Masking against Side-Channel Attacks: A Formal Security Proof

Emmanuel Prouff[1] and Matthieu Rivain[2]

[1] ANSSI
emmanuel.prouff@ssi.gouv.fr
[2] CryptoExperts
matthieu.rivain@cryptoexperts.com

Abstract. Masking is a well-known countermeasure to protect block cipher implementations against side-channel attacks. The principle is to randomly split every sensitive intermediate variable occurring in the computation into $d + 1$ shares, where d is called the masking order and plays the role of a security parameter. Although widely used in practice, masking is often considered as an empirical solution and its effectiveness is rarely proved. In this paper, we provide a formal security proof for masked implementations of block ciphers. Specifically, we prove that the information gained by observing the leakage from one execution can be made negligible (in the masking order). To obtain this bound, we assume that every elementary calculation in the implementation leaks a noisy function of its input, where the amount of noise can be chosen by the designer (yet linearly bounded). We further assume the existence of a leak-free component that can refresh the masks of shared variables. Our work can be viewed as an extension of the seminal work of Chari *et al.* published at CRYPTO in 1999 on the soundness of combining masking with noise to thwart side-channel attacks.

1 Introduction

Side-channel analysis is a class of cryptanalytic attacks that exploit the physical environment of a cryptosystem to recover some *leakage* about its secrets. It is often more efficient than a cryptanalysis in the so-called *black-box model* in which no leakage occurs. Two attack categories are usually considered: the *bounded side-channel attacks* and the *continuous side-channel attacks*. In a bounded side-channel attack [9], the total amount of leakage accessible to the adversary is bounded. In a continuous side-channel attack, the adversary gets some information at each invocation of the cryptosystem, and the amount of leakage can thus be arbitrarily large. Attacks where the adversary measures the running-time [24], the power consumption [25] or the electromagnetic radiations [15] of a cryptographic implementation fall into this category.

Continuous side-channel attacks have proved to be especially effective to break unprotected cryptographic implementations. And although many ingenious countermeasures have been developed during past years, very few of them

T. Johansson and P. Nguyen (Eds.): EUROCRYPT 2013, LNCS 7881, pp. 142–159, 2013.

gave rise to concrete security guaranties. This has raised the need for models and methods to formally prove the security of cryptographic implementations against continuous side-channel attacks. A pioneering work in this direction is the *physically observable cryptography* framework introduced by Micali and Reyzin in [29] which puts forward a general theory of side-channel attacks. In particular, they formalize the assumptions that a cryptographic device can at least keep some secrets and that *only computation leaks information* [29]. A few years later, Dziembowski and Pietrzak introduced the *leakage resilient cryptography* model [12], which is a generalization of the *bounded retrieval model* [9] where every step of the computation leaks information on the processed part of the device state trough a function whose range is bounded (*i.e.* taking values in $\{0,1\}^\lambda$ for some parameter λ). Under this assumption, the authors were able to design secure pseudo-random number generators [12,32]. Further leakage resilient cryptographic primitives were then constructed under the same – or sometimes stronger – assumptions (see for instance [10, 13, 23, 42]). The issue of designing generic compilers that can transform any cryptographic algorithm into a leakage resilient implementation was also recently addressed [11, 17, 18, 22]. These works are nice proofs of concept but the proposed constructions are not suited for practical implementation, especially in constrained environments such as embedded systems. Moreover the practical meaning of the underlying bounded range leakage model with respect to power or electromagnetic side channels is questionable [40].

A more practical and traditionally used approach to secure implementations against side-channel attacks is *secret sharing* [1, 37] also called *masking* in this context. The idea is to randomly split a secret into several shares such that the adversary needs all of them to reconstruct the secret. Masking was soon identified as a sound countermeasure when side-channel attacks appeared in the literature [4, 19]. Since then, many works have been published to address the practical implementation and/or the security of masking for various ciphers. However a formal security proof is still missing at this day. Our goal is to fill this gap.

1.1 Related Works

Soundness of Masking. In [4], Chari *et al.* conduct a formal study of masking in the presence of noisy leakage. More precisely, the authors investigate the soundness of randomly sharing a secret bit into d shares when the adversary has only access to a noisy version of those shares, the noise having a Gaussian distribution with variance σ^2. They prove that the number of observations required to distinguish, with success probability α, the leakage distribution when the secret equals 0 from the leakage distribution when the secret equals 1 is lower bounded by $\sigma^{d+4\log\alpha/\log\sigma}$. This bound is frequently recalled to argue for the soundness of masking when combined with noise. Despite its great interest and impact, Chari *et al.*'s analysis has an important limitation: no solution is provided to apply masking to the whole implementation of a cryptographic algorithm and to analyze the global security of such an implementation.

Private Circuits and Extensions. In [21], Ishai *et al.* show that any circuit with n logical gates can be transformed into a circuit of size $O(nd^2)$ which is secure against *probing attacks* spying up to d wires. The main contribution of [21] is a method to compute an AND gate between shared inputs while ensuring the security against a d-probing adversary. However, a security proof against probing attacks does not give full satisfaction since it does not encompass an adversary exploiting the entire leakage produced during the processing.

In [14], Faust *et al.* propose an extension of Ishai *et al.*'s scheme. Their scheme requires a leak-free hardware component but it is provably secure under two different and more general leakage models. In the first model, the leakage at each cycle is any function of the circuit internal state (*i.e.* the logical values carried by all the wires) which is computationally bounded: it is assumed to be in the complexity class AC^0 (*i.e.* it must be computable by a circuit of constant depth). In the second model, the leakage reveals the value of each internal state bit, flipped with a probability $p < 1/2$ (*i.e.* xor-ed with a p-Bernoulli noise). In a recent paper [35], Rothblum further showed how to remove the requirement of leak-free component in the AC^0 leakage model. These works achieve an important progress towards provable security against side-channel attacks since they show that masking can bring security even in the presence of a global leakage on the entire state. However, the practicability of the considered models is questionable. In particular it is unclear whether the AC^0 leakage or the full leakage with Benouilli noise really fit the physical reality of power and/or electromagnetic leakages.

Masking Schemes. On the other hand, several works have shown how to apply masking to various algorithms in practice. They however often omit to prove the security of the resulting implementations. The first masking scheme was proposed by Goubin and Patarin in [19] for the DES cipher. Further schemes were subsequently published in which masking is applied at hardware or software level at the cost of different area-time-memory trade-offs (see for instance [2, 28, 30]). Originally, most of these schemes deal with *first-order masking* which splits each sensitive variable in two shares (a mask and a masked variable). Then *higher-order masking schemes* were defined to get security against side-channel attacks exploiting the leakage of several, say d, intermediate computations [3, 16, 33, 34]. The purpose of these schemes is analogous to the d-probing secure circuit of Ishai *et al.* : the computation is performed such that any d intermediate variables occurring in the algorithm reveal no sensitive information. Most of these schemes are actually based on the method of Ishai *et al.* to securely process a multiplication between two shared variables. Consequently, they suffer the same limitation as Ishai *et al.* 's scheme: they only thwart a limited adversary that does not exploit the overall leakage.

1.2 Our Contribution

In this paper we formally prove the security of masked implementations of block ciphers in the *only computation leaks information* model [29]. In this model,

every *step* of the processing reveals a leakage function of the touched part of the device state. This function is chosen adaptively by the adversary in some predetermined class. For our security proof, we split the computation into several *elementary calculations* (in practice, a sequence of few CPU instructions) that each accesses a subpart x of the device state and leaks a function of x. Starting from the observation that masking is sound when combined with noise [4] and that many practical solutions exist to amplify leakage noise (see for instance [6–8, 20, 27, 39, 41]), we assume the leakage functions to be *noisy*. The noisy feature of a leakage function f is captured by assuming that an observation of $f(x)$ only implies a *bounded bias* in the probability distribution of x. Namely the statistical distance between the distributions $\mathrm{P}[x]$ and $\mathrm{P}[x|f(x)]$ is bounded. We further assume that this bound depends on a noise parameter ω that can be chosen by the designer according to the required security level. Our security proof has a natural limitation which is the requirement of a leak-free component, an elementary calculation refreshing the masks of a shared variable. Under these assumptions, we achieve an information theoretic security proof: we show that the mutual information between the cipher input (plaintext and secret key) and the overall leakage on the block cipher processing is upper bounded by $\omega^{-(d+1)}$, where d is the masking order.

This bound can be seen as an extension of the seminal work of Chari *et al.* [4] as it is derived from the combination of masking with noise. We extend their analysis in two ways. First we consider a more general leakage model which no longer requires particular assumptions (single-bit target variable, Gaussian noise). More importantly, we provide a security bound for a full masked block cipher implementation whereas Chari *et al.* analysis focus on leaking shares independently of any computation. Our work can also be viewed as an alternative to previous works on program or circuit compilers with formal security proofs against side-channel attacks [11,14,17,18,22,35]. Whereas the practical meaning of the leakage models considered in these works is questionable, our leakage model aims to be compliant with practical investigations about side-channel leakage (see for instance [27,31,36,38]).

2 Preliminaries

Calligraphic letters, like $\mathcal{X}$, are used to denote finite sets. The corresponding large letter X is used to denote a random variable over $\mathcal{X}$, while the lower-case letter x denotes a particular element from $\mathcal{X}$. To every discrete random variable X, one associates a probability mass function P_X defined by $\mathrm{P}_X(x) = \mathrm{P}[X = x]$. Let Y be a random variable defined over some set $\mathcal{Y}$ and let $y \in \mathcal{Y}$. Then $(X|Y = y)$ denotes the random variable with probability mass function $x \mapsto \mathrm{P}[X = x|Y = y]$. The *entropy* (or *Shannon entropy*) $\mathrm{H}[X]$ of a discrete random variable X is defined by $\mathrm{H}[X] = -\sum_{x\in\mathcal{X}} \mathrm{P}_X(x) \log_2(\mathrm{P}_X(x))$. The *mutual information* between two random variables X and Y is then defined by $\mathrm{I}(X;Y) = \mathrm{H}[X] - \mathrm{H}[X|Y]$, where $\mathrm{H}[X|Y]$ is called the *conditional entropy of X given Y* and is defined by $\mathrm{H}[X|Y] = \sum_{y\in\mathcal{Y}} \mathrm{P}_Y(y)\ \mathrm{H}[(X|Y = y)]$. The *statistical distance* between

two random variables X_0 and X_1 is denoted by $\mathrm{d}(X_0; X_1)$ and is defined by $\mathrm{d}(X_0; X_1) = \|\mathrm{P}_{X_0} - \mathrm{P}_{X_1}\|$ where $\|\cdot\|$ denotes the Euclidean norm and P_{X_i} denotes the vector $(\mathrm{P}_{X_i}(x))_{x\in\mathcal{X}}$. We recall that for any N-dimensional vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$ we have

$$\|\boldsymbol{y}\| \leq L_1(\boldsymbol{y}) \leq \sqrt{N}\|\boldsymbol{y}\| \ , \tag{1}$$

where $L_1(\boldsymbol{y})$ denotes the *Manhattan norm* $\sum_i |y_i|$.

We now introduce the notion of *bias* that is extensively used in our security proof.

Definition 1. *Let X and Y be two random variables. The* bias of X given $Y = y$*, denoted $\beta(X|Y = y)$, is defined as*

$$\beta(X|Y = y) = \mathrm{d}\big(X; (X|Y = y)\big) \ .$$

The bias of X given Y*, $\beta(X|Y)$, is then defined as the expected bias of X given $Y = y$ over $\mathcal{Y}$, that is*

$$\beta(X|Y) = \sum_{y\in\mathcal{Y}} \mathrm{P}_Y(y)\beta(X|Y = y) \ .$$

The bias of X given Y is an information metric between X and Y. If X and Y are independent then $\beta(X|Y)$ equals zero. Moreover, as shown in the next proposition, it is related to the mutual information between X and Y (the proof is provided in the full version).

Proposition 1. *Let X and Y be two random variables, with X uniformly distributed over a set $\mathcal{X}$ of cardinality N. The mutual information between X and Y satisfies* $\mathrm{I}(X; Y) \leq \frac{N}{\ln 2}\beta(X|Y)$.

3 Model of Leaking Computation

We describe hereafter our model of leaking computation.

An algorithm is modelled by a sequence of *elementary calculations* $(\mathsf{C}_i)_i$ that are Turing machines augmented with a common random access memory called *the state*. Each elementary calculation reads its input and writes its output on the state. When an elementary calculation C_i is invoked, its input is written from the state to its input tape, then C_i is executed, afterwards its output is written back to the state.

A *physical implementation* of an algorithm is modelled by a sequence of *physical elementary calculations*. A physical elementary calculation $(\mathsf{C}_i, f_i)_i$ is composed of an elementary calculation C_i and a *leakage function* f_i. A leakage function is defined as a function that takes two parameters: the value held by the accessed part of the state and a random string long enough to model the leakage noise. Let $\mathcal{I} = (\mathsf{C}_i, f_i)_i$ be a physical implementation of an algorithm $\mathcal{A}$ as defined above. Each execution of $\mathcal{I}$ leaks the values $\big(f_i(x_i, r_i)\big)_i$ where x_i denotes

the input of C_i and r_i is a fresh random string. In particular all the r_i involved in successive executions of $\mathcal{I}$ are uniformly and independently drawn.

