an attacker combines the leakage of two rounds of AES, since this involves too many key bytes ($\geqslant 5$ for AES). Therefore, the GLUT equation simplifies as:

$$S'_{\text{RSM}}(Y, M) = S'(Y, M, \text{succ}(M)) \ . \tag{4}$$

This table occupies a memory of $n \times 2^{n+\log_2(w)} = w \times (n2^n)$, in units of bit, *i.e.* the size of w unmasked sboxes. Therefore, when $w = 16 = N_B$, there is not overhead in terms of memory for RSM: the N_B *identical* SubBytes tables are traded for 16 *different* masked sboxes.

3 Vulnerabilities of the Homomorphic Masking

The leakage $\mathscr{L}$ given in Eqn. (3) can be attacked by various means, namely information-theoretic distinguishers, such as the mutual information $\mathsf{I}[\mathscr{L}; X, K = k]$ introduced in [5], or high-order correlation power analysis (HO-CPA) [11].

3.1 Analysis of the HO-CPA

Information-theoretic distinguishers are generic, hence attractive when the leakage is unknown. When the leakage can be modelled, the HO-CPA of minimal order are known to be more efficient than generic distinguishers [23]. In this case, as the leakage is univariate (*i.e.* only one intermediate variable leaks), the adequate attacks are the *zero-offset* correlation power analyses introduced by Waddle and Wagner in [24]. They consist in testing whether the conditional expectation in $X = x$ of a given power $d > 0$ of the leakage depends on x. Thus, the attacker checks for the feasibility of a dth-order zero-offset attack by computing $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d | X, K\right]\right]$, and by selecting the smallest d such as this *inter-class variance* is nonzero. Owing to the linearity in $\mathbb{F}_2^n$ of the key addition in Eqn. (3), $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d | X, K = k\right]\right]$ does not depend on k, thus writes simply $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d | X, K\right]\right]$.

For the implementation to remain small (refer to Sec. 2), the random variable M is not uniformly distributed in $\mathbb{F}_2^n$, but in a *subset* C of $\mathbb{F}_2^n$. In the sequel, we see C as a code, and we call f the indicator of C. This code is characterized by a *distance enumerator polynomial* (see Definition 1 in the next section 4). The smallest strictly positive index of nonzero coefficients is called the *dual distance* $d_C^\perp$ of code C.

It has been proven that the countermeasure cannot be attacked at orders $1, \ldots, d_C^\perp - 1$. Now, at order $d_C^\perp$, the inter-class variance is nonzero, thus there is the possibility of an attack. This result is recalled in Appendix A. In this section, we denote by d the attack order; attacks of order $d < d_C^\perp$ fail, and of order $d \geqslant d_C^\perp$ succeed. However, it is well known that the success rate of the attack decreases with the attack order [17]; this is why the preferred choice for d will be $d_C^\perp$.

We note $\mathscr{L}^\star$ the actual leakage, that is $\mathcal{L}^\star(X \oplus$

$M^\star \oplus k^\star$), where the symbols with stars denote the correct values used by the device, and not necessarily known by the attacker. As stated above, the inter-class variance is not a side-channel distinguisher because it takes the same values for all key bytes $k \in \mathbb{F}_2^n$. Therefore, an attack with a "*model*" shall be used. As explained in [18] and recalled in Lemma 3 in Appendix A, the optimal strategy in an HO-CPA attack is to use as a prediction function $P(x,k) = \mathbb{E}\left[\mathscr{L}^{\star d} | X = x, K = k\right]$, for any guess k on the key byte. But, as $\mathcal{L}^\star$ is *a priori* not known exactly by the attacker, $P(x,k)$ is actually taken equal to $\mathbb{E}\left[\mathscr{L}^d | X = x, K = k\right]$; this is the *model* used in the HO-CPA.

The Pearson correlation coefficient (denoted "ρ") calculated in an HO-CPA is $\rho\left[\mathcal{L}^{\star d}(X \oplus M^\star \oplus k^\star); P(X,k)\right]$. In the sequel, it is abridged as $\rho_{k^\star,k}$, because the other random variables (*viz.* X and $M^\star$) are averaged by the ρ operator. It is proportional to $\mathrm{cov}_{k^\star,k}$, equal to the numerator of $\rho_{k^\star,k}$, and computed in the next Lemma.

Lemma 1. *When the actual leakage is given by Eqn.* (3) *(unknown variables are marked with a star), the covariance $\mathrm{cov}_{k^\star,k}$ distinguisher in a dth-order attack is equal to*[1]

$$\frac{1}{2^n\,|C^\star|\,|C|}\left((\mathcal{L}^{\star d} \otimes f^\star) \otimes (\mathcal{L}^d \otimes f)\right)(k^\star \oplus k)\ .$$

Proof. The covariance $\mathrm{cov}_{k^\star,k}$ is computed as (recall $\mathcal{L}$ and $\mathcal{L}^\star$ are assumed balanced):

$$\mathbb{E}\left[\left(\mathscr{L}^{\star d} | K = k^\star\right) \cdot \left(\mathbb{E}\left[\mathscr{L}^d | X = x, K = k\right]\right)\right] = \frac{2^{-n}}{|C^\star|} \sum_{x \in \mathbb{F}_2^n} \sum_{m^\star \in C^\star} \mathcal{L}^{\star d}(x \oplus m^\star \oplus k^\star) \cdot \left(\frac{1}{|C|} \sum_{m \in C} \mathcal{L}^d(x \oplus m \oplus k)\right).$$

We can introduce the indicators $f^\star$ of $C^\star$ and f of C; hence $\mathrm{cov}_{k^\star,k}$ is equal to:

$$\begin{aligned}
&\frac{1}{2^n|C^\star|} \sum_{x \in \mathbb{F}_2^n} \sum_{m^\star \in \mathbb{F}_2^n} f^\star(m^\star) \cdot \mathcal{L}^{\star d}(x \oplus m^\star \oplus k^\star) \\
&\qquad \cdot \left(\frac{1}{|C|} \sum_{m \in \mathbb{F}_2^n} f(m) \cdot \mathcal{L}^d(x \oplus m \oplus k)\right) \\
&= \frac{1}{2^n|C^\star|} \sum_{x \in \mathbb{F}_2^n} \left(\mathcal{L}^{\star d} \otimes f^\star(x \oplus k^\star)\right) \\
&\qquad \cdot \left(\frac{1}{|C|} \mathcal{L}^d \otimes f(x \oplus k)\right) \\
&= \frac{1}{2^n|C^\star||C|} \left((\mathcal{L}^{\star d} \otimes f^\star) \otimes (\mathcal{L}^d \otimes f)\right)(k^\star \oplus k)\ .
\end{aligned}$$

□

Recall the *convolution product* is commutative and associative, *i.e.* $(\phi_1 \otimes \phi_2) \otimes \phi_3 = \phi_1 \otimes (\phi_2 \otimes \phi_3) = \phi_1 \otimes \phi_2 \otimes \phi_3$. Thus, $\rho_{k^\star,k}$ is equal to:

$$\frac{\left(\mathcal{L}^{\star d} \otimes \mathcal{L}^d \otimes f^\star \otimes f\right)(k^\star \oplus k)}{\sqrt{\left(\mathcal{L}^{\star d} \otimes \mathcal{L}^{\star d} \otimes f^\star \otimes f^\star\right)(0) \cdot (\mathcal{L}^d \otimes \mathcal{L}^d \otimes f \otimes f)(0)}}\ . \tag{5}$$

When d is even, constant terms shall be added in Eqn. (5), which do not change our conclusions.

Remark 1. *Both $\mathrm{cov}_{k^\star,k}$ and $\rho_{k^\star,k}$ depend only in the difference $k^\star \oplus k$ between the correct and the guessed key bytes, and not in the actual key byte $k^\star$. Thus the EIS (Equal Images under different Sub-keys*[2]*) assumption made in [19] holds for the homomorphic masking.*

Let us call $g^\star = \mathcal{L}^{\star d} \otimes f^\star$ and $g = \mathcal{L}^d \otimes f$. In the rest of this section 3.1, we derive optimistic results (that are individually already well known), when the attacker knows $\mathcal{L} = \mathcal{L}^\star$ and $f = f^\star$ (hence $g = g^\star$).

Lemma 2 (Autocorrelation maximum at origin)**.** *Let g be a pseudo-Boolean function. We have $g \otimes g(x) \leqslant g \otimes g(0)$, with equality for $x \in \{x' \in \mathbb{F}_2^n / \forall y \in \mathbb{F}_2^n, g(y) = g(x' \oplus y)\}$ — (Cauchy-Schwarz theorem).*

Remark 2. *We have $\{x \in \mathbb{F}_2^n / \forall y \in \mathbb{F}_2^n, f(y) = f(x \oplus y)\} \subseteq \{x \in \mathbb{F}_2^n / \forall y \in \mathbb{F}_2^n, g(y) = g(x \oplus y)\}$.*

The kernel $\ker(f)$ *of a Boolean function f is the set $\{x \in \mathbb{F}_2^n / \forall y \in \mathbb{F}_2^n, f(x \oplus y) = f(x) \oplus f(y) \oplus f(0)\}$. Any x in this set is a* linear structure *of f (see [4]).* $\ker(f)$ *is the set of $x \in \mathbb{F}_2^n$ such that the derivative*

[1] In this equation, we underline that the function $\frac{1}{2^n|C^\star||C|}\left((\mathcal{L}^{\star d} \otimes f^\star) \otimes (\mathcal{L}^d \otimes f)\right)$ is applied to $k^\star \oplus k$. Please notice it *is not* a product.

[2] In the AES, a *sub-key* is any *byte* of the key.

$D_x f(y) = f(y) \oplus f(x \oplus y)$ *of* f *is constant.* $\ker(f)$ *is endowed with a space vector structure.*

By expressing that $y \mapsto D_x f(y)$ *is null iff it is constant and that this constant is equal to zero, it can be noticed that the set* $\{x \in \mathbb{F}_2^n / \forall y \in \mathbb{F}_2^n, f(y) = f(x \oplus y)\}$ *is equal to* $\ker(f) \cap \{x \in \mathbb{F}_2^n / f(x) = f(0)\}$. *This set is called the* space of null linear structures *of* f. *It is a space vector. Notice that the space of null linear structures of* f *is equal to:*

- $\ker(f) \cap \mathrm{supp}(f)$ *if* $f(0) = 1$, *and to*
- $\ker(f) \cap (\mathbb{F}_2^n \setminus \mathrm{supp}(f))$ *if* $f(0) = 0$.

Let $x_0 \in \mathbb{F}_2^n$ *and* $f' : x \in \mathbb{F}_2^n \mapsto f(x \oplus x_0) \in \mathbb{F}_2$. *The space of linear structures of* f *and* f' *are identical. In particular, the space of null linear structures of* f *and* f' *are identical.*

Let p *a bijective linear function and* $f' : x \in \mathbb{F}_2^n \mapsto f \circ p(x) \in \mathbb{F}_2$. *The space of null linear structures of* f' *is the image of the space of null linear structures of* f *through* p. *This holds in particular for a permutation of the coordinates.*

We say that two codes (non-necessarily linear) are equivalent *if they can be deduced one from the other by a combination of:*

- *addition of constant* $\kappa \in \mathbb{F}_2^n$ *to all codewords, and*
- *permutation* p *of* $[\![1, n]\!]$ *of the codeword coordinates.*

This is a non-canonical *equivalence notion, as usually codes* are not *translated.*

Let f *and* $f^\star$ *be the indicators of two equivalent codes* C *and* $C^\star$. *Then the space of null linear structures of* f *and* $f^\star$ *have the same cardinality.*

If $g = g^\star$, *the HO-CPA (Eqn.* (5)*) is* sound, i.e. *distinguishes the correct key byte, because* $(g \otimes g)(k^\star \oplus k)$ *is maximal if* $k^\star \oplus k = 0$ *(*i.e. *if* $k = k^\star$*). However, this maximum is attained for more key bytes, namely all key bytes whose difference with* $k^\star$ *belongs to the space of null linear structures of* $g = \mathcal{L}^d \otimes f$.

The goal of the countermeasure designer is thus to maximize the size of this space vector. The next subsection proves that the maximal size is obtained for affine codes C.

3.2 Largest Space of Null Linear Structures

Proposition 1. *If* f *is the indicator of a linear code* C, *then* $C \subseteq \ker(f)$. *More precisely,* $\ker(f) = \mathbb{F}_2^n$ *if* C *is a hyperplane, otherwise* $\ker(f) = C$. *Besides, the space of null linear structures of* f *is always* C.

Proof. Let f be the indicator of a space vector C, and $x, y \in C$. We have $f(x \oplus y) = f(y)$ for all y if and only if $x \oplus C = C$, and we have $f(x \oplus y) = f(y) \oplus 1$ for all y if and only if $x \oplus C = \mathbb{F}_2^n \setminus C$. □

Corollary 1. *If* f *is the indicator of an affine code* C, *then the set of the space of null linear structures of* f *is the direction of* C *(the space vector from which* C *is translated).*

Proof. It is well known that for an affine space C, the set of the x such as $x \oplus C = C$ is the direction of C. □

Proposition 2 (Max. No of *ex æquo* key bytes)**.** *Let* f *a Boolean function of weight* w. *The cardinality of the space of null linear structures of* f *is at most* w, *and it equals* w *if and only if* f *is the indicator of an affine code.*

Proof. We call C the support of f. Without loss of generality (by translation), we can assume $0 \in C$. Therefore, the set of x such as $x \oplus C = C$ is included in C and has thus at most $|C|$ elements. It has exactly $|C|$ elements if and only if C is stable by translation of all its elements, which is equivalent to saying that C is a space vector. □

4 Secrecy Analysis of the Homomorphic Masking

Instead of using code C, any equivalent code $C^\star$ can be used (thereby ensuring an equal security level). First of all, it is observed in Sec. 4.1 that such a substitution does not alter the security because distance enumerator polynomials of equivalent codes are identical. In particular their dual distances are the same. Second, we show in Sec. 4.2 that if the exact equivalent code $C^\star$ used in the device is unknown by the attacker, then the homomorphic countermeasure is unconditionally secure. Third, we prove in Sec. 4.3 that if the code $C^\star$ is

known by the attacker, then the correct key byte $k^\star$ cannot be distinguished from bad key bytes $k^\star \oplus k$, where k is an element from the space of null linear structures of $g = \mathcal{L}^d \otimes f$.

4.1 Equivalent Codes Preserve the Security

First of all, we recall the definition of the distance enumerator polynomial.

Definition 1. *The dual distance enumerator polynomial coefficient $B_d^\perp$ of a code C (linear or not), where $0 \leqslant d \leqslant n$, is the coefficient of $X^{n-d}Y^d$ in $\frac{1}{|C|}D_C(X+Y, X-Y)$. We recall that*

$$D_C(X,Y) = \frac{1}{|C|}\sum_{x,y\in C} X^{n-w_H(x\oplus y)}Y^{w_H(x\oplus y)} \ ,$$

where w_H is the Hamming weight ($\forall x \in \mathbb{F}_2^n$, $w_H(x) = \sum_{i=1}^n x_i \in [\![0,n]\!]$, hence w_H takes $(n+1)$ different values).

Proposition 3. *An alternative definition of $B_d^\perp$ is*

$$B_d^\perp = \sum_{\substack{b\in\mathbb{F}_2^n, \text{ such that} \\ w_H(b)=d}} \left(\frac{1}{|C|}\sum_{c\in C}(-1)^{b\cdot c}\right)^2 .$$

Proof. This proposition 3 is well known; we briefly give its proof hereafter. According to the MacWilliams' identity (shown for instance in [2, 8]), $D_C(X+Y, X-Y)$ is equal to

$$\frac{1}{|C|}\sum_{b\in\mathbb{F}_2^n} X^{n-w_H(b)}Y^{w_H(b)}\left(\sum_{c\in C}(-1)^{b\cdot c}\right)^2 .$$

Thus, by identification (in blue color), $\frac{1}{|C|}D_C(X+Y, X-Y)$ is equal to both:

$$\begin{cases} \bullet\ \sum_{d=0}^n B_d^\perp X^{n-d}Y^d \quad \text{and} \\ \bullet\ \frac{1}{|C|^2}\sum_{b\in\mathbb{F}_2^n} X^{n-w_H(b)}Y^{w_H(b)}\left(\sum_{c\in C}(-1)^{b\cdot c}\right)^2 . \end{cases}$$

□

Two codes are equivalent if they can be deduced one from the other by the combination of an addition with a constant to all codewords and a permutation of the codewords coordinates. As those two transformations are commutative, we shall prove the invariance the distance enumerator polynomial under each operation independently. We denote by $B_C^\perp$ the set $(B_d^\perp)_{0\leqslant d\leqslant n}$ for code C.

Remark 3. *Let $\kappa \in \mathbb{F}_2^n$ a constant. Let C and $C^\star = \kappa \oplus C$ two codes ($C^\star = \{\kappa \oplus c, c \in C\}$). Then $B_C^\perp = B_{C^\star}^\perp$.*

Remark 4. *Let p a permutation of $[\![1,n]\!]$. Let C and $C^\star = p(C)$ two codes ($C^\star = \{p(c), c \in C\}$). Then $B_C^\perp = B_{C^\star}^\perp$.*

In the two next subsections 4.2 and 4.3, we illustrate how to take advantage of the invariance of the dual distance by an arbitrary *translation* of the code (which derives new codes called *cosets*). For simplification, we assume that the attacker knows the exact leakage model (and thus takes $\mathcal{L} = \mathcal{L}^\star$), and attacks at the most appropriate order (for her), *i.e.* $d = d_C^\perp$ $(= d_{C^\star}^\perp)$.

4.2 Perfect Secrecy if the Masking Scheme is Private

Let us assume that the used code equivalence classes are known, but not the exact code $C^\star$ (a particular coset). More specifically, we assume that the device uses $C^\star$ and that the attacker guesses $C = \kappa \oplus C^\star$ is used instead. The constant $\kappa \in \mathbb{F}_2^n$ is an *implementation key* byte. The indicator of the two mask sets satisfies: $m^\star \in C^\star \iff f^\star(m^\star) = 1 \iff \kappa \oplus m^\star \in C \iff f(\kappa \oplus m^\star) = 1$. Thus, the coefficient $\mathrm{cov}_{k^\star,k}$ can be computed as in Lemma 1, and is equal to:

$$\mathbb{E}\left[\left(\mathcal{L}^{d_C^\perp}(X \oplus M^\star \oplus k^\star)\right)\cdot\left(\mathbb{E}\left[\mathcal{L}^{d_C^\perp}(X \oplus M \oplus k)|X\right]\right)\right]$$
$$= \frac{1}{2^n\,|C|^2}\left((\mathcal{L}^{d_C^\perp} \otimes f) \otimes (\mathcal{L}^{d_C^\perp} \otimes f)\right)([k^\star \oplus \kappa] \oplus k) \ .$$

