Spectral features of higher-order side-channel countermeasures

V. Zaccaria, F. Melzani, G. Bertoni

Abstract

This brief deals with the problem of mathematically formalizing hardware circuits' vulnerability to side-channel attacks. We investigate whether spectral analysis is a useful analytical tool for this purpose by building a mathematically sound theory of the vulnerability phenomenon. This research was originally motivated by the need for deeper, more formal knowledge around vulnerable nonlinear circuits. However, while building this new theoretical framework, we discovered that it can consistently integrate known results about linear ones as well. Eventually, we found it adequate to formally model side-channel leakage in several significant scenarios. In particular, we have been able to find the vulnerability perimeter of a known cryptographic primitive (i.e., Keccak [1]) and thus tackle the analysis of vulnerability when signal glitches are present. We believe the conceptual framework we propose will be useful for researchers and practitioners in the field of applied cryptography and side-channel attacks.

1 Introduction

In modern days, designing a hardware cryptographic primitive requires a countermeasure against side-channel attacks as well [2]. Nevertheless, a complete theory for

- *V. Zaccaria is with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano - Italy. Email: vittorio.zaccaria@polimi.it*
- *F. Melzani is with STMicroelectronics, Agrate Brianza - Italy. Email: filippo.melzani@st.com*
- *G. Bertoni is with STMicroelectronics, Agrate Brianza - Italy. