Codes for Side-Channel Attacks and Protections

Sylvain Guilley, Annelie Heuser, Olivier Rioul

▶ **To cite this version:**

Sylvain Guilley, Annelie Heuser, Olivier Rioul. Codes for Side-Channel Attacks and Protections. C2SI 2017 - International Conference on Codes, Cryptology, and Information Security, Apr 2017, Rabat, Morocco. pp.35-55, 10.1007/978-3-319-55589-8_3 . hal-01629876

HAL Id: hal-01629876
https://hal.science/hal-01629876

Submitted on 21 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Codes for Side-Channel Attacks and Protections

Sylvain Guilley[1,2], Annelie Heuser[3], and Olivier Rioul[2]

[1] Secure-IC S.A.S., 15 Rue Claude Chappe, Bât. B, 35 510 Cesson-Sévigné, France,
[2] LTCI, Télécom ParisTech, Université Paris-Saclay, 75 013 Paris, France,
[3] IRISA, 263 Avenue Général Leclerc, 35 000 Rennes, France.

Abstract. This article revisits side-channel analysis from the standpoint of coding theory. On the one hand, the attacker is shown to apply an optimal decoding algorithm in order to recover the secret key from the analysis of the side-channel. On the other hand, the side-channel protections are presented as a coding problem where the information is mixed with randomness to weaken as much as possible the sensitive information leaked into the side-channel. Therefore, the field of side-channel analysis is viewed as a struggle between a coder and a decoder. In this paper, we focus on the main results obtained through this analysis. In terms of attacks, we discuss optimal strategy in various practical contexts, such as type of noise, dimensionality of the leakage and of the model, etc. Regarding countermeasures, we give a formal analysis of some masking schemes, including enhancements based on codes contributed via fruitful collaborations with Claude Carlet.

1 Introduction

Digital information is handled by electronic devices, such as smartphones or servers. Some information, such as keys, is sensitive, in the sense that it shall remain confidential. In general, information is present in three states within devices: *at rest*, *in transit*, and *in computation*. The protection of information at rest can be ensured by on-chip encryption in the memories. The same technique applies to the data in transit: the buses can be encrypted (e.g., in a lightweight way, in which case one uses the term *scrambling*). Therefore, the protection of information during computation is the big issue to be dealt with. It is a real challenge, as a computing devices inadvertently leak some information about the data they manipulate. In this context, three questions are of interest:

1. How does an attacker best exploit the leaked information? The situation is similar to that of a decoding problem, and one aims at finding the optimal decoder.
2. Second, the designer (and the end user) aim at being protected against such attacks. Their goal is thus to try and weaken the side-channel. Randomization is one option, referred to as masking in the literature. We will illustrate that it can be seen as the use of code to optimally mix some

random bits into the computations, with the possibility to eventually get rid off this entropy, e.g., at the end of the computation. Another interesting usage of codes is to detect faults in circuits. This dual use of codes is of interest in general security settings, where attacks can choose to be either passive or active. It is also very relevant in the case the circuit is trapped with a Hardware Trojan Horse.

3. Third, it is interesting to know in which respect the circuit leakage favors or not attacks. In particular, we will investigate the effect of glitches as a threat to masking schemes.

Outline. We start with the adversarial strategies in Sec. 2. Protection strategies, especially masking, are presented in Sec. 3. We will show how the circuit itself can contribute to the attack, through the analysis of glitches, in Sec. 4. Conclusions are in Sec. 5. Eventually, appendix A gives some computation evidences why masking protection can be seen as reducing the signal-to-noise ratio, by increasing the noise.

2 Side-channel Analysis as a Decoding Problem

In this section, we first describe the setup and the objective of the attacker. Second, we solve the objective of the attacker in various different setups.

2.1 Setup

We assume the device manipulates some data known by the attacker, such as a plaintext or a ciphertext, called T. This data is mixed with some secret, say a key k^*. The attacker manages to capture some noisy function of T and k^*, and attempts to extract k^*. For this purpose, he will enumerate (manageable) parts of the key (e.g., bytes), denoted k, and choose the key candidate $\hat{k}$ which is the most likely. Therefore, the attack resembles a communication channel, where the input is k^* and the output is $\hat{k}$. The attack is termed successful if $\hat{k}=k^*$.

Two kinds of leakage models are realistic in practice:

1. **direct probing** model, where the attacker uses some kind of probes, each being able to measure one bit,
2. **indirect measurement** of an aggregated function of the bits, using for instance an electromagnetic probe.

These two ways of capturing the signal are, by nature, very different. They are illustrated in Fig. 1.

The first one is noiseless. However, the bits in integrated circuits are nanometric, whereas probes are mesometric. Therefore, only few such probes

(a) probing model (b) bounded-moments model

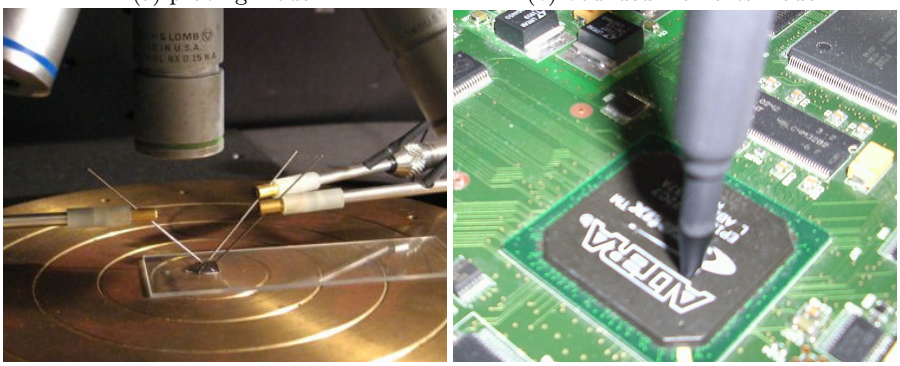


Fig. 1. Settings for side-channel analysis. In the probing model (a), a few bits (here, $d=3$) are measured with dedicated probes. In the bounded moments model (b), the attacker measures an integrated quantity of several bits.

can be used simultaneously. The security parameter is thus linked to the ability for the attacker to recover some useful information out of d probes (where d is typically 1, 2, 3 or 4). Besides, the probing requires a physical access to the wires, which is challenging, since it is possible that the contact breaks the bit to be probed. Such attack is termed semi-invasive, since it leaves an evidence that the circuit has been tampered with (an opening is necessity to insert the probe).

The second one is noisy and also leaks some function of the bits. Therefore, the attacker needs to capture more than one trace to extract some information. This is why we model, in the sequel, traces by random variables. By convention, the variables are printed with capital letters, such as X, when designating a random variable, and with small letters, such as x, when designating the realization of random variables. We also denote by Q the number of queries (= of measurements), and by $\mathbf{x}=(x_1,...,x_Q)$ the vector of measurements. This attack will require a statistical analysis, which in general consists in the study of the leakage probability distribution. This starts in general by the analysis of the leakage moments.

We will link the two models in the case of RSM countermeasure (Sec. 3.5). The next section 2.2 discusses the channel $k^\star \to \hat{k}$, for the second case.

2.2 Example of AWGN Channel

The key recovery setup is illustrated in Fig. 2 (see Fig. 1 in [24]). When the noise is Gaussian and independent from one measurement to others, it is

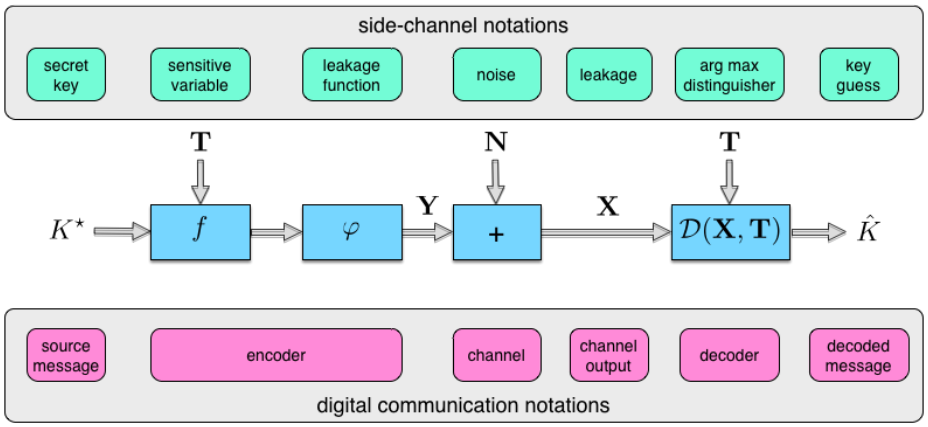


Fig. 2. Side-channel analysis as a communication channel

referred to as AWGN (Additive white Gaussian noise). We write:

$$X = y(T,k^*) + N, \qquad \text{where } N \sim \mathcal{N}(0,\sigma^2). \tag{1}$$

The random variable $y(T,k^*)$ is the aggregated leakage model, and N is the noise (independent from Y). Let n the bitwidth of the key k and of the texts T. The function $y : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}$ is, in practice, the composition two functions $y = \varphi \circ f$, where:

- f is an algorithmic function called *sensitive variable*, such as $f(T,k^*) = S(T \oplus k^*)$, where S is a substitution box, and
- $\varphi : \mathbb{F}_2^n \to \mathbb{R}$ accounts for the way the sensitive variable leaks, such as the Hamming weight $\varphi : z \mapsto w_H(z) = \sum_{i=1}^{n} z_i$.

2.3 Absence of Countermeasures

The optimal distinguisher is the key guess $\hat{k}$ which maximizes the success probability, that is the probability that $\hat{k}$ is actually k^*.

When there is no protection, all the uncertainty resides in the measurement noise. Thus, as the attacker knows T, he also knows $Y = Y(T,k)$ (for all key guess k).

Theorem 1 ([24, Thm. 4]). *In the AWGN setup, the optimal distinguisher is demonstrated to be equal to:*

$$\mathcal{D}_{opt}(\mathbf{x},\mathbf{t}) = \operatorname{argmin}_k \|\mathbf{x} - \mathbf{y}(\mathbf{t},k)\|_2^2 = \operatorname{argmax}_k \langle \mathbf{x} | \mathbf{y}(\mathbf{t},k) \rangle - \frac{1}{2}\|\mathbf{y}(\mathbf{t},k)\|_2^2, \tag{2}$$

where $\|\cdot\|_2$ *is the Euclidean norm and* $\langle \cdot | \cdot \rangle$ *is the canonical scalar product.*

2.4 Multivariate and Multimodel Setting

In the multivariate and multimodel case, the attacker is able to collect:

- not only one sample, but D (dimensionality) samples, and
- each function of the bits (e.g., $z \mapsto 1$, $z \mapsto z_i$ for $1 \le i \le n$, but also any selection of $z \mapsto \bigwedge_{i \in I} z_i$ where $I \subseteq \mathbb{F}_2^n$) has a different contribution.

We call S the number of models, and α the $D \times S$ matrix of the leakages, such that Eqn. (1) is generalized as:

$$\mathbf{X} = \alpha \mathbf{y}(\mathbf{T}, k^*) + \mathbf{N}, \qquad \text{where } \mathbf{N} \sim \mathcal{N}(\mathbf{0}, \Sigma), \tag{3}$$

where $\mathbf{N}$ is multivariate normal of $D \times D$ covariance matrix Σ, and $\mathbf{Y} = \mathbf{y}(\mathbf{T}, k^*)$ is set of S models (e.g., $S = 1$ if the leakage model is the Hamming weight, or $S = n+1$ if there is a non-zero offset (such offset is modeled by $z \mapsto 1$) and each bit $1 \le i \le n$ of the leakage model leaks differently). In this case also, **boldface** variables are vectorial (either *multivariate* or *multimodel*).

We have a generalization of Theorem 1:

Theorem 2 ([7, Thm. 1]). *Let us define* $\mathbf{x}' = \Sigma^{-1/2}\mathbf{x}$ *and* $\alpha' = \Sigma^{-1/2}\alpha$. *Then, in the multivariate and multimodel AWGN setup, the optimal distinguisher is demonstrated to be equal to:*

$$\begin{aligned} \mathcal{D}_{opt}^{D,S}(\mathbf{x}, \mathbf{t}) &= \operatorname{argmin}_k \sum_{d=1}^{D} \|\mathbf{x}'_d - \alpha'_d \mathbf{y}((\mathbf{t}, k)\|_2^2 \\ &= \operatorname{argmax}_k \operatorname{tr}\left(\mathbf{x}' \left(\alpha' \mathbf{y}(\mathbf{t}, k)\right)^{\mathsf{T}}\right) - \frac{1}{2}\left\|\alpha' \mathbf{y}(\mathbf{t}, k)\right\|_F^2, \end{aligned}$$

where $\operatorname{tr}(\cdot)$ *is the trace operator of a square matrix and* $\|\cdot\|_F$ *is the Frobenius normal of a (rectangular) matrix.*

2.5 Collision

In some situations, the attacker does not know the leakage function $y = \varphi \circ f$, but knows that it is reused several times for different bytes, say $L > 1$. We denote by $x^{(\cdot)} = (x^{(1)}, \ldots, x^{(\ell)}, \ldots, x^{(L)})$ the L leakages. Therefore, the optimal attack consists in a collision attack where all the coefficients of the leakage function are regressed.

Theorem 3 ([5, Thm. 2.5]). *The optimal collision attack is:*

$$\mathcal{D}_{opt}^{L}(\mathbf{x}^{(\cdot)}, \mathbf{t}^{(\cdot)}) = \operatorname{argmax}_{k^{(\cdot)} \in (\mathbb{F}_2^n)^L} \sum_{u \in \mathbb{F}_2^n} \frac{\left(\sum_\ell \sum_{q / t_q^{(\ell)} \oplus k^{(\ell)} = u} x_q^{(\ell)}\right)^2}{\sum_\ell \sum_{q / t_q^{(\ell)} \oplus k^{(\ell)} = u} 1} \ .$$

Notice that in general, this attack allows to recover $(L-1)$ n-bit keys when the collision is involving L samples with identical leakage model.

2.6 General Setting with Countermeasures

In general, the device defends itself, by the implementation of protections. Masking (cf. Sec. 3.2) is one of them. In the expression of y, in addition to T and k, another random variable M is introduced, called the mask, unknown to the attacker. It is usually assumed that it is uniformly distributed.

Theorem 4 ([8, Proposition 8]). *The optimal attack in case of masking countermeasure is:*

$$\mathcal{D}_{opt}^{M;L}(\mathbf{x}^{(\cdot)},\mathbf{t}^{(\cdot)}) = \operatorname{argmax}_k \sum_{q=1}^{Q} \log\left\{\sum_m \exp\left\{\sum_{d=1}^{D} \frac{1}{\sigma^{(d)^2}}\left(x_q^{(d)} y_q^{(d)} - \frac{1}{2} y_q^{(d)^2}\right)\right\}\right\},$$

assuming that the noise at each sample d is normal of variance $\sigma^{(d)^2}$.

2.7 Link Between Success Probability, SNR and Leakage Function

The optimal distinguishers $\mathcal{D}_{\text{opt}}$ given in various scenarios ($\mathcal{D}_{\text{opt}}$ for nominal case in Sec. 2.3, $\mathcal{D}_{\text{opt}}^{D,S}$ for multivariate and multimodel case in Sec. 2.3, $\mathcal{D}_{\text{opt}}^{L}$ for the collision case in Sec. 2.5, and $\mathcal{D}_{\text{opt}}^{M;L}$ for the masked case in Sec. 2.6) allow to recover the secret key with the largest success rate (denoted as SR), but do not help in predicting the number of traces to reach a given success rate (or vice-versa).

Such relationship can be easily derived from the analysis of so-called *first-order exponents* [23]. Let us denote $\mathcal{A}_{\text{opt}}(\mathbf{x},\mathbf{t},k)$ the argument of maximization in either of $\mathcal{D}_{\text{opt}}$, $\mathcal{D}_{\text{opt}}^{D,S}$, $\mathcal{D}_{\text{opt}}^{L}$ or $\mathcal{D}_{\text{opt}}^{M;L}$. We have:

Theorem 5 ([23, Corollary 1]).

$$1-\text{SR}(\mathcal{D}) \approx e^{-Q\cdot \text{SE}(\mathcal{D})} \tag{4}$$

where the first-order success exponent $\text{SE}(\mathcal{D})$ *is equal to:*

$$\text{SE}(\mathcal{D}) = \frac{1}{2} \min_{k\neq k^*} \frac{\left(\mathcal{A}_{opt}(\mathbf{x},\mathbf{t},k^*) - \mathcal{A}_{opt}(\mathbf{x},\mathbf{t},k)\right)^2}{\text{Var}\left(\mathcal{A}_{opt}(\mathbf{x},\mathbf{t},k^*) - \mathcal{A}_{opt}(\mathbf{x},\mathbf{t},k)\right)}. \tag{5}$$

For the sake of the introduction of a signal-to-noise, we rewrite Eqn. (1) as:

$X = \alpha y(T,k^*) + N$, where $\mathbb{E}(y(T,k^*)) = 0$, $\text{Var}(y(T,k^*)) = 1$ and $N \sim \mathcal{N}(0,\sigma^2)$.

Let us introduce generalized *confusion coefficients* [20]:

Definition 6 (General 2-way confusion coefficients [23, Def. 8 & 10]). *For $k \neq k^*$ we define*

$$\kappa(k^*,k) = \mathbb{E}\left\{\left(\frac{Y(k^*)-Y(k)}{2}\right)^2\right\}, \tag{6}$$

$$\kappa'(k^*,k) = \mathbb{E}\left\{\left(\frac{Y(k^*)-Y(k)}{2}\right)^4\right\}. \tag{7}$$

For example, for the optimal distinguisher in the nominal case, the success exponent expression is:

Lemma 7 (SE for the optimal distinguisher, [23, Proposition 5]). *The success exponent for the optimal distinguisher takes the closed-form expression*

$$\mathrm{SE}(\mathcal{D}_{opt}) = \frac{1}{2} \min_{k \neq k^*} \frac{\alpha^2 \kappa^2(k^*,k)}{\sigma^2 \kappa(k^*,k) + \alpha^2(\kappa'(k^*,k) - \kappa(k^*,k)^2}. \tag{8}$$

This closed-form expression simplifies for high noise $\sigma \gg \alpha$ in a simple equation:

Corollary 8 ([23, Corollary 2]). *For low SNR:*

$$\mathrm{SE}(\mathcal{D}_{opt}) \approx \frac{1}{2} \min_{k \neq k^*} \frac{\alpha^2 \kappa^2(k^*,k)}{\sigma^2 \kappa(k^*,k)} = \frac{1}{2} \cdot \mathrm{SNR} \cdot \min_{k \neq k^*} \kappa(k^*,k), \tag{9}$$

where $\mathrm{SNR} = \alpha^2/\sigma^2$ is the signal-to-noise ratio (see [6] for the definition of SNR in the multivariate case).

3 Side-Channel Protection

Side-channel attacks threaten the security of cryptographic implementations. Protections against such attacks can be devised using the coding theory. We illustrate in this section several techniques which randomize leakages in a view to decorrelate them from the internally manipulated data, and that (in some cases) also allow to detect malicious fault injections.

3.1 Strategies to Thwart Side-Channel Attacks

As discussed in Sec. 2.7 (especially in (9)), the success of an attack is all the larger as the leakage function has a higher confusion (6) and the SNR is high. However, the input of confusion is limited, since $0 \leq \min_{k \neq k^*} \kappa(k^*,k) \leq 1/2$ is bounded. Moreover, the defender cannot always change the algorithm nor its leakage model, that is $\min_{k \neq k^*} \kappa(k^*,k)$ is fixed. Thus, the defender is better off focusing on the reduction of the SNR.

This can be achieved in two flavors:

1. reduce the signal, as done in strategies aiming at flattening the leakage. This is easily achieved for some side-channels, such as timing: the execution time is made constant, e.g., by inserting dummy instructions or by balancing the code in each branch when the control flow forks. However, balancing an analogue quantity (such as power or electromagnetic field) is more challenging, let alone because of process variations, two identical gates or structures behave differently after fabrication. For instance, this is the working factor of physically unclonable functions (PUFs). Therefore, the quality of the protection depends on the ability of the fabrication plant to produce reproducible patterns. This fact naturally limits the quality of the designer's work, hence does not encourage to reach very high levels of security. In case this case, the second option is preferred;
2. increase the noise, by resorting to some extra random variables independant of that involved in the leakage function. Obviously, some artificial noise can be easily produced: one practical example consists in running an algorithm known to produce a lot of leakage (such as an asymmetrical engine, e.g., RSA) in parallel to the algorithm to protect. However, there remains the risk that the attacker manages, by a subtle placement of the probes, to limit or completely avoid the externally added noise; imagine an attacker with a very selective electromagnetic probe which would place its probe over the targetted algorithm, which is micrometers apart from the noise source (RSA). Therefore, it sounds wiser to entangle the computation and the random variables. This is what is achieved by so-called masking schemes. Appendix A explains why masking reduces the SNR.

Notice that the two strategies are orthogonal, that is, it is beneficial to employ them at the same time. Still, in the sequel, we will focus on masking, since it allows (at least in theory) to increase the noise at the maximal extent.

3.2 Masking Schemes

Masking schemes have been introduced to obfuscate the internals of a computation, in a view to make it more difficult to be attacked. The strategy in masking is based on randomization:

- for *data* (e.g., in algorithms with constant-execution flow, such as AES), and
- for *operations* (e.g., in algorithms where the sequence of operations leak some secrets, such as RSA).

In practice, a masking scheme consists in four algorithms, as depicted in Fig. 3.

Initially, the input data must be masked, thanks to a first algorithm. Second, the masked data is manipulated, so as to implement the intended cryptographic

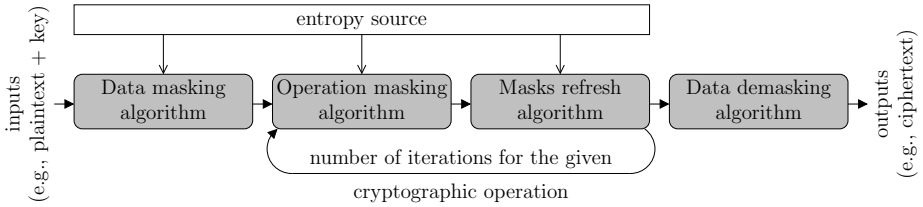


Fig. 3. Masking schemes

operation. Many techniques exist. One way to envision masking is to see all the operations making up the cryptographic function as look-up tables. In this case, the masked look-up tables can be implemented as [38, Tab. 1]:

- new larger look-up tables, where the masking material is now part of the addressing strategy,
- table recomputation specifically for the current mask, or
- computation style which is able to operate on masked data.

After the operation has been computed, it can be necessary to refresh the masks. Indeed, if the value is intended to be used more than once, then some masks would be duplicated during the computation. It is thus wise to re-randomize the current masks. Eventually, at the end of the computation, the masked data shall be freed from its mask. Hence a demasking step. The first three algorithms require entropy, whereas the last one destroys entropy.

3.3 Security of Masking Schemes

It is easy to measure the amount of entropy consumed by a masking scheme (see top of Fig. 3). However, this does not obviously reflect its actual security level. Indeed, the entropy can be wasted, e.g., by being badly used: XORing together entropy reduces it, while bringing no additional difficulty for the attacker.

The first attempt to measure security arise from [1, Definition 1]. The order is defined as the minimum number of *intermediate values* an attacker must collect to recover part of the secret. In this framework, the overall security is that of the weakest link.

Still, the exact definition of an intermediate variable is unclear. The difficulty arises from the fact the designer would like to link the security to properties of its design. However, the intermediate variables encompass different notions depending on the refinement stage: after compilations, variables are mapped to internal resources. Thus, the granularity [1, §3] can change between

the cryptographic algorithm specification, the source code, the machine code, and what is actually executed on the device.

Some early works considered intermediate values are bits, such as in *private circuits* [26,25]. This makes sense for hardware circuits, for which (in general CMOS processes) an equipotential has only two licit values, that is carries one bit. However, private circuits have been extended to software implementations (see e.g. [41]), where intermediate variables become bitvectors of the machine word length. But after considering some new threats, such as glitches, a new trend has consisted in looking back to bit-oriented masking. This is typically the case of *threshold implementations* [36], where the granularity is again the bit.

In this article, we are interested with the lowest possible level of security analysis, hence we consider that intermediate variables are bits.

3.4 Orthogonal Direct Sum Masking (ODSM): a Masking Scheme Based on Codes

We illustrate in this section several masking schemes, and show in which respect they relate to coding theory.

We will show that the two security notions related to masking (*probing* and *bounded-moment* models) are equivalent when conducting analyses at bit-level. We model a circuit as a parallel composition of bits, seen as elements of $\mathbb{F}_2$. The exemple, when there are n wires in the circuit, we model the circuit state as an element of $\mathbb{F}_2^n$, that is the Cartesian product $\mathbb{F}_2 \times ... \times \mathbb{F}_2$.

At this stage, we use the following new notations. Let X a k-bit information word to be concealed. Let Y an $(n-k)$-bit mask used to protect X. The protected variable is $Z = XG + YH$, where:

- G is an $k \times n$ generating matrix of a code,
- H is an $(n-k) \times n$ generating matrix of a code of dual distance $d+1$,
- $+$ is the bitwise addition in $\mathbb{F}_2^n$, sometimes also denoted by $\oplus$.

The random variable YH is the mask. In practice, the bits making up Z can be manipulated in whatever order, i.e., they can even be scheduled to be manipulated one after the other, like in a bitslice implementation. We call Z an encoding with codes, or ODSM [3].

Then, we have the following twain theorems.

Theorem 9. *Encoding with codes is secure against probing of order d.*

Proof. By definition of a code of dual distance $d+1$, any tuple of less than d coordinates is uniformly distributed [9]. Thus, if the attacker probes up to d (inclusive) wires, this word seen as an element of $\mathbb{F}_2^d$ is perfectly masked. Therefore, no information on X can be recovered. □

Theorem 10 (Masking with codes is d-th order secure in the bounded-moments model). *For all pseudo-Boolean function $\psi : \mathbb{F}_2^n \to \mathbb{R}$ (leakage function, denoted $y = \varphi \circ f$ in Sec. 2.2) of degree $d^\circ(\psi) \leq d$, we have*

$$\mathsf{Var}(\mathbb{E}(\psi(XG+YH|X))) = 0. \tag{10}$$

Proof. Let ψ' the indicator of the code generated by H. Since H has dual-distance $d+1$, we have that for all $z \in \mathbb{F}_2^n$, $0 < w_H(z) \leq d$, $\hat{\psi'}(z) = 0$, where $\hat{\psi'}(z) = \sum_{z' \in \mathbb{F}_2^n} \psi'(z)(-1)^{z' \cdot z}$. Now, owing to Lemma 1 in [4], we also know that for all $z \in \mathbb{F}_2^n$, $w_H(z) > d^\circ(\psi)$, $\hat{\psi}(z) = 0$.

Now, we must prove that $\mathsf{Var}(\mathbb{E}(\psi(XG+YH|X))) = 0$, that for all $x \in \mathbb{F}_2^k$, $\sum_{y \in \mathbb{F}_2^{n-k}} \psi(xG + yH) = \sum_{z \in \mathbb{F}_2^n} \psi(xG + z)\psi'(z) = (\psi \otimes \psi')(xG)$ is the same, where $\otimes$ is the convolution product.

Actually, we can prove more than that, namely that $\psi \otimes \psi'$ is constant on the full $\mathbb{F}_2^n$. This is equivalent to proving that $\widehat{\psi \otimes \psi'} = \hat{\psi}\hat{\psi'}$ is equal to zero on $\mathbb{F}_2^n \setminus \{0\}$. Indeed, let $z \in \mathbb{F}_2^n$, $z \neq 0$. If $w_H(z) > d^\circ(\psi)$, then $\hat{\psi}(z) = 0$. And if $w_H(z) \leq d^\circ(\psi) \leq d$, then $\hat{\psi'}(z) = 0$. So, in both cases, we have $\hat{\psi}(z)\hat{\psi'}(z) = 0$. □

Notice that the function $\psi : \mathbb{F}_2^n \to \mathbb{R}$ such that $\psi(x) = \sum_{i=0}^{n-1} x_i 2^i$, has degree one. It is sometimes (abusively) referred to as the identity function. Obviously, if the attacker gets to know $\psi(Z)$, then he can recover Z, hence deduce X by projection on subspace vector C. But this is not our security hypothesis. Our result from Theorem 10 (and in particular its Eqn. (10)) is that the inter-class variance of $\psi(Z)$ knowing X is equal to zero, for all $d^\circ(\psi) \leq d$.

In Equation (10), the degree of function ψ can be accounted by two reasons:

1. High-degree leakage in $y = \varphi \circ f$, arising from spurious physical interactions within the circuit, and occurring before a leakage can be observed outside of the circuit boundary. Such high-degree leakage can be caused by *glitches* (see Sec. 4), *capacitive coupling* such as *cross-talk*, *IR drop*, *ground/substrate coupling*, etc. (refer to [18, Sec. 4.2]);
2. Intentional combination function computed by the attacker, which can be: multivariate (which involves a *product* of several shares), or monovariate (raising to a given *power*, hence necessarily high-order *zero-offset* [42]).

As another remark, we notice that, although it is not strictly mandatory, the randomized variable Z can be manipulated by subwords, a bit like for classical masking, where the subwords coincide with shares.

Let us give the example of the look-up table, in the case $k = 8$ and $n = 16$. We know that we can reach 4-th order security [4]. But we can decide not to

manipulate only Z as such, but to cut it into two parts, $Z=(Z_H,Z_L)$, where $Z_H,Z_L\in\mathbb{F}_2^8$. This cut is motivated by the adequation between the masking scheme and the machine architecture, where maybe the basic register size is 8 bits. Then, we also cut the T-table(s) into two tables, namely T_H and T_L, both of 256 bytes. The algorithm 1 allows to evaluate the T-table using bytes only, i.e., without placing Z_H and Z_L side-by-side for all data Z.

Input :

- $(z_H,z_L)\in\mathbb{F}_2^8\times\mathbb{F}_2^8$
- T_H,T_L, two tables of size 2^{16} bytes

Output : The result of the lookup $(T_H[z_H\times 2^8+z_L],T_L[z_H\times 2^8+z_L])$

```
Initialize z'_H ∈ F_2^8 and z'_L ∈ F_2^8 to zero
for h=0 to 2^8−1 do
    for l=0 to 2^8−1 do
        z'_H ← z'_H ⊕ T_H[h×2^8+l] ∧ (h=z_H) ∧ (l=z_L)
        z'_L ← z'_L ⊕ T_L[h×2^8+l] ∧ (h=z_H) ∧ (l=z_L)
    end
end
return (z'_H, z'_L)
```

Algorithm 1: S-box evaluation by block, without ever using a 16-bit word

3.5 Illustration for Some Coding-Based Masking Schemes

In the previous section, we have shown with Theorems 9 and 10 that the two models (bit-level probing and bounded moments) are equivalent, which motivates to consider the probing model at bit level (as opposed to at word level, as done in many papers (to cite a few: [16,19]). We give hereafter some examples of masking with codes at bit-level.

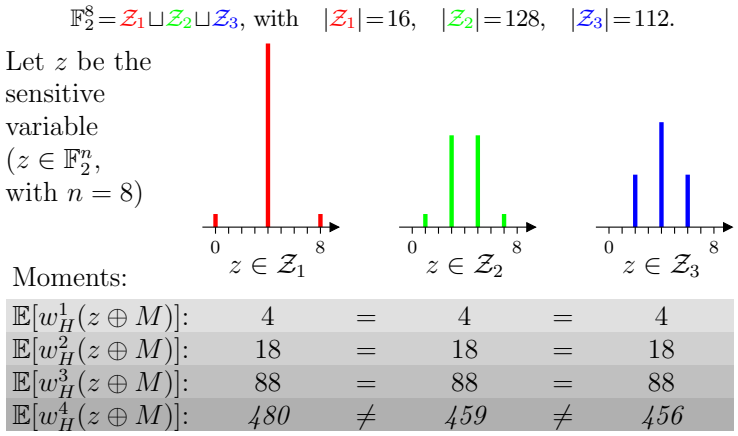
Perfect masking The masks M_1, M_2, etc. are chosen uniformly in $\mathbb{F}_2^k$. We assume here that $k|n$. It is possible to see perfect masking as a special case of ODSM [3], where:

$$G=\begin{pmatrix}I_k & 0 & 0 & \dots & 0\end{pmatrix} \qquad \text{and} \qquad H=\begin{pmatrix}I_k & I_k & 0 & \dots & 0\\ I_k & 0 & I_k & \dots & 0\\ \vdots & 0 & 0 & \ddots & 0\\ I_k & 0 & 0 & \dots & I_k\end{pmatrix}. \tag{11}$$

Rotating Substitution-box Masking (RSM [33]) Let us illustrate RSM on $n=8$ bits. The mask M is chosen uniformly in:

- the set $\mathcal{C}_0=\{\texttt{0x00}\}$ for no resistance,
- the set $\mathcal{C}_1=\{\texttt{0x00},\texttt{0xff}\}$ for resistance to first-order attacks,
- the set $\mathcal{C}_2$, a non-linear code of length 8, size 12 and dual distance $d^{\perp}_{\mathcal{C}_2}=3$,
- the set $\mathcal{C}_3$, a linear code of length 8, dimension 4 and dual distance $d^{\perp}_{\mathcal{C}_3}=4$. This code is fully described in [15]. It is a self-dual code of parameters [8,4,4].

The case $\mathcal{C}_3$ is interesting since there are sixteen masks, hence (in hardware), the sixteen Substitution-boxes (S) of an algorithm such as AES can be implemented masked. When $\varphi=w_H$ and $Z=f(T,k^*)=S(T\oplus k^*)$, then the leakage distributions $X=\varphi(Z\oplus M)$ are represented in Fig. 4.



Moments:

$\mathbb{E}[w_H^1(z\oplus M)]$:	4	=	4	=	4
$\mathbb{E}[w_H^2(z\oplus M)]$:	18	=	18	=	18
$\mathbb{E}[w_H^3(z\oplus M)]$:	88	=	88	=	88
$\mathbb{E}[w_H^4(z\oplus M)]$:	*480*	$\neq$	*459*	$\neq$	*456*

Fig. 4. Leakage distribution of RSM using $M\sim\mathcal{U}(\mathcal{C}_3)$ on $n=8$ bits

RSM involves a random index, that is the choice of the initial codeword in $\mathcal{C}_d$, for a protection order of d. This choice can be done in a leak-free manner by using a *one-hot* representation. In the case of $\mathcal{C}_3$, sixteen such indices can be selected. The *one-hot* representation is given in Fig. 5. The random index is selected at random initially; then, from round to round, it is simply shifted.

Leakage Squeezing (LS) In leakage squeezing, the shares are like for perfect masking, except that some bijective functions (linear or non-linear) are applied to them, thereby mixing bits better [17,12,10,13].

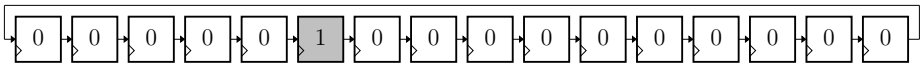


Fig. 5. Example of *one-hot* counter (out of 16), used to designate the round index position

Results For the illustration of the *bounded moment* model, we use for our illustrations the *Hamming weight* leakage model. Notice that any other first-order leakage model would yield comparable results.

Also, we illustrate the leakage based on two extreme plaintexts, that is `0x00` and `0xff`. However, in some situations, these two plaintexts lead to the same leakage (e.g., for symmetry reasons).

In all the presented schemes, security holds only provided there is no *high-order leakage*. Said differently, it is possible to consider that there is a high-order leakage. For instance, in recap figure 6, the indicated security order is the attack total order. The total attack order is the sum of multiplicative contribution from the hardware and the operations carried out by the attacker. That is, poor hardware which couples bits contributes to facilitates attacks by combining bits.

3.6 Masking and Faults Detection

Codes are also suitable tools when both side-channel leakage must be masked and faults must be detected. This need is general in cryptography, and has specific applications when thwarting Hardware Trojan Horses (HTH) [35,34,11]. Indeed, the *activation part* of a HTH is impeded by masking, whereas the *payload part* is caught red-handed by a detection code.

4 Leakage Model and Glitches

The term *glitch* refers to a (some) non-functional transition(s) occurring in combinational logic. They exist because combinational gates are non-synchronizing, i.e., they evaluate as soon as one input arrive. In terms of *hardware description languages* (VHDL, Verilog, etc.), they are modelled as processes where all inputs belong to the sensitivity list. Thus, for the vast majority of gates with many inputs, there is the possibility of a race between the inputs. Therefore, some gates can evaluate several times within one clock period. Actually, the deeper the combinational gates, the more likely it is that:

– there is a large timing difference between the inputs, thereby generating new glitches, and

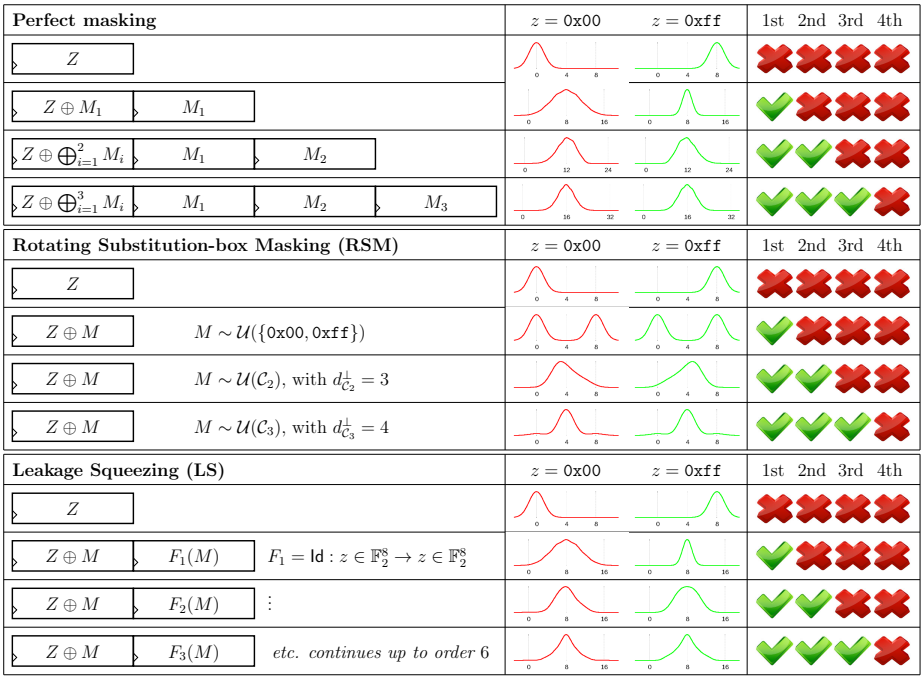


Fig. 6. Security level of several masking schemes. The order d=1,2,3,4 corresponds both to the number of probes (see Fig. 1(a)) used by the attacker and to the moment of leakage when the attacker uses an integrating probe (see Fig. 1(b))

– some input is already the output of a glitching gate, thereby amplifying the number of glitches.

It is known that glitches can defeat masking schemes [30,29,31]. Some masking schemes which somehow *tolerate* [21,36,40,22] or *avoid* glitches [28,32] have been put forward. However, the real negative effect of glitches on security is usually perceived in a *qualitative* manner.

Therefore, we would like to account *quantitatively* for the effect of glitches. Let us start by an illustrative example[1], provided in Fig. 7. The upper part of this figure represents a pipeline, where some combinational gates (`AND` gates represented by ⊐ and `XOR` gate represented by ⊐) form a partial netlist between two barriers of flip-flops (`DFF` gates represented by □). For the sake

[1] Leakage of combinational gates arises from their *transitions*, that is it actually depends both on their *initial* and *final* states. In order to make the two use-cases presented in Fig. 7 and 8 simple to understand, we initialize the netlists at an all-zero state, thence we can reason only on the final values.

of this explanation, all the gates are assumed to have the same propagation time, namely 1 ns. The lower part of this figure gives the chronograms of the execution of this netlist, when initially all signals are set to zero. It appears that, owing to the difference of paths between the two inputs of the final `XOR` gate, this gate generates a glitch, highlighted with symbol ●, which lasts 3 ns, between time 1 and 4 ns within the depicted clock period. The condition for this glitch to appear is the following: $x_1 \wedge x_2 \wedge x_3 \wedge x_4$. This means that this glitch is a 4th-order leakage. So, if the masking scheme is only 3rd-order resistant, the setup of Fig. 7 would generate a glitch which compromises the security in a 1st-order side-channel attack. That is, the circuit itself contributes to the attack, in combining the bits on behalf of the attacker.

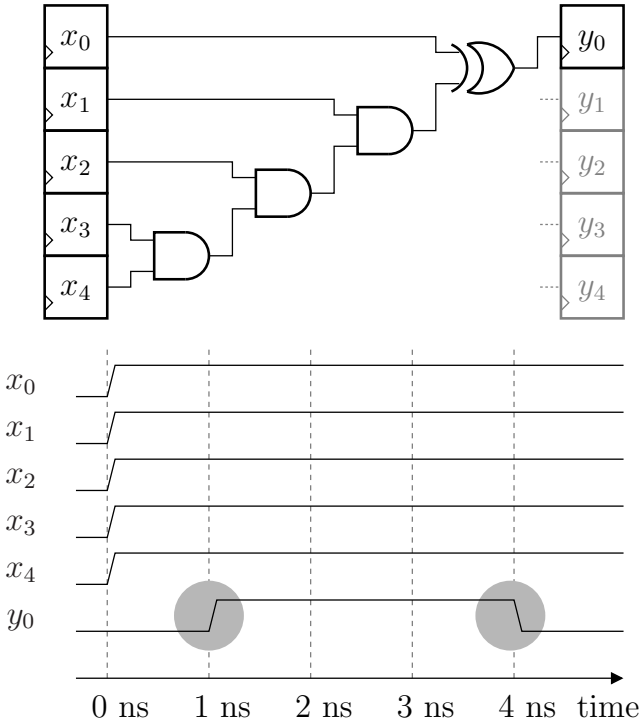


Fig. 7. Example of 4th-order glitch occurring upon 4th-order conjunction $\bigwedge_{i=1}^{i=4} x_i$

Assume now a setup slightly more simple than that of Fig. 7, where there is only one `AND` gate behind the second input of the `XOR` gate. However, we assume such pattern is present twice, once computing $y_0 = x_0 \oplus (x_1 \wedge x_2)$, and

another time computing $y_5 = x_5 \oplus (x_4 \wedge x_3)$. Then, in this case depicted in Fig. 8, the leakage incurred by the glitches at the output of the `XOR` gates would only combine two bits amongst the x_i (namely x_1 & x_2, and x_3 & x_4). Therefore, it suffices for the attacker to conduct a 2nd-order attack on the glitchy traces to succeed a $2 \times 2 = 4$th order attack on the masking scheme. The circuit and the attacker collaborate in the objective of realizing a 4th-order attack: half of the combination is carried out by the circuit ($(x_1 \wedge x_2)$ and $(x_3 \wedge x_4)$), while the other half is left remaining to the attacker. Indeed, by raising the traces to the second power, the attacker obtains a term $(x_1 \wedge x_2) \times (x_3 \wedge x_4)$, which coincides with the leakage condition of Fig. 7, that is $\bigwedge_{i=1}^{i=4} x_i$.

To conclude on the leakage model complexification, we underline that it has a negative impact on two situations:

- on low-entropy masking schemes, where the *individual shares* are not protected at the maximum order (see for instance RSM in Sec. 3.5), and
- on any masking schemes, where *shares interact between themselves* by some combinational logic.

In those two cases, a great care must be taken; tools as that described in [18] can help check the design is secure (or not).

5 Conclusion

Throughout this paper, we have seen how coding and side-channel analysis can benefit one from another, for attack as well as for protection.

This is a nice example of cross fertilization between disciplines, in which Claude Carlet played a decisive role. Thanks to you, Claude!

Acknowledgements

Part of this work has been funded by the ANR CHIST-ERA project SECODE (*Secure Codes to thwart Cyber-physical Attacks*).

References

1. 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.
2. Éric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Joye and Quisquater [27], pages 16–29.

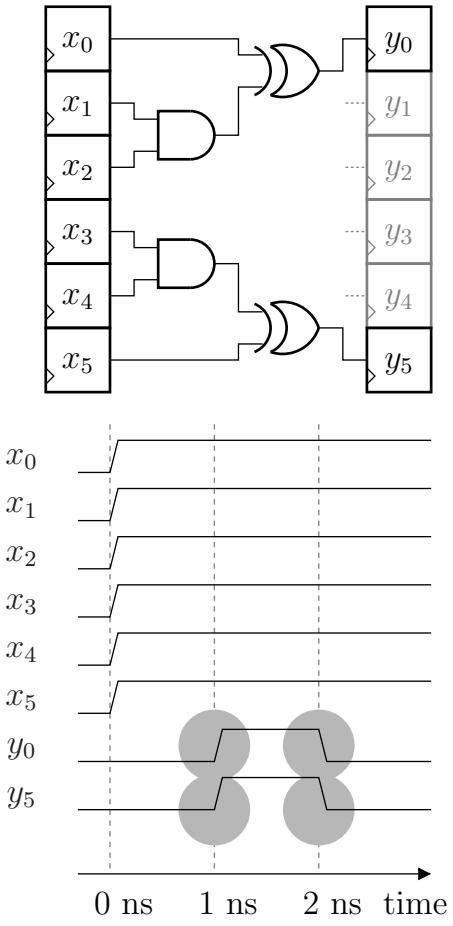


Fig. 8. Example of two 2nd-order glitches occurring upon 2th-order conjunctions $\bigwedge_{i=1}^{i=2} x_i$ and $\bigwedge_{i=3}^{i=4} x_i$

3. Julien Bringer, Claude Carlet, Hervé Chabanne, Sylvain Guilley, and Houssem Maghrebi. Orthogonal Direct Sum Masking – A Smartcard Friendly Computation Paradigm in a Code, with Builtin Protection against Side-Channel and Fault Attacks. In *WISTP*, volume 8501 of *LNCS*, pages 40–56. Springer, June 2014. Heraklion, Greece.
4. 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 [3]).
5. Nicolas Bruneau, Claude Carlet, Sylvain Guilley, Annelie Heuser, Emmanuel Prouff, and Olivier Rioul. Stochastic collision attack. *IEEE Trans. Information Forensics and Security*, 12(9):2090–2104, 2017.

6. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is More - Dimensionality Reduction from a Theoretical Perspective. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 22–41. Springer, 2015.
7. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Optimal Side-Channel Attacks for Multivariate Leakages and Multiple Models, August 20 2016. Santa Barbara, CA, USA. Online reference: http://www.proofs-workshop.org/2016/program.html. To appear in the Journal of Cryptographic Engineering.
8. 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.
9. Claude 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. Preliminary version available at http://www.math.univ-paris13.fr/~carlet/chap-fcts-Bool-corr.pdf.
10. 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.
11. Claude Carlet, Abderrahman Daif, Jean-Luc Danger, Sylvain Guilley, Zakaria Najm, Xuan Thuy Ngo, Thibault Porteboeuf, and Cédric Tavernier. Optimized linear complementary codes implementation for hardware Trojan prevention. In *European Conference on Circuit Theory and Design, ECCTD 2015, Trondheim, Norway, August 24-26, 2015*, pages 1–4. IEEE, 2015.
12. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. In *INDOCRYPT*, volume 7668 of *LNCS*, pages 120–139. Springer, December 9-12 2012. Kolkata, India.
13. 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.
14. Claude Carlet and Sylvain Guilley. Side-Channel Indistinguishability. In *HASP*, pages 9:1–9:8, New York, NY, USA, June 23-24 2013. ACM.
15. Claude Carlet and Sylvain Guilley. Side-Channel Indistinguishability, July 19 2014. On HAL: http://hal.archives-ouvertes.fr/hal-00826618. Extended version of [14] with more results in appendix.
16. 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.
17. Jean-Luc Danger and Sylvain Guilley. Protection des modules de cryptographie contre les attaques en observation d'ordre élevé sur les implémentations à base de masquage, 20 Janvier 2009. Brevet Français FR09/50341, assigné à l'Institut TELECOM.
18. Jean-Luc Danger, Sylvain Guilley, Philippe Nguyen, Robert Nguyen, and Youssef Souissi. Analyzing Security Breaches of Countermeasures Throughout the Refinement Process in Hardware Design Flow. In *DATE*, March 27-31 2017. Lausanne, Switzerland.

19. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In 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*, pages 401–429. Springer, 2015.
20. Yunsi Fei, Qiasi Luo, and A. Adam Ding. A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES*, volume 7428 of *LNCS*, pages 233–250. Springer, 2012.
21. Wieland Fischer and Berndt M. Gammel. Masking at Gate Level in the Presence of Glitches. In *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 187–200. Springer, August 29 – September 1 2005. Edinburgh, UK.
22. Mahadevan Gomathisankaran and Akhilesh Tyagi. Glitch resistant private circuits design using HORNS. In *IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2014, Tampa, FL, USA, July 9-11, 2014*, pages 522–527, 2014.
23. 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.
24. Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 55–74. Springer, 2014.
25. Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private Circuits II: Keeping Secrets in Tamperable Circuits. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, May 28 – June 1 2006. St. Petersburg, Russia.
26. 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.
27. 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*. Springer, 2004.
28. Kuan Jen Lin, Shan Chien Fang, Shih Hsien Yang, and Cheng Chia Lo. Overcoming glitches and dissipation timing skews in design of dpa-resistant cryptographic hardware. In Rudy Lauwereins and Jan Madsen, editors, *2007 Design, Automation and Test in Europe Conference and Exposition, DATE 2007, Nice, France, April 16-20, 2007*, pages 1265–1270. EDA Consortium, San Jose, CA, USA, 2007.
29. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *CT-RSA*, volume 3376 of *LNCS*, pages 351–365. Springer, 2005. San Francisco, CA, USA.
30. Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 157–171. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.
31. Stefan Mangard and Kai Schramm. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *CHES*, volume 4249 of *LNCS*, pages 76–90. Springer, October 10-13 2006. Yokohama, Japan.

32. Amir Moradi and Oliver Mischke. Glitch-free implementation of masking in modern fpgas. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012, San Francisco, CA, USA, June 3-4, 2012*, pages 89–95. IEEE, 2012.
33. Maxime Nassar, Youssef Souissi, Sylvain Guilley, and Jean-Luc Danger. RSM: a Small and Fast Countermeasure for AES, Secure against First- and Second-order Zero-Offset SCAs. In *DATE*, pages 1173–1178. IEEE Computer Society, March 12-16 2012. Dresden, Germany. (TRACK A: "Application Design", TOPIC A5: "Secure Systems").
34. Xuan Thuy Ngo, Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. Linear complementary dual code improvement to strengthen encoded circuit against hardware Trojan horses. In *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2015, Washington, DC, USA, 5-7 May, 2015*, pages 82–87. IEEE, 2015.
35. Xuan Thuy Ngo, Sylvain Guilley, Shivam Bhasin, Jean-Luc Danger, and Zakaria Najm. Encoding the State of Integrated Circuits: A Proactive and Reactive Protection Against Hardware Trojans Horses. In *Proceedings of the 9th Workshop on Embedded Systems Security*, WESS '14, pages 7:1–7:10, New York, NY, USA, 2014. ACM.
36. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptology*, 24(2):292–321, 2011.
37. 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).
38. Emmanuel Prouff and Matthieu Rivain. A Generic Method for Secure SBox Implementation. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007.
39. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
40. Emmanuel Prouff and Thomas Roche. Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *LNCS*, pages 63–78. Springer, 2011.
41. 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.
42. Jason Waddle and David A. Wagner. Towards Efficient Second-Order Power Analysis. In Joye and Quisquater [27], pages 1–15.

A SNR in the Presence of First-Order Masking

Let us consider a first-order masking scheme [1]. By design, a first-order side-channel attack fails. However, a second-order side-channel attack, combining two samples, can succeed. The setup is the following: the leakage is:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 Y_1^\star \\ \alpha_2 Y_2^\star \end{pmatrix} + \begin{pmatrix} N_1 \\ N_2 \end{pmatrix},$$

where:

- $N_1 \sim \mathcal{N}(0,\sigma_1^2)$ and $N_2 \sim \mathcal{N}(0,\sigma_2^2)$ are two independent noise sources,
- α_1 and α_2 are the amount of leakage,

– $Y_1^\star$ and $Y_2^\star$ are leakage functions (assumed normalized, that is $\mathbb{E}(Y_i^\star)=0$ and $\mathsf{Var}(Y_i^\star)=1$, for $i\in\{1,2\}$).