Hence, if κ is unknown, Corollary 2 states that the attacker can retrieve those *ex æquo* keys $[k^\star \oplus \kappa] \oplus (\ker(k) \cap \{y \in \mathbb{F}_2^n / g(y) = g(0)\})$, where $g = \mathcal{L}^{d_C^\perp} \otimes f$. So, the attacker gets no information on $k^\star$; indeed, it is *blinded* — as in a *Vernam cipher* — by a secret key byte κ (see the term in square brackets), assumed unknown (that can thus be modeled by a random variable uniformly distributed on $\mathbb{F}_2^n$). Said differently, the correct key byte $k^\star$ is not necessarily within the key candidates

that maximize the correlation coefficient. This conclusion is not in contradiction with Lemma 2, as in this Lemma we assume that the attacker knows the masks ($f = f^\star$) and the leakage model ($\mathcal{L} = \mathcal{L}^\star$). Of course, the same conclusion can be drawn if $C^\star$ is totally secret. Besides, notice that the security argument can be reversed: as the secret key byte $k^\star$ is unknown, the implementation key byte κ is protected (at least from side-channel attacks).

In theory, this solution requires only to store $(k^\star, \kappa)$ secretly in a tamper-proof memory; the secrets are now a pair of key bytes: one *cryptographic key byte* and one *implementation key byte*. However, in practice, to avoid leaking κ when computing the masks set $C^\star = \kappa \oplus C$, all the w codewords $\{c^\star \in C^\star\}$ would be "burnt" in the device (*e.g.* implemented in a $n \times w$-bit RAM or ROM for every sbox). Now, it is known that ROM arrays can be reverse-engineered easily[3]. FPGA designs implement the ROMs as RAMs initialized at the boot time (from an encrypted bitstream), and therefore the homomorphic masking countermeasure can be claimed unconditionally secure. For cost reasons, ASIC designs are likely to base their architecture on a ROM, since ROM blocks are far less costly than RAM blocks.

Besides, for the unconditional security to hold on the whole key (the 16 key bytes of the AES), a different code $C^\star$ must be used for every sbox. Thus, the total number of memory required is $(16 \times w) \times n2^n$ bit, which is 16 times more than announced for RSM (recall Eqn. (4)). Actually, in RSM, there is only one code C for the entire AES, hence a memory requirement of $w \times n2^n$ bit. As there is only one implementation key byte κ for the 16 sboxes, every cryptographic key byte is offset by the same quantity.

Therefore, it is relevant to investigate the security level of the countermeasure if the masks are public or have been reverse-engineered.

4.3 Indistinguishable Keys if the Masking is Public

We assume that the attacker knows $C^\star$ (hence $C = C^\star$, or equivalently $\kappa = 0$). Despite this knowledge, there remains for the attacker a brute-force exhaustive search. For instance, when C is affine and $w = 16$, for each of the 16 key bytes of an AES, the side-channel attack manages to reduce the key byte space size from 2^8 to $w = 2^4$, but stops at this point (there is no other useful information to extract from the side-channel measurements, *i.e.* a purported *back-tracking* is not applicable). The subsequent exhaustive search for the complete key (16 key bytes) can be made prohibitory. In the example of AES, the final key search is $16^{16} = 2^{64}$. Using Rijndael, the initial proposal for AES, with a 32 bytes datapath ($N_B = 32$, and not 16 as in AES [14]), the final search has complexity 2^{128}, *i.e.* the same complexity as that of brute-forcing the 128-bit key (recall that the key $k^\star$ is the primary asset to protect). Notice that the knowledge of the 16 key candidates of one sbox does not help finding the secret key of another sbox. Indeed, such strategy would require to *combine* the leakages of the two sboxes, which is not permitted in parallel implementations of the homomorphic masking, because all the sboxes are evaluated concomitantly.

Remark 5 (Side-Channel Indistinguishability)**.** *Ties (*i.e. ex æquo *keys) exist irrespective of the attack order (*i.e. *also at orders* $> d_C^\perp$*), thus even for information-theoretic attacks (*e.g. *the MIA [5]), and so, even without hypothesis on the leakage model* $\mathcal{L}^\star$*.*

Our homomorphic masking is thus a provably secure side-channel resilient countermeasure[4].

5 Success Rates in the Case of AES

5.1 Studied Codes for $n = 8$

Seven codes C are studied. They are defined in Tab. 1. Their distance enumerator polynomial is given in Tab. 2, and their codewords and null linear structures in Tab. 3.

Here is a verbose description of these codes:

- Code 1 (also nicknamed $M0\_1$) contains only the 0 element, denoted `0x00` $\in \mathbb{F}_2^8$, and thus

[3] Refer for instance to chemical preparations in [6, Fig. 5 & 6] and to automatic recognition with tool `rompar` (presented in this webpage `http://www.aperturelabs.com/tools.html`).

[4] Notice that indistinguishable keys were already discussed in [10], but with the limitation that plaintexts had to be formatted and that attacks should restrict to the first round.

Table 1: Some codes C of length $n = 8$ and size $w = 16$.

Code #	**Nickname**	**Truth table of f, under the form $\sum_{x\in\mathbb{F}_2^n} 2^{\sum_{i=1}^{n} 2^{i-1}x_i} \cdot f(x)$**	$d_C^\perp$	$B_{d_C^\perp}^\perp$
1	$M0\_1$	0x0000000000000000000000000000000000000000000000000000000000000001	1	8
2	$M1\_2$	0x0000000000000000000004000000000000000000002000000000000000000000	2	28
3	$M2\_16$	0x0040000002004002000104208000000001800800000000100000200010080004	3	4.5
4	$M2\_16\_bis$	0x0000000220040400011080000000080080010400020000000800004000020 1	3	3.5
5	$M2\_16\_bis2$	0x0000000110080800022040000000040040020800010000000400008000010 2	3	3.5
6	$M2\_16\_ter$	0x0000000000814200180000240000000042000081000000000000000000182400	3	4
7	$M3\_16$	0x0000800102400000100800000000042004200000000010080000024080010000	4	14

Table 3: Codewords (*first line*) and null linear structures (*second line*) of the codes of Tab. 1.