For the sake of simplicity, we shall omit the random string parameter, which leads to the notation $f_i(x)$ where x is the accessed value. Note that $f_i(x)$ is not the result of a function but it can be seen as the output of a probabilistic Turing machine. In particular, $f_i(x)$ can take several values with a given probability distribution, and is therefore considered as a random variable in the following. Let $\mathcal{X}$ be the definition set of the accessed part of the state. We shall then say that f_i is defined over $\mathcal{X}$ (or equivalently that $\mathcal{X}$ is the domain of f_i).

For our security proof, we will consider special classes of leakage functions that we shall call *noisy leakage functions*. Let f be a leakage function defined over some set $\mathcal{X}$ and let X denote a uniform random variable over $\mathcal{X}$. The noisy property of f is captured by assuming that the bias introduced in the distribution of X given the leakage $f(X)$ is bounded.[1] For any positive real number ε, we define the class of *noisy leakage functions w.r.t. bias* ε, and we denote by $\mathcal{N}(\varepsilon)$, the set of noisy functions such that $\beta(X|f(X)) \leq \varepsilon$. In this paper, we shall assume that the designer can constrain as willing the set of noisy leakage functions related to any elementary calculation by linearly increasing the amount of noise in the leakage. More precisely, we assume that the designer controls a noise parameter ω such that an elementary calculation C can yield a physical elementary calculation (C, f) with $f \in \mathcal{N}(1/\omega)$, where ω is linear in the security parameter.

3.1 Discussion

Our model can be seen as a specialization of the physically observable cryptography framework [29] with leakage functions belonging to the class of noisy functions as defined above (the similarities between our model and this framework are discussed in the full version). Our model is also comparable to the leakage resilient cryptography model [12] with two important differences relating to the computation granularity and the class of leakage functions.

Computation Granularity. As nicely explained in [17], a computation in the *only computation leaks* model is divided into several *sub-computations* and a leakage function applies to the input of each sub-computation. In [12,32] the authors construct stream ciphers that output an unbounded number of key-stream blocks from a secret key block. A sub-computation is then naturally identified as the computation of one block. In contrast, we consider a finer granularity: the computation of one block (*i.e.* a single block cipher computation) is divided into several elementary calculations, each one leaking a function of its input. In other words, [12, 32] address the issue of constructing secure protocols from

[1] For the above definition of noisy leakage functions to be sound, we need to precise the distribution of X while bounding $\beta(X|f(X))$, and a natural choice is the uniform distribution. Of course, this does not constrain the leakage function to only apply on uniformly distributed inputs.

a cryptographic primitive with limited leakage whereas we address the issue of constructing secure cryptographic primitives from elementary calculations with noisy leakages.

Class of Leakage Functions. The most noticeable difference between our work and the previous leakage-resilient constructions resides in the considered class of leakage functions. Most previous works consider the class of bounded-range functions taking values in $\{0,1\}^\lambda$ for some parameter λ [10–13,17,18,22,23,32]. This hypothesis is conservative in terms of security since it encompasses leakage functions with complex structures. However its practical meaning is unclear with regard to power and electromagnetic side channels for which the leakage is usually substantially larger than the secret state (but hopefully does not contain its overall entropy).

In contrast, several techniques are known to add some noise in the side-channel leakage in practice [6–8,20,26,27,39,41]. By noise addition we mean that the relevant signal in the leakage is lowered compared to random variations, although this may not literally result from noise addition (the terminology of *hiding* is sometimes used). This motivates our definition of noisy leakage functions. Note that in practice, power and electromagnetic leakages can realistically be modeled by a multivariate deterministic function g of the processed data with an additional multivariate Gaussian noise N [5,27,36,38]. The class $\mathcal{N}(\varepsilon)$ corresponding to such a leakage function can then be determined from the description of g and N (see the full version for further details).

4 Masked Implementation of Block Ciphers

Several schemes have been proposed to securely process a block cipher composed of linear layers and non-linear s-boxes. In this paper, we prove the security of the scheme described in [3] with a secure multiplication processing close to those of [14,21], and with additional mask-refreshing computations. Before presenting the masking scheme, we start by formalizing the considered block cipher processing.

4.1 Block Cipher Processing

A block cipher is a cryptographic algorithm which, from a secret key, transforms a plaintext block into a ciphertext block. In this paper, we focus on block ciphers designed as a succession of linear functions and substitution boxes (s-boxes). S-boxes are nonlinear functions from $\{0,1\}^n$ to $\{0,1\}^m$ with $m \leq n$ and n small (typically $n \in \{4,6,8\}$). We assume that the addition law is the bitwise addition, denoted $\oplus$, and that the s-boxes are *balanced*; namely every $y \in \{0,1\}^m$ has the same number of preimages in $\{0,1\}^n$ under the s-box. In the computation model introduced in Section 3, the processing of such a block cipher is represented as a sequence of elementary calculations, each of them implementing either a linear function or an s-box. The input of this processing is a pair composed of the plaintext and the secret key and the output is the corresponding ciphertext.

Uniformity Property. We shall assume that the block cipher satisfies the following uniformity property: a uniform distribution of the cipher input (plaintext-key pair) induces a uniform distribution of the input of every of its elementary calculation. The uniformity property is satisfied by classical block cipher designs such as DES and AES.

4.2 Securing the Block Cipher Processing

We start with the following definition that formalizes the notion of d^{th}*-order encoding.*

Definition 2 (d^{th}-order encoding). *Let d be a positive integer and let I denote the integer interval $\{0, 1, \ldots, d\}$. The* d^{th}-order encoding of $x \in \mathcal{X}$ *is a $(d+1)$-tuple $(x_i)_{i\in I}$ satisfying $\bigoplus_i x_i = x$ and such that $(x_i)_{i\in I\setminus\{i_0\}}$ is uniformly distributed over $\mathcal{X}^d$ for every $i_0 \in I$.*

Masking a block cipher implementation consists in choosing a security parameter d (called *masking order*) and in performing the computation on a d^{th}-order encoding of the state. Namely, the plaintext and the secret key are split into $d+1$ shares. Then, a scheme is defined that specifies how each elementary calculation is replaced by a sequence of new elementary calculations operating only on the shares. At the end, the new sequence must return an encoding of the ciphertext (from which the actual ciphertext can be straightforwardly recovered).

According to the block cipher model of Section 4.1, we describe hereafter how to process a linear function and an s-box computation on shared inputs as proposed in [3]. We first introduce the mask-refreshing component used at several steps in the masking scheme and which is assumed to be *leak-free* in our security proof.