In the Boolean masking where the attacker target the pair (mask, masked substitution box S), the leakage model is:

– $Y_1=\frac{2}{\sqrt{n}}\left(w_H(S(T\oplus k)\oplus M)-\frac{n}{2}\right)=-\frac{1}{\sqrt{n}}\sum_{b=1}^{n}(-1)^{S_b(T\oplus k)\oplus M_b}$ and
– $Y_2=\frac{2}{\sqrt{n}}\left(w_H(M)-\frac{n}{2}\right)=-\frac{1}{\sqrt{n}}\sum_{b=1}^{n}(-1)^{M_b}$.

The notation M_b means bit $b\in\{1,...,n\}$ in bitvector $M\in\mathbb{F}_2^n$.

As the masking is first-order perfect, we indeed have that $\mathbb{E}(Y_i|T=t)$ does not depend on the key, for each share $i\in\{1,2\}$. However, the attacker is inclined to *combine* the two leakages by a centered product, since the expectation of this combination $Y_c=Y_1Y_2$ depends on the key, despite the masking with the uniform $M\sim\mathcal{U}(\mathbb{F}_2^n)$. Precisely, let $t\in\mathbb{F}_2^n$ one realization of T. We have that:

$$\begin{aligned}\mathbb{E}(Y_c|T=t)&=\frac{1}{2^n}\sum_{m\in\mathbb{F}_2^n}\frac{1}{n}\sum_{b,b'}(-1)^{S_b(T\oplus k)\oplus m_b\oplus m_{b'}}\\&=\frac{1}{n2^n}\sum_{m\in\mathbb{F}_2^n}\sum_{b}(-1)^{S_b(T\oplus k)}\qquad\text{(because } m \text{ is uniform on } \mathbb{F}_2^n)\\&=-\frac{1}{2\sqrt{n}}\left(w_H(S(T\oplus k))-\frac{n}{2}\right),\end{aligned}\tag{12}$$

which happens to be proportional to the leakage model of the substitution box when the masking is disabled ($M=0$). Indeed, one can derive from Eqn. (12) that:

$$\mathbb{E}(Y_c|T=t)=-\frac{1}{2\sqrt{n}}\mathbb{E}(Y_1|T=t,M=0).$$

The second-order attack thus consists in applying the regular correlation power analysis (CPA [2]):

– targeting $X_c=X_1X_2$ instead of X_1 or X_2,
– using as leakage model $\mathbb{E}(Y_c|T)$, where we recall that $Y_c=Y_1Y_2$ [39].

Thus, the new leakage to analyse is:

$$\begin{aligned}X_c=X_1X_2&=(\alpha_1Y_1^\star+N_1)(\alpha_2Y_2^\star+N_2)\\&=\underbrace{\alpha_1\alpha_2Y_1^\star Y_2^\star}_{\text{signal}}+\underbrace{\alpha_1Y_1^\star N_2+\alpha_2Y_2^\star N_1+N_1N_2}_{\text{noise}}.\end{aligned}$$

Indeed, the term $Y_1^\star Y_2^\star$ conditionally to the known plaintext T depends on the key (recall Eqn. (12)), whereas the other terms $\alpha_1Y_1^\star N_2+\alpha_2Y_2^\star N_1+N_1N_2$ do not.

Therefore, the SNR in the case of the second-order attack is:

$$\mathrm{SNR}(2o)=\frac{\mathsf{Var}(\alpha_1\alpha_2 Y_1^\star Y_2^\star)}{\mathsf{Var}(\alpha_1 Y_1^\star N_2+\alpha_2 Y_2^\star N_1+N_1N_2)}. \tag{13}$$

Proposition 11. *The SNR in the case of the second-order attack is:*

$$SNR(2o)=\frac{SNR_1\cdot SNR_2}{1+SNR_1+SNR_2},$$

where $SNR_i=\alpha_i^2/\sigma_i^2$ *for* $i\in\{1,2\}$.

Proof. We have:

$$\begin{aligned}
\mathbb{E}_{T,M}(Y_1^\star Y_2^\star)&=\frac{1}{2^{2n}}\sum_{t\in\mathbb{F}_2^n,m\in\mathbb{F}_2^n} Y_1^\star Y_2^\star\\
&=\frac{1}{2^{2n}}\left(\frac{2}{\sqrt{n}}\right)^2\sum_m\left(w_H(m)-\frac{n}{2}\right)\sum_t\left(w_H(S(t\oplus k^\star)\oplus m)-\frac{n}{2}\right)\\
&=\frac{1}{2^{2n}}\left(\frac{2}{\sqrt{n}}\right)^2\sum_m\left(w_H(m)-\frac{n}{2}\right)\sum_z\left(w_H(z)-\frac{n}{2}\right) \qquad (14)\\
&=0\times 0=0.
\end{aligned}$$

At line (14), we used the fact that S is a bijection of $\mathbb{F}_2^n$ (as is `SubBytes` in AES [37]).

Besides, we also have:

$$\begin{aligned}
\mathbb{E}_{T,M}\left((Y_1^\star Y_2^\star)^2\right)&=\frac{1}{2^{2n}}\sum_{t\in\mathbb{F}_2^n,m\in\mathbb{F}_2^n}(Y_1^\star)^2(Y_2^\star)^2\\
&=\frac{1}{2^{2n}}\left(\frac{2}{\sqrt{n}}\right)^4\sum_m\left(w_H(m)-\frac{n}{2}\right)^2\sum_t\left(w_H(S(t\oplus k^\star)\oplus m)-\frac{n}{2}\right)^2\\
&=\frac{1}{2^{2n}}\left(\frac{2}{\sqrt{n}}\right)^4\sum_m\left(w_H(m)-\frac{n}{2}\right)^2\sum_z\left(w_H(z)-\frac{n}{2}\right)^2 \qquad (15)\\
&=1\times 1=1 \qquad \text{(as per the normalization of } Y_1^\star \text{ and } Y_2^\star).
\end{aligned}$$

Therefore, the variance of the signal is equal to $\alpha_1^2\alpha_2^2$.

Regarding the noise part, we have:

$$\mathbb{E}(\alpha_1 Y_1^\star N_2+\alpha_2 Y_2^\star N_1+N_1N_2)=0,$$

by independence between N_1, N_2 and $Y_i^\star$ for $i\in\{1,2\}$. We also have:

$$\begin{aligned}\mathsf{Var}(\alpha_1 Y_1^\star N_2+\alpha_2 Y_2^\star N_1+N_1N_2)&=\mathbb{E}\big((\alpha_1 Y_1^\star N_2+\alpha_2 Y_2^\star N_1+N_1N_2)^2\big)-0\\&=\alpha_1^2\sigma_2^2+\alpha_2^2\sigma_1^2+\sigma_1^2\sigma_2^2.\end{aligned}$$

As a result, we have:

$$\mathrm{SNR}(2\mathrm{o})=\frac{\alpha_1^2\alpha_2^2}{\alpha_1^2\sigma_2^2+\alpha_2^2\sigma_1^2+\sigma_1^2\sigma_2^2}=\frac{\mathrm{SNR}_1\cdot\mathrm{SNR}_2}{1+\mathrm{SNR}_1+\mathrm{SNR}_2}.$$

□

Corollary 12 (Limit of SNR(2o) in the presence of large noise). *When the noise is large, that is $SNR_i\ll 1$ for $i\in\{1,2\}$, then*

$$SNR(2o)\approx SNR_1\cdot SNR_2\approx SNR^2 \qquad (\text{if } SNR_1\approx SNR_2=SNR). \tag{16}$$

Proof. Immediate first-order simplification of SNR(2o) as given in Proposition 11. □