Code #	**Codewords** and **null linear structures**
1	{0x00}
	{0x00}
2	{0x55, 0xaa}
	{0x00, 0xff}
3	{0x02, 0x13, 0x1c, 0x2d, 0x44, 0x6b, 0x77, 0x78, 0x9f, 0xa5, 0xaa, 0xb0, 0xc1, 0xce, 0xd9, 0xf6}
	{0x00}
4	{0x00, 0x09, 0x1e, 0x33, 0x55, 0x66, 0x6c, 0x7b, 0x87, 0xaf, 0xb4, 0xb8, 0xca, 0xd2, 0xdd, 0xe1}
	{0x00}
5	{0x01, 0x08, 0x1f, 0x32, 0x54, 0x67, 0x6d, 0x7a, 0x86, 0xae, 0xb5, 0xb9, 0xcb, 0xd3, 0xdc, 0xe0}
	{0x00}
6	{0x0a, 0x0d, 0x13, 0x14, 0x60, 0x67, 0x79, 0x7e, 0xa2, 0xa5, 0xbb, 0xbc, 0xc9, 0xce, 0xd0, 0xd7}
	{0x00, 0x07, 0x19, 0x1e}
7	{0x10, 0x1f, 0x26, 0x29, 0x43, 0x4c, 0x75, 0x7a, 0x85, 0x8a, 0xb3, 0xbc, 0xd6, 0xd9, 0xe0, 0xef}
	{0x00, 0x0f, 0x36, 0x39, 0x53, 0x5c, 0x65, 0x6a, 0x95, 0x9a, 0xa3, 0xac, 0xc6, 0xc9, 0xf0, 0xff}

Table 2: Coefficients of the distance enumerator polynomial for the codes of Tab. 1 ($B^{\perp}_{d^{\perp}_C}$ in **bold**).

Code #	$B^{\perp}_0$	$B^{\perp}_1$	$B^{\perp}_2$	$B^{\perp}_3$	$B^{\perp}_4$	$B^{\perp}_5$	$B^{\perp}_6$	$B^{\perp}_7$	$B^{\perp}_8$
1	1	**8**	28	56	70	56	28	8	1
2	1	0	**28**	0	70	0	28	0	1
3	1	0	0	**4.5**	5	3	2	0.5	0
4	1	0	0	**3.5**	7	3.5	0	0	1
5	1	0	0	**3.5**	7	3.5	0	0	1
6	1	0	0	**4**	5	4	2	0	0
7	1	0	0	0	**14**	0	0	0	1

represents an unmasked implementation (of lowest security);

- Code 2 (also nicknamed $M1\_2$) contains two elements, that are 1*'s complement* one from each other; this masking is sufficient to resist first order attacks;

- Codes 3 to 6 contain $w = 16$ elements of $\mathbb{F}_2^8$, and allow to resist up order $d = 3$ attacks. They are nicknamed $M2\_16$, $M2\_16\_bis$, $M2\_16\_bis2$ and $M2\_16\_ter$. The codes 4 & 5 are equivalent: $C_4 = C_5 \oplus \texttt{0x01}$, *i.e.* $f_5(x) = f_4(x \oplus \texttt{0x01})$). As shown in Remark 2, the functions f_4 and f_5 have the same space of null linear structures, namely $\{\texttt{0x00}\}$. However, the space of null linear structures of $g_4 = w_H^3 \otimes f_4$ is $\{\texttt{0x00}\}$ whereas that of $g_5 = w_H^3 \otimes f_5$ is $\{\texttt{0x00}, \texttt{0x08}\}$. This means that a third-order attack will give one maximal key when the countermeasure uses $M2\_16\_bis$, but two *ex æquo* when $M2\_16\_bis2$ is used instead[5] Otherwise, codes $M2\_16$, $M2\_16\_bis$ and $M2\_16\_ter$ are interesting because they have different $B^{\perp}_3$ values.

- Code 7 (also nicknamed $M3\_16$) is an affine code equivalent to the linear code C of parameters $[8, 4, 4]$. It is self-dual, and belongs to the only class of equivalent codes C that have $d^{\perp}_C = 4$ (indeed, $d^{\perp}_C = n - d_{C^{\perp}} = n - d_C = 8 - 4 = 4$). Being affine, its space of null linear structures set is of maximal size $w = 16$ (Proposition 2). Additional properties are discussed in appendix B.

[5] This is an illustration that the space of null linear structures of f can be strictly included in that of g.

Codes 3 to 7 contain $w = 16$ codewords of length $n = 8$, and thus allow to define 16 new sboxes (as per Eqn. (2)), that would implement the homomorphic masking when traded for the genuine (unmasked) sboxes of AES. For example, in RSM, the new sboxes are: $S'_{\text{RSM}} : (Y, i) \in \mathbb{F}_2^8 \times \mathbb{F}_2^4 \mapsto S(Y \oplus M_i) \oplus M_{i+1 \mod 16} \in \mathbb{F}_2^8$, where $C = \{M_i\}_{i \in [\![0,16[\![}$.

In this case, the overhead of the countermeasure is virtually zero, because the number of sboxes is not increased between the unprotected and the protected implementations. Nonetheless, some extra resources are needed for the management of the masks. Nassar *et al.* report in [13] an implementation in Altera Stratix-II FPGA of a homomorphic masking on AES; they achieve a moderate increase of 40% in memory usage, of 48% in user LUT logic, and a decrease of 34% of the maximal operating frequency.

5.2 Simulated High-Order Attacks on RSM

Because of the degeneracy of $w = 16$ key byte guesses with code 7, we can only evaluate whether the correct key is amongst the 16 first ones. Thus, a 16th-order success rate is computed [20]. By definition, it is the probability that the correct key byte is ranked between 16th to 1st, inclusively. For the mask set corresponding to code 7, this is equivalent to saying that the correct key byte equivalence class is ranked first. In particular, when considering the 16th-order success rate, codes 4 and 5 yield the same result, hence only the results for code 4 are displayed.

In the attack, we consider that $\mathscr{L}^{\star}$ is noisy: it is added a random variable N, that follows a centered normal law of variance σ^2. The simulation results are shown in Fig. 1 for attacks at order $d = 3$, with 100 experiments (in which both the seed for the random noise and the correct key are updated). As predicted by the theory, the attack does not succeed on the mask set 7 (*a.k.a.* code $M3\_16$). Indeed, all the correlations between the cube of the centered traces and any key hypothesis are the same (*viz.* zero). Thus, the 16th-order success rate is actually converging to $16/256 = 1/16$ (which is < 1); $1/16$ is the probability of finding the correct key after 16 successive trials without replacement.