Mask Refreshing. Our scheme requires a special kind of elementary calculation to refresh the masks of an encoding *without leaking information.* This mask-refreshing oracle is denoted by $\mathcal{O}^{\$}$. From an encoding of any value x, it computes a new encoding of x with fresh random values. For the sake of simplicity, we assume that when invoked the input and output of our leak-free component do not leak. However this assumption could be relaxed since the input comes from a previous elementary calculation and the output is used in the subsequent elementary calculations (otherwise its masks would not need to be refreshed), therefore they both leak at some point in the computation.

Secure Linear Function Processing. To secure the processing of any linear function g, the following process is applied:

1. For every $i \in \{0, 1, \ldots, d\}$, process $z_i \leftarrow g(x_i)$.
2. Output $(z_i)_i \leftarrow \mathcal{O}^{\$}\big((z_i)_i\big)$.
3. If the encoding $(x_i)_i$ is used subsequently in the block cipher processing, process $(x_i)_i \leftarrow \mathcal{O}^{\$}\big((x_i)_i\big)$.

Secure S-Box Processing. Let S be an s-box with input dimension n and output dimension $m \leq n$. Then S can be represented by a polynomial function $x \in \mathbb{F}_{2^n} \mapsto \bigoplus_{i=0}^{2^n-1} \alpha_i x^i$ where the α_i are constant coefficients in $\mathbb{F}_{2^n}$. The α_i can be computed from the s-box look-up table by applying Lagrange's Theorem.[2] Thanks to this representation, the s-box calculation can be done by processing four kinds of elementary calculations over $\mathbb{F}_{2^n}$: addition, multiplication by a constant, square, and regular multiplication (*i.e.* of two different elements). The three former kinds of calculations are linear (or affine including the addition by a non-zero constant) and their processing can hence be done exactly as for linear transformations. When the calculation is a regular multiplication, the following scheme is applied.

Secure Regular Multiplication Processing. To secure the processing of a regular multiplication, we use a method similar to that of [14, 21]. The input is a pair of encodings of the multiplication operands a and b. The definition of the sequence of elementary calculations computing the encoding of $a \times b$ is deduced from the following relation: $a \times b = \left(\bigoplus_i a_i\right) \times \left(\bigoplus_i b_i\right) = \bigoplus_{i,j} a_i \times b_j$. It is described hereafter:

1. For every $(i,j) \in \{0,1,\ldots,d\}^2$, process $v_{i,j} \leftarrow a_i \times b_j$.
2. For every $j \in \{0,1,\ldots,d\}$, process $(v_{0j}, v_{1j}, \ldots, v_{dj}) \leftarrow \mathcal{O}^{\$}(v_{0j}, v_{1j}, \ldots, v_{dj})$.
3. Process $(v_{00}, v_{01}, \ldots, v_{0d}) \leftarrow \mathcal{O}^{\$}(v_{00}, v_{01}, \ldots, v_{0d})$.
4. For every $i \in \{0,1,\ldots,d\}$, process $z_i \leftarrow \bigoplus_{j=0}^{d} v_{i,j}$
5. Output $(z_i)_i \leftarrow \mathcal{O}^{\$}\big((z_i)_i\big)$.
6. If one of the input encodings $(a)_i$ and $(b_i)_i$ is involved in a subsequent elementary calculation, then process $(a_i)_i \leftarrow \mathcal{O}^{\$}\big((a_i)_i\big)$ and/or $(b_i)_i \leftarrow \mathcal{O}^{\$}\big((b_i)_i\big)$.

Note that Steps 2 and 3 intend to refresh the masks between the subsequences of elementary calculations in Steps 1 and 4. Namely, these steps render the probability distributions $\big((a_i)_i, (b_j)_j | (a,b)\big)$ (in Step 1) and $\big((v_{i,j})_{i,j} | (a,b)\big)$ (in Step 4) mutually independent. Note that Step 3 is mandatory to this aim as Step 2 only makes $\big((v_{i,j})_i | (a,b)\big)$ independent of $\big((a_i)_i, (b_j)_j | (a,b)\big)$ for every column j separately. From this point, Step 3 ensures the global independence.

For our security proof, we shall consider each sum $z_i \leftarrow \bigoplus_{j=0}^{d} v_{i,j}$ in Step 4 as d successive elementary calculations $t_{i,j} \leftarrow t_{i,j-1} \oplus v_{i,j}$ for $1 \leq j \leq d$ with $t_{i,0} = v_{i,0}$ and giving $z_i = t_{i,d}$.

It is clear from the above description that securing a regular multiplication is expensive compared to securing a linear function. The complexity of a secure multiplication is quadratic in d whereas the complexity of a secure linear function is linear in d. Moreover the secure multiplication requires several calls to the mask refreshing oracle. For efficiency purpose, one should hence try to minimize the number of multiplications in the polynomial representation of the s-box. We

[2] When m is strictly lower than n, the m-bit outputs can be embedded in $\mathbb{F}_{2^n}$ by padding them to n-bit outputs (*e.g.* by setting most significant bits to 0). The padding is then removed after the polynomial evaluation.

refer to [3] where efficient heuristics of polynomial evaluation are proposed with respect to this criterion.

5 Main Theorems

It is well-known that any subset of at most d shares X_i gives no information on a secret X encoded at order d, while the whole $d+1$ shares enable to fully reconstruct the secret. In [4], Chari *et al.* consider an adversary which has access to the noisy version of all the shares, i.e. $X_i + N_i$ where N_i is a Gaussian noise with standard deviation σ. They restrict themselves to the case of a single secret bit and show that distinguishing the distribution of the noisy shares associated to a secret bit at 0 and that associated to a secret bit at 1 with a probability $\alpha \in [0, 1)$ requires at least $\sigma^{d+4 \log \alpha / \log \sigma}$ samples. In other words, the required number of leakage observations increases exponentially with the masking order, the underlying base being the noise standard deviation.

Chari *et al.* 's result demonstrates the soundness of using masking under a practically relevant leakage model. However, they only focus on a static leakage of the shares and not on the leakage occurring while computing on the shares. In this paper, we fill this gap by providing security bounds for masked implementations that process shared variables. As explained in Section 4, a block cipher may be decomposed into linear operations and multiplications in a finite field. We derive hereafter upper bounds on the amount of sensitive information leakage for both operations when protected by masking. The proofs of the corresponding theorems are given in the full version. Then in Section 6, we derive an upper bound on the information leakage for the full masked implementation of the cipher.

5.1 Sequential Processing of the Shares

Our first context deals with the case where the shares are processed sequentially *e.g.* by applying the same linear function to them. For such a processing, we provide hereafter an upper bound on the bias of the secret value distribution given noisy leakages on its shares.

Theorem 1. *Let X be a uniform random variable over some set $\mathcal{X}$ of cardinality N, let d be a positive integer and let $(X_i)_i$ be a d^{th}-order encoding of X. Let $\varepsilon \in [0, 1)$ and let f_0, f_1, ..., f_d be noisy leakage functions defined over $\mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. We have:*

$$\beta\big(X \big| f_0(X_0), f_1(X_1), \ldots, f_d(X_d)\big) \leq N^{\frac{d}{2}} \varepsilon^{d+1} \ .$$

Theorem 1 shows that the bias of X given the leakages on its shares decreases exponentially with the order d, provided that the initial bias is sufficiently low, namely lower than $\frac{1}{\sqrt{N}}$. Seeing the *amount of noise* as the inverse of the bias, we hence obtained an upper-bound that decreases exponentially with the encoding order, the base of the exponent being the amount of noise. This result is in

accordance with [4] while being more general since it encompasses any (univariate or multivariate) leakage distribution and any data dimension.

In some contexts, the shared variable is not uniformly distributed, but it is a deterministic function of a uniform secret variable. This case is addressed in the following corollary of Theorem 1.

Corollary 1. *Let X be a uniform random variable over some set $\mathcal{X}$ of cardinality N and let g be a deterministic function from $\mathcal{X}$ to $\mathcal{X}$. Let d be a positive integer and let $(G_i)_i$ be a d^{th}-order encoding of $g(X)$. Let $\varepsilon \in [0,1)$ and let f_0, f_1, ..., f_d be noisy leakage functions defined over $\mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. We have:*

$$\beta\big(X\big|f_0(G_0), f_1(G_1), \ldots, f_d(G_d)\big) \leq N^{\frac{d+3}{2}} \varepsilon^{d+1} \ .$$

5.2 Multiplication of the Shares

The previous theorem deals with a situation where all the shares leak separately which matches the context of the secure linear functions processing. However it is not sufficient alone to deduce an upper bound for the secure multiplication processing given in Section 4. In the latter case, the secure multiplication of two variables A and B from their respective encodings $(A_i)_i$ and $(B_j)_j$ requires to compute the cross-terms $A_i \times B_j$. Hence each share A_i of A appears in $d+1$ different multiplications, one per share B_j, and *vice versa*. Our strategy is first to bound the bias on each share A_i and B_j given the multiple leakages on each share. We can then bound the bias of A and B using a similar approach as in Theorem 1, and we finally derive a bound for the bias of the pair (A, B).

We give hereafter our result for the bias given multiple leakages, and then provide our result for the bias given the cross-term leakages.

Bias Given Multiple Leakages. The next theorem deals with the case of repeated leakages on a variable X. We will then apply it in the secure multiplication context.

Theorem 2. *Let X be a uniform random variable defined over a finite set $\mathcal{X}$ of cardinality N. Let $\varepsilon \in [0,1)$ and let f_1, f_2, ..., f_t be t noisy leakage functions defined over $\mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. For any real number $\alpha \in (0,1]$, if $\varepsilon \leq \frac{\alpha}{tN}$, then we have*

$$\beta(X|f_1(X), f_2(X), \ldots, f_t(X)) \ \leq \ \left(\left(\frac{\mathrm{e}^\alpha - 1}{\alpha}\right)t + \mathrm{e}^\alpha\right)\varepsilon \ .$$

The bound in Theorem 2 shows that the bias of X given t leakages increases linearly with t. A requirement is that the bias given a single leakage, namely ε, is t times lower than $\frac{1}{N}$ or less, namely $\varepsilon \leq \frac{\alpha}{tN}$ for some $\alpha \in (0,1]$. Then the bias of X given t leakages is smaller that $\lambda(t)\,\varepsilon$ where λ is an affine function. The value α provides a trade-off between the constraint on ε and the coefficients of λ. If $\alpha = 1$ then $\lambda(t) = (\mathrm{e}-1)t + \mathrm{e} \approx 1.72\,t + 2.72$, and when α approaches 0, then $\lambda(t)$ tends towards $t+1$.

Bias Given Cross-Term Leakages. The next theorem gives an upper bound on the bias of a uniform pair (A, B) given the leakage $(f_{i,j}(A_i, B_j))_{i,j}$.

Theorem 3. *Let A and B be two random variables uniformly distributed over some finite set $\mathcal{X}$ of cardinality N. Let d be a positive integer, and let $(A_i)_i$ and $(B_j)_j$ be two d^{th}-order encodings of A and B respectively. Let ε be a real number such that $\varepsilon \leq \frac{\alpha}{(d+1)N^2}$ for some $\alpha \in (0, 1]$ and let $(f_{i,j})_{i,j}$ be noisy leakage functions defined over $\mathcal{X} \times \mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. We have:*

$$\beta\big((A, B)|(f_{i,j}(A_i, B_j))_{i,j}\big) \leq 2N^{\frac{3d+2}{2}}\big((\lambda_1 d + \lambda_0)\varepsilon\big)^{d+1} ,$$

$\lambda_1 = \frac{e^\alpha - 1}{\alpha}$ and $\lambda_0 = \lambda_1 + e^\alpha$.

In our context, the pair (A, B) is not uniformly distributed but it is of the form $(A, B) = (g(X), h(X))$ where X is uniform, and g and h are polynomial functions. We will then use the following corollary of Theorem 3.

Corollary 2. *Let X be a random variable uniformly distributed over some set $\mathcal{X}$ and let g and h be deterministic functions from $\mathcal{X}$ to $\mathcal{X}$. Let d be a positive integer and let $(G_i)_i$ and $(H_j)_j$ be d^{th}-order encodings of $g(X)$ and $h(X)$ respectively. Let ε be a real number such that $\varepsilon \leq \frac{\alpha}{(d+1)N^2}$ for some $\alpha \in (0, 1]$. And let $(f_{i,j})_{i,j}$ be noisy leakage functions defined over $\mathcal{X} \times \mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. We have:*

$$\beta\big(X \mid (f_{i,j}(G_i, H_j))_{i,j}\big) \leq 2N^{\frac{3d+7}{2}}\big((\lambda_1 d + \lambda_0)\varepsilon\big)^{d+1} ,$$

$\lambda_1 = \frac{e^\alpha - 1}{\alpha}$ and $\lambda_0 = \lambda_1 + e^\alpha$.

The bound in Corollary 2 shows that the bias of X given the leakages on all the pairs of shares (A_i, B_j) decreases exponentially with d. A requirement is that the bias given a single leakage, namely ε, is $(d + 1)$ times lower than $\frac{1}{N^2}$ or less, namely $\varepsilon \leq \frac{\alpha}{(d+1)N^2}$ for some $\alpha \in (0, 1]$. Then the bias of X given the $(d + 1)^2$ leakages is smaller that $(\lambda(d)\,\varepsilon)^{d+1}$ where λ is an affine function. Once again, the value α provides a trade-off between the constraint on ε and the coefficients of λ.

In the next section, we will use the theorems and corollaries introduced above to deduce a security bound for a full masked implementation of block cipher.

6 Overall Security Proof

In this section, we formally prove the security of the physical implementation $\mathcal{I} = (\mathsf{C}_i, f_i)_i$ of a block cipher following the scheme described in Section 4 with masking order d. Our security proof is *information theoretic*: we show that the information about the cipher input (message and secret key) provided by the overall leakage within an execution of $\mathcal{I}$ is upper bounded by a *negligible function* of the masking order d.

The cipher is assumed to involve t_{lin} linear transformations, t_{nlm} nonlinear multiplications and t_{aff} affine functions (in the s-boxes evaluations). The plaintext and the secret key in input of the cipher are modeled as uniform random variables M and K respectively. The input of every elementary calculation C_i is modeled as a random variable X_i and C_i leaks a noisy function f_i of X_i. The designer is then allowed to choose a noise parameter ψ_i linear in the security parameter d, such that the leakage function f_i lies in the class of noisy functions $\mathcal{N}(1/\psi_i)$.

Theorem 4. *Let d be a positive integer and let $\mathcal{I} = (\mathsf{C}_i, f_i)_{1 \leq i \leq q}$ be the physical implementation of a block cipher under the scheme described in Section 4 with masking order d. For any positive real number ω, there exists a family of real numbers $\psi_i = O(d)\omega$ such that if f_i lies in $\mathcal{N}(1/\psi_i)$ for every i, then the mutual information between the cipher input (M, K) and the overall leakage $(f_1(X_1), f_2(X_2), \ldots, f_q(X_q))$ satisfies*

$$\mathrm{I}\big((M,K); (f_1(X_1), f_2(X_2), \ldots, f_q(X_q))\big) \;\leq\; \frac{T}{\omega^{d+1}}\ , \tag{2}$$

where $T = 2t_{\boldsymbol{nlm}} + t_{\boldsymbol{aff}} + t_{\boldsymbol{lin}}$.

The rest of this section provides a proof of Theorem 4 and exhibits the noise parameters ψ_i that must be chosen for every elementary calculation C_i.

From the description of the scheme in Section 4.2, the overall sequence of elementary calculations of the protected block cipher algorithm can be split into several subsequences separated by masks-refreshing calculations. These subsequences are of four types:

1. $\big(z_i \leftarrow g(x_i)\big)_i$ where g is a linear function of the block cipher,
2. $\big(z_i \leftarrow g(x_i)\big)_i$ where g is an affine function of an s-box evaluation,
3. $\big(v_{i,j} \leftarrow a_i \times b_j\big)_{i,j}$ (first step of a secure nonlinear multiplication),
4. $\big(t_{i,j} \leftarrow t_{i,j-1} \oplus v_{i,j}\big)_{i,j}$ (fourth step of a secure nonlinear multiplication).

All the remaining elementary calculations are masks-refreshing calculations which are leak-free by assumption.

Let us denote by $\mathcal{I}_1$, $\mathcal{I}_1$, ..., $\mathcal{I}_T$ the successive subsequences of elementary calculations and by L_1, L_2, ..., L_T their respective leakages. For every $t \in \{1, 2, \ldots, T\}$, the leakage L_t satisfies

$$L_t = \begin{cases} \big(f_i(X_i)\big)_i & \text{if } \mathcal{I}_t \text{ is of type 1 or 2,} \\ \big(f_{i,j}(A_i, B_j)\big)_{i,j} & \text{if } \mathcal{I}_t \text{ is of type 3,} \\ \big(f_{i,j}(T_{i,j-1}, V_{i,j})\big)_{i,j} & \text{if } \mathcal{I}_t \text{ is of type 4.} \end{cases}$$

where the f_i or $f_{i,j}$ are the leakage functions associated to the elementary calculations in $\mathcal{I}_t$ and where the $(X_i)_i$, $(A_i)_i$, $(B_j)_j$, $(V_{i,j})_{i,j}$, or $(T_{i,j})_{i,j}$ are random variables modeling the elementary calculations inputs in $\mathcal{I}_t$ (note that the indexing is intra-subsequence and it differs from that in Theorem 4).

A crucial point of our security proof is that every subsequence operates on shares with fresh random masks. As a result, given a cipher input $(M, K) = (m, k)$, the encodings processed in the different subsequences are mutually independent, which in turn implies that the leakages of the different subsequences are mutually independent (since by definition, the randomness introduced by a noisy leakage function is independent of the randomness introduced by the others). We deduce that the entropy of the overall leakage $(L_1, L_2, \ldots, L_T)$ knowing the cipher input (M, K) satisfies:

$$\mathrm{H}[(L_1, L_2, \ldots, L_T)|(M, K)] = \sum_{t=1}^{T} \mathrm{H}[L_t|(M, K)] \ .$$

The mutual information between the cipher input and the overall leakage therefore satisfies:

$$\begin{aligned} \mathrm{I}\big((M, K); (L_1, L_2, \ldots, L_T)\big) &= \mathrm{H}[(L_1, L_2, \ldots, L_T)] - \sum_{t=1}^{T} \mathrm{H}[L_t|(M, K)] \\ &\leq \sum_{i=1}^{T} \mathrm{I}((M, K); L_t) \ , \end{aligned}$$

where the inequality holds since $\mathrm{H}[(L_t)_t] \leq \sum_t \mathrm{H}[L_t]$.

For every subsequence $\mathcal{I}_t$, there exists a uniform random variable[3] $X_t = g(M, K)$ such that $\mathrm{I}((M, K); L_t) = \mathrm{I}(X_t; L_t)$. To complete the proof of Theorem 4, we now demonstrate that the mutual information $\mathrm{I}(X_t; L_t)$ is upper bounded by $(1/\omega)^{d+1}$ for some noise parameter $\psi_t = O(d)\omega$, for every $t \in \{1, 2, \ldots, T\}$. Due to Proposition 1, this is equivalent to prove that the bias of X_t given L_t satisfies

$$\beta(X_t|L_t) \leq \frac{\ln 2}{N} \left(\frac{1}{\omega}\right)^{d+1} \ , \tag{3}$$

where N is the cardinal of the definition set of X_t. In the rest of the section, we demonstrate the claim for every type of subsequence.