Figure 2 reports for attack results at order $d =$

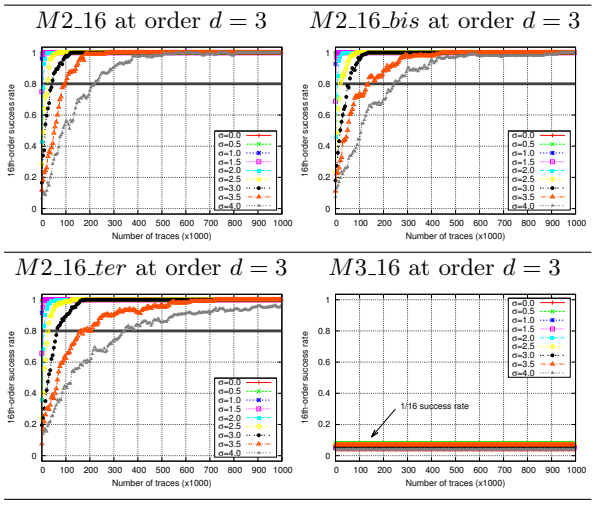

Figure 1: 16th-order success rates for 3rd-order CPA for $\sigma \in \{0, \frac{1}{2}, 1, ..., 4\}$, and a budget of 1 million traces.

4. Interestingly[6], the most secure mask sets are $M2\_16$ and $M2\_16\_ter$, followed by $M2\_16\_bis$, the less secure being $M3\_16$. But of course, $M2\_16$, $M2\_16\_bis$ and $M2\_16\_ter$ are better off be attacked at order $d = 3$, whereas $M3\_16$ cannot.

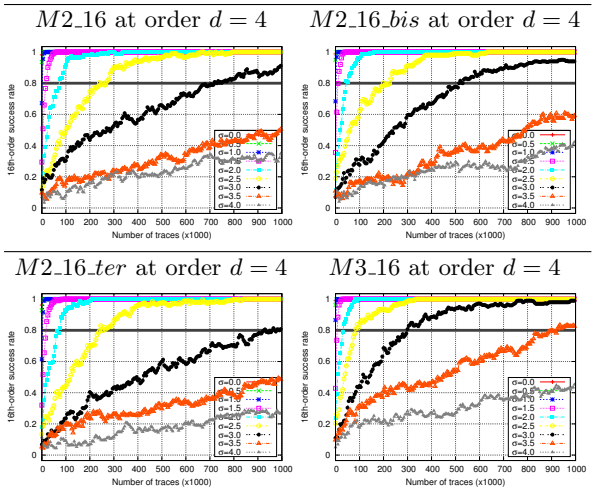

Figure 2: 16th-order success rates for 4th-order CPA for $\sigma \in \{0, \frac{1}{2}, 1, ..., 4\}$, and a budget of 1 million traces.

[6]It can be proven that the success rate of a zero-offset dth-order CPA is all the larger as $B_d^{\perp}$ is large.

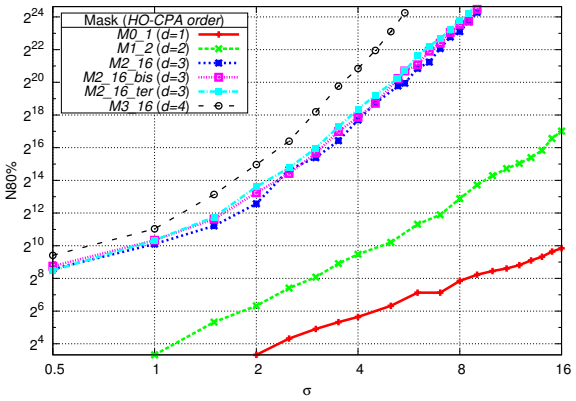

Figure 3: Number of traces to achieve a success rate $\geqslant 80\%$ for various noise standard deviations σ.

We call $N80\%$ the number of traces for the 16th-order success rate to overcome 80%. Informally, it represents the *data complexity* for a successful attack with a confidence of 80%. Each point is obtained as follows: 100 attacks are launched, to derive a success rate. In [24], a high-order masking scheme involving a sharing of a bit into $d + 1$ shares was studied. It assumes all the shares have an additive noise modeled as a centered Gaussian distribution $\mathcal{N}(0, \sigma^2)$. Under these conditions, they prove that:

$$N80\% \quad \geqslant \quad \sigma^{d+4\cdot\log(0.8)/\log(\sigma)} \ . \qquad (6)$$

A recent paper has generalized this law for multi-bit leakage [17]. To confront these results to our masking scheme (where $N80\%$ is evaluated with a success rate of order 16), we represent in Fig. 3 a $N80\%$ *vs* σ law in log-log, for the lowest-order attack that works on the given countermeasures. To obtain this figure, we pushed the simulations up to 32 million ($\approx 2^{25}$) traces. Qualitatively, the slope indeed increases with the attack order d, albeit as a $2d$ (and not d).

6 Conclusions and Perspectives

In this paper, we have studied a homomorphic masking scheme, applicable to block ciphers. While

linear parts of cryptographic algorithms can be computed homomorphically, non-linear parts cannot. Therefore, the non-linear parts (the sboxes) are stored precomputed for a given pool of masks. The cost of the implementation is all the smaller as this pool is small, which motivates for a homomorphic depleted masking scheme. We prove in this article that when masks are $n = 8$ bit long, and with the constraint of precomputing only as many (namely $w = 16$) sboxes as in the original AES algorithm, it is possible to have the AES protected against monovariate attacks of order $d = 3$. Furthermore, we analyze this set under the theory of codes. It is an affine subcode of $\mathbb{F}_2^n$. We know that there exists only one class of equivalent codes C of length $n = 8$, size $w = 16$, and dual distance $d_C^\perp = 4$. Still, if the exact member of the class remains secret, the masking can be proven unconditionally secure. If the code has been compromised (or is deliberately made public), then an attacker will exhibit a set of possible key bytes, that is an affine space containing the correct key plus a space vector of size up to w. This demands for the attacker an extra exhaustive search on top of the side-channel attack; if the algorithm can be tuned, this exhaustive search can be made as complex as desired by increasing the number of sboxes. Finally, we illustrate the complexity of the high-order correlation power analyses by computing success rates of order w: the attack requires more than 1 million traces for a noise standard deviation $\sigma \geqslant 3.5$. Therefore, the homomorphic computation countermeasure presented in this article is a strong solution for implementations that can only be attacked by monovariate analysis, such as FPGA or ASIC implementations of *Rotating Sboxes Masking* (RSM [13]).

As perspectives, we intend to extend RSM and its security proofs to software implementations.

Acknowledgments

The authors of this article would like to thank particularly Shivam Bhasin, Rachid Dafali, Jean-Luc Danger, Thibault Portebœuf, and Thomas Roche for relevant suggestions and interesting discussions about the computation of success rates. Parts of this work have been funded by the MARSHAL+ (*Mechanisms Against Reverse-engineering for Secure Hardware and Algorithms*) FUI #12 project, co-labellized by competitivity clusters System@tic and SCS.

References

[1] S. Bhasin, C. Carlet, and S. Guilley. Theory of masking with codewords in hardware: low-weight dth-order correlation-immune Boolean functions. Cryptology ePrint Archive, Report 2013/303, 2013. `http://eprint.iacr.org/2013/303/`.

[2] C. Carlet. Boolean Functions for Cryptography and Error Correcting Codes: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering. pages 257–397. Cambridge University Press, Y. Crama and P. Hammer eds, 2010.

[3] C. Carlet and S. Guilley. Side-Channel Indistinguishability. In *HASP*, pages 9:1–9:8, New York, NY, USA, June 23-24 2013. ACM.

[4] S. Dubuc. Characterization of Linear Structures. *Des. Codes Cryptography*, 22(1):33–45, 2001.

[5] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *CHES*, volume 5154 of *LNCS*, pages 426–442. Springer, August 10-13 2008. Washington, D.C., USA.

[6] O. Kömmerling and M. G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *WOST '99 (USENIX Workshop on Smartcard Technology)*, pages 9–20, Berkeley, CA, USA, May 10-11 1999. USENIX Association. Chicago, Illinois, USA (On-line paper). ISBN: 1-880446-34-0.

[7] K. Lemke-Rust and C. Paar. Gaussian Mixture Models for Higher-Order Side Channel Analysis. In P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *LNCS*, pages 14–27. Springer, 2007.

[8] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier, Amsterdam, North Holland, 1977. ISBN: 978-0-444-85193-2.

[9] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, `http://www.dpabook.org/`.

[10] M. Medwed, F.-X. Standaert, and A. Joux. Towards Super-Exponential Side-Channel Security with Efficient Leakage-Resilient PRFs. In E. Prouff and P. Schaumont, editors, *CHES*, volume 7428 of *LNCS*, pages 193–212. Springer, 2012.

[11] T. 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.

[12] M. Nassar, S. Guilley, and J.-L. Danger. Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks. In *INDOCRYPT*, volume 7107 of *LNCS*, pages 22–39. Springer, December 11-14 2011. Chennai, Tamil Nadu, India.

[13] M. Nassar, Y. Souissi, S. Guilley, and J.-L. Danger. RSM: a Small and Fast Countermeasure for AES, Secure against First- and Second-order Zero-Offset SCAs. In *DATE*, pages 1173–1178, March 12-16 2012. Dresden, Germany.

[14] NIST/ITL/CSD. Advanced Encryption Standard (AES). FIPS PUB 197, Nov 2001.

[15] T. Popp, M. Kirschbaum, and S. Mangard. Practical attacks on masked hardware. In *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 211–225. Springer, April 20-24 2009. San Francisco, CA, USA.

[16] E. Prouff and M. Rivain. A Generic Method for Secure SBox Implementation. In S. Kim, M. Yung, and H.-W. Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007.

[17] E. Prouff and M. Rivain. Masking against Side Channel Attacks: a Formal Security Proof. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 142–159. Springer, May 2013. Athens, Greece.

[18] E. Prouff, M. Rivain, and R. Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[19] W. Schindler, K. Lemke, and C. Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, editor, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK.

[20] F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.

[21] F.-X. Standaert, G. Rouvroy, and J.-J. Quisquater. FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks. In *FPL*. IEEE, August 2006. Madrid, Spain.

[22] TELECOM ParisTech SEN research group. DPA Contest (4^{th} edition), 2013–2014. `http://www.DPAcontest.org/v4/`.

[23] N. Veyrat-Charvillon and F.-X. Standaert. Mutual Information Analysis: How, When and Why? In *CHES*, volume 5747 of *LNCS*, pages 429–443. Springer, September 6-9 2009. Lausanne, Switzerland.

[24] J. Waddle and D. Wagner. Towards Efficient Second-Order Power Analysis. In *CHES*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004. Cambridge, MA, USA.

A Known Results on Side-Channel Analysis and Masking

The best correlation attack (also refer to [18]) is described in the following Lemma 3.

Lemma 3. *Let $\mathscr{L} \in \mathbb{R}$ and $X \in \mathbb{F}_2^n$ be two random variables, and $P : \mathbb{F}_2^n \to \mathbb{R}$ an arbitrary pseudo-Boolean function. The function $P(X)$ that maximizes the linear correlation coefficient $\rho[\mathscr{L}; P(X)]$ is $x \mapsto P(X = x) = \mathbb{E}[\mathscr{L}|X = x]$.*

Proof. We have (with or without the absolute values):

$$|\rho\left[\mathscr{L}; P(X)\right]| = |\rho\left[\mathscr{L}; \mathbb{E}\left[\mathscr{L}|X\right]\right]| \times |\rho\left[\mathbb{E}\left[\mathscr{L}|X\right]; P(X)\right]| \ .$$

As $|\rho\left[\mathbb{E}\left[\mathscr{L}|X\right]; P(X)\right]| \leqslant 1$, it is clear that $\mathbb{E}\left[\mathscr{L}|X\right]$ is the optimal prediction function $P(X)$. □

Eventually, the security level obtained by the homomorphic masking countermeasure is unveiled in Theorem 1.

Theorem 1 (RSM security)**.** *Let $\mathscr{L} = \mathcal{L}(X \oplus M \oplus k^\star)$, where $\mathcal{L} : \mathbb{F}_2^n \to \mathbb{R}$ is a form, $X \sim \mathcal{U}(\mathbb{F}_2^n)$ and $M \sim \mathcal{U}(C)$ are two random variables, and $k^\star \in \mathbb{F}_2^n$ is a secret key. Then,*

$$d = \min\left\{i > 0,\ \mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^i|X\right]\right] \neq 0\right\} \iff$$
$$C \text{ is a code of dual distance } d_C^\perp = d \ .$$

Proof. It is in Sec. 5 of [12] for $d = 2$. The proof for any d is the Theorem 1 of [1]. □

B Properties of Code $M3\_16$

In this appendix, we study the code $C = M3\_16$. It is the code 7, *i.e.* the last one in Tab. 1, 2 and 3.