Security of Type 1 Subsequences. In a type 1 subsequence every elementary calculation processes a share of the uniform input X_t of a linear function. The security of such a subsequence directly holds from Theorem 1. Indeed, according to Theorem 1 with $\varepsilon = \psi_t^{-1}$, choosing a noise parameter $\psi_t = c\omega$ with a constant c satisfying $c^{d+1} \geq (\ln 2)^{-1} N^{\frac{d+2}{2}}$ implies bound (3). In particular c can be taken equal to $(\ln 2)^{-\frac{1}{2}} N^{\frac{3}{4}} \approx 1.44 N^{\frac{3}{4}}$ for any $d \geq 1$ and equal to a value close to $1.44\sqrt{N}$ for a large d.

[3] For a subsequence of type 1, this variable is simply the (unmasked) input of the corresponding linear transformation (which is indeed uniform since the cipher input is uniform by assumption, and according to the uniformity property stated in Section 4.1). For a subsequence of type 2, 3 or 4, this variable is the (unmasked) input of the corresponding s-box (which is also uniformly distributed for the same reasons).

Security of Type 2 Subsequences. In a type 2 subsequence every elementary calculation processes a share G_i of an encoding of $g(X_t)$ where X_t is a uniform s-box input and g is some polynomial function.

According to Corollary 1, choosing a noise parameter $\psi_t = c\omega$ with a constant c satisfying $c^{d+1} \geq (\ln 2)^{-1} N^{\frac{d+5}{2}}$ implies bound (3). In particular c can be taken equal to $(\ln 2)^{-\frac{1}{2}} N^{\frac{3}{2}} \approx 1.44 N^{\frac{3}{2}}$ for any $d \geq 1$ and equal to a value close to $1.44\sqrt{N}$ for a large d.

Security of Type 3 Subsequences. In a type 3 subsequence every elementary calculation processes a multiplication from a pair of shares (A_i, B_j), where $(A_i)_i$ and $(B_j)_j$ are d^{th}-order encodings of two variables A and B to multiply. Both A and B rely on a uniform s-box input X_t by $A = g(X_t)$ and $B = h(X_t)$ for some polynomial functions g and h.

According to Corollary 2, choosing a noise parameter $\psi_t = \lambda(d)\,\omega$ for a subsequence $\mathcal{I}_t$ of type 3 implies bound (3), as long as there exists $\alpha \in (0; 1]$ such that $\lambda(d)$ satisfies

$$\lambda(d) \geq \frac{N^2}{\alpha\,\omega}(d+1) \quad \text{and} \quad \lambda(d) \geq c(\lambda_{1,\alpha}\, d + \lambda_{0,\alpha}) \ ,$$

where $\lambda_{1,\alpha} = \frac{\mathrm{e}^\alpha - 1}{\alpha}$, $\lambda_{0,\alpha} = \lambda_{1,\alpha} + \mathrm{e}^\alpha$ and c is a constant satisfying $c^{d+1} \geq 2(\ln 2)^{-1} N^{\frac{3d+9}{2}}$. In particular c can be taken to $(\ln 2)^{-\frac{1}{2}} N^3 \approx 1.44 N^3$ for any $d \geq 1$ and equal to a value close to $1.44 N^{\frac{3}{2}}$ for a large d.

The first constraint aims to meet the requirement on ε for Corollary 2 and the second constraint implies bound (3). To summarize, the best choice is to take λ as

$$\lambda(d) = \min_{\alpha \in (0;1]} \max\Big(\frac{N^2}{\alpha\,\omega}(d+1)\ ,\ c(\lambda_{1,\alpha}\, d + \lambda_{0,\alpha})\Big) \ .$$

Security of Type 4 Subsequences. In a type 4 subsequence every elementary calculation processes an addition $T_{i,j} = T_{i,j-1} \oplus V_{i,j}$ for $0 \leq i \leq d$ and $1 \leq j \leq d$, and where $T_{i,0} = V_{i,0}$. For every i, the sequence of additions results in a share $Z_i = T_{i,d}$ of the underlying multiplication output. That is $(Z_0, Z_1, \ldots, Z_d)$ is an encoding of $g(X_t)$ where X_t is a uniform s-box input and g is some polynomial function over $\mathcal{X}$. Each elementary calculation takes as input a pair $(T_{i,j-1}, V_{i,j})$ and leaks $f_{i,j}(T_{i,j-1}, V_{i,j})$ where $f_{i,j} \in \mathcal{N}(1/\psi_t)$. Our goal is to set ψ_t such that the bias $\beta\big(X_t | (F_0(Z_0), F_1(Z_1), \ldots, F_d(Z_d))\big)$ satisfies bound (3), where

$$F_i(Z_i) = \big(f_{i,1}(T_{i,0}, V_{i,1}), f_{i,2}(T_{i,1}, V_{i,2}) \ldots, f_{i,d}(T_{i,d-1}, V_{i,d})\big) \ .$$

Note that F_i can be seen as a noisy leakage function applied to Z_i (and where $V_{i,1}$, $V_{i,2}$, …, $V_{i,d}$ are seen as the internal randomness of F_i).

We first analyse the bias on each Z_i given the leakage $F_i(Z_i)$. Let $f'_{i,j}$ be the noisy leakage functions defined by $f'_{i,j} : (X, Y) \mapsto f_{i,j}(X, X \oplus Y)$. We can then rewrite $F_i(Z_i)$ as

$$F_i(Z_i) = \big(f'_{i,1}(T_{i,0}, T_{i,1}), f'_{i,2}(T_{i,1}, T_{i,2}) \ldots, f'_{i,d}(T_{i,d-1}, T_{i,d})\big) \ ,$$

and we have $f'_{i,j} \in \mathcal{N}(1/\psi_t)$ for every (i,j) by definition of the bias.[4] Moreover $T_{i,0}$, $T_{i,1}$, ..., $T_{i,d}$ are uniformly distributed and mutually independent, and $T_{i,d} = Z_i$. We then apply the following lemma.

Lemma 1. *Let $T_0, T_1, \ldots, T_d$ be $d+1$ independent random variables uniformly distributed over some set $\mathcal{X}$ of cardinality N. Let $\varepsilon \in [0,1)$ and let $f_1, f_2, \ldots, f_d$ be noisy leakage functions defined over $\mathcal{X} \times \mathcal{X}$ and belonging to $\mathcal{N}(\varepsilon)$. We have:*

$$\beta\big(T_d \big| f_1(T_0,T_1), f_2(T_1,T_2), \ldots, f_d(T_{d-1},T_d)\big) \leq 2N\varepsilon \ .$$

Lemma 1 implies $\beta(Z_i|F_i(Z_i)) \leq 2N/\psi_t$ for every i. Then by Corollary 1, we get

$$\beta\big(X_t|(F_0(Z_0), F_1(Z_1), \ldots, F_d(Z_d))\big) \leq N^{\frac{d+3}{2}}\Big(\frac{2N}{\psi_t}\Big)^{d+1} = N^{\frac{3d+5}{2}}\Big(\frac{2}{\psi_t}\Big)^{d+1} \ .$$

According to the above inequality, choosing a noise parameter $\psi_t = c\,\omega$ with a constant c satisfying $c^{d+1} \geq 2^{d+1}(\ln 2)^{-1}N^{\frac{3d+7}{2}}$ implies bound (3). In particular c can be taken equal to $2(\ln 2)^{-1}N^{\frac{5}{2}} \approx 2.89N^{\frac{5}{2}}$ for any $d \leq 1$ and equal to a value close to $2.89N^{\frac{3}{2}}$ for a large d.

References

1. Blakely, G.: Safeguarding cryptographic keys. In: National Comp. Conf., vol. 48, pp. 313–317. AFIPS Press, New York(1979)
2. Blömer, J., Guajardo, J., Krummel, V.: Provably Secure Masking of AES. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 69–83. Springer, Heidelberg (2004)
3. Carlet, C., Goubin, L., Prouff, E., Quisquater, M., Rivain, M.: Higher-order masking schemes for s-boxes. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 366–384. Springer, Heidelberg (2012)
4. Chari, S., Jutla, C., Rao, J., Rohatgi, P.: Towards Sound Approaches to Counteract Power-Analysis Attacks. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 398–412. Springer, Heidelberg (1999)
5. Chari, S., Rao, J., Rohatgi, P.: Template Attacks. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003)
6. Clavier, C., Coron, J.-S., Dabbous, N.: Differential Power Analysis in the Presence of Hardware Countermeasures. In: Koç, Ç.K., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 252–263. Springer, Heidelberg (2000)
7. Coron, J.-S., Kizhvatov, I.: Analysis and Improvement of the Random Delay Countermeasure of CHES 2009. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 95–109. Springer, Heidelberg (2010)
8. Coron, J.-S., Kocher, P., Naccache, D.: Statistics and secret leakage. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 157–173. Springer, Heidelberg (2001)
9. Di Crescenzo, G., Lipton, R.J., Walfish, S.: Perfectly Secure Password Protocols in the Bounded Retrieval Model. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 225–244. Springer, Heidelberg (2006)

[4] For every noisy leakage function f defined over $\mathcal{X} \times \mathcal{X}$, and for $f' : (X,Y) \mapsto f(X, X \oplus Y)$, we have $\beta\big((X,Y)\big|f(X,Y)\big) = \beta\big((X,Y')\big|f'(X,Y')\big)$ where $Y' = X \oplus Y$.

10. Dodis, Y., Pietrzak, K.: Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 21–40. Springer, Heidelberg (2010)
11. Dziembowski, S., Faust, S.: Leakage-Resilient Circuits without Computational Assumptions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 230–247. Springer, Heidelberg (2012)
12. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302. IEEE Computer Society (2008)
13. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-Resilient Signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010)
14. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 135–156. Springer, Heidelberg (2010)
15. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic Analysis: Concrete Results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001)
16. Genelle, L., Prouff, E., Quisquater, M.: Thwarting higher-order side channel analysis with additive and multiplicative maskings. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 240–255. Springer, Heidelberg (2011)
17. Goldwasser, S., Rothblum, G.N.: Securing computation against continuous leakage. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 59–79. Springer, Heidelberg (2010)
18. Goldwasser, S., Rothblum, G.N.: How to Compute in the Presence of Leakage. In: 53rd Annual IEEE Symposium on Foundations of Computer Science – FOCS 2012, pp. 31–40. IEEE Computer Society (2012)
19. Goubin, L., Patarin, J.: DES and Differential Power Analysis – The Duplication Method. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 158–172. Springer, Heidelberg (1999)
20. Herbst, C., Oswald, E., Mangard, S.: An AES Smart Card Implementation Resistant to Power Analysis Attacks. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 239–252. Springer, Heidelberg (2006)
21. Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
22. Juma, A., Vahlis, Y.: Protecting Cryptographic Keys against Continual Leakage. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 41–58. Springer, Heidelberg (2010)
23. Kiltz, E., Pietrzak, K.: Leakage Resilient ElGamal Encryption. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 595–612. Springer, Heidelberg (2010)
24. Kocher, P.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
25. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
26. Macé, F., Standaert, F.-X., Quisquater, J.-J.: Information Theoretic Evaluation of Side-Channel Resistant Logic Styles. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 427–442. Springer, Heidelberg (2007)
27. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks – Revealing the Secrets of Smartcards. Springer (2007)

28. Messerges, T.: Securing the AES Finalists against Power Analysis Attacks. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 150–164. Springer, Heidelberg (2001)
29. Micali, S., Reyzin, L.: Physically Observable Cryptography (Extended Abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
30. Oswald, E., Mangard, S., Pramstaller, N., Rijmen, V.: A Side-Channel Analysis Resistant Description of the AES S-box. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 413–423. Springer, Heidelberg (2005)
31. Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons. Integration 40(1), 52–60 (2007)
32. Pietrzak, K.: A Leakage-Resilient Mode of Operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)
33. Prouff, E., Roche, T.: Higher-order glitches free implementation of the aes using secure multi-party computation protocols. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 63–78. Springer, Heidelberg (2011)
34. Rivain, M., Prouff, E.: Provably Secure Higher-Order Masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 413–427. Springer, Heidelberg (2010)
35. Rothblum, G.N.: How to compute under$\mathcal{AC}^0$leakage without secure hardware. In: Safavi-Naini, R. (ed.) CRYPTO 2012. LNCS, vol. 7417, pp. 552–569. Springer, Heidelberg (2012)
36. Schindler, W., Lemke, K., Paar, C.: A Stochastic Model for Differential Side Channel Cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005)
37. Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612–613 (1979)
38. Standaert, F.-X., Archambeau, C.: Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 411–425. Springer, Heidelberg (2008)
39. Standaert, F.-X., Örs, S.B., Preneel, B.: Power Analysis of an FPGA: Implementation of Rijndael: Is Pipelining a DPA Countermeasure? In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 30–44. Springer, Heidelberg (2004)
40. Standaert, F.-X., Pereira, O., Yu, Y., Quisquater, J.-J., Yung, M., Oswald, E.: Leakage resilient cryptography in practice. Cryptology ePrint Archive, Report 2009/341 (2009), `http://eprint.iacr.org/`
41. Tiri, K., Verbauwhede, I.: A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In: Design, Automation and Test in Europe Conference and Exposition – DATE 2004, pp. 246–251. IEEE Computer Society (2004)
42. Yu, Y., Standaert, F.-X., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security – CCS 2010, pp. 141–151 (2010)