This linear code has parameters $[8, 4, 4]$, and is generated from this matrix, given in systematic form:

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right) \ .$$

More precisely, the code $M3\_16$ (also used in [22]) has its coordinates in another order. If we denote by 1 the leftmost coordinate of $M3\_16$, the coordinates are permuted as follows:

$$(1, 2, 3, 4, 5, 6, 7, 8) \to (4, 5, 3, 2, 1, 6, 7, 8) \ .$$

This code has noteworthy properties when the leakage function $\mathcal{L}$ is the Hamming weight.

Property 1. *There are only* three *different distributions of $w_H\left(y \oplus M\right)$, when $M \sim \mathcal{U}(C)$ when $y \in \mathbb{F}_2^8$. Namely, the set of probabilities $(\mathbb{P}\left[w_H\left(y \oplus M\right) = \ell\right])_{\ell \in [\![0,8]\!]}$ are equal to:*

1. $\left(\frac{1}{16}, 0, 0, 0, \frac{14}{16}, 0, 0, 0, \frac{1}{16}\right)$ *if $y \in C$;*
2. $\left(0, \frac{1}{16}, 0, \frac{7}{16}, 0, \frac{7}{16}, 0, \frac{1}{16}, 0\right)$ *if there exists a codeword of Hamming weight* 1 *in $y \oplus C$;*
3. $\left(0, 0, \frac{4}{16}, 0, \frac{8}{16}, 0, \frac{4}{16}, 0, 0\right)$ *if there exists a codeword of Hamming weight* 2 *in $y \oplus C$.*

Proof. As C is a linear code, the distribution of $y_1 \oplus M$ and $y_2 \oplus M$ are identical if $y_1 \oplus y_2 \in C$ (where y_1 and y_2 are two elements of $\mathbb{F}_2^8$). This defines an equivalence relation $\mathcal{R}$, that partitions $\mathbb{F}_2^8$ into 256/16 classes $C_y = y \oplus C$. We call the generator of C_y the smallest element of C_y, seen as list of integers written in hexadecimal. The 16 partitions are generated by `0x00`, `0x01`, `0x02`, `0x03`, `0x04`, `0x05`, `0x06`, `0x07`, `0x10`, `0x11`, `0x12`, `0x13`, `0x14`, `0x15`, `0x16` and `0x17`. Let us now consider the distribution of $w_H(y \oplus M) \in [\![0, 8]\!]$ instead of that of $y \oplus M \in \mathbb{F}_2^8$. The distribution of $w_H\left(y \oplus M\right)$ merges some classes; specifically, the 16 classes are regrouped so as to yield only 3. By exhaustive computer computation, it happens that the distributions are those listed in the property 1, where the generators are grouped as:

1. 16 values of y, namely $y \in \mathcal{C}_1 \doteq C_{\texttt{0x00}}$;
2. 128 values of y, namely $y \in \mathcal{C}_2 \doteq C_{\texttt{0x01}} \cup C_{\texttt{0x02}} \cup C_{\texttt{0x04}} \cup C_{\texttt{0x07}} \cup C_{\texttt{0x10}} \cup C_{\texttt{0x13}} \cup C_{\texttt{0x15}} \cup C_{\texttt{0x16}}$;
3. 112 values of y, namely $y \in \mathcal{C}_3 \doteq C_{\texttt{0x03}} \cup C_{\texttt{0x05}} \cup C_{\texttt{0x06}} \cup C_{\texttt{0x11}} \cup C_{\texttt{0x12}} \cup C_{\texttt{0x14}} \cup C_{\texttt{0x17}}$.

It can be checked, that:

1. each element of the first partition generates a code that contains the null vector;
2. each element of the second partition generates a code that contains a vector of Hamming weight 1;
3. each element of the second partition generates a code that contains a vector of Hamming weight 2.

□

In terms of code theory, the property 1 has a short phrasing.

Property 2 (Equivalent of property 1)**.** *There are only* three *distinct weight enumerator polynomials for all the cosets of* $C = M3\_16$*. They are:* $16 \times \sum_{\ell=0}^{8} \mathbb{P}\left[w_H\left(y \oplus M\right) = \ell\right] X^{n-\ell} Y^{\ell}$*, or specifically:*

- $X^8 + 14X^4Y^4 + Y^8$ *(16 times),*
- $X^7Y + 7X^3Y^5 + 7X^3Y^5 + XY^7$ *(128 times),*
- $4X^6Y^2 + 8X^4Y^4 + 4X^2Y^6$ *(112 times).*

In terms of security, the property 1 yields an interesting result.

Corollary 2. *When* $\mathcal{L} = w_H$*, then the optimal prediction function* $P : y \in \mathbb{F}_2^8 \mapsto \mathbb{E}\left[w_H(y \oplus M)^d\right]$ *takes either only* one *value (at orders* $d < 4$*), or only* three *values (at orders* $d \geq 4$*).*

Proof. At order d,

$$\begin{aligned} P(y) &= \frac{1}{16} \sum_{m \in C} w_H\left(y \oplus m\right)^d \\ &= \frac{1}{16^{1+d}} \begin{cases} 14 \times 4^d + 8^d & \text{if } y \in \mathcal{C}_1 \\ 1 + 7 \times (3^d + 5^d) + 7^d & \text{if } y \in \mathcal{C}_2 \\ 4 \times (2^d + 6^d) + 8 \times 4^d & \text{if } y \in \mathcal{C}_3 \end{cases}. \end{aligned} \tag{7}$$

Now,

- For $d = 1$, $14 \times 4^1 + 8^1 = 1 + 7 \times (3^1 + 5^1) + 7^1 = 4 \times (2^1 + 6^1) + 8 \times 4^1 = 64$;
- For $d = 2$, $14 \times 4^2 + 8^2 = 1 + 7 \times (3^2 + 5^2) + 7^2 = 4 \times (2^2 + 6^2) + 8 \times 4^2 = 288$;
- For $d = 3$, $14 \times 4^3 + 8^3 = 1 + 7 \times (3^3 + 5^3) + 7^3 = 4 \times (2^3 + 6^3) + 8 \times 4^3 = 1408$;
- But for $d \geq 4$, the values in Eqn. (7) are different.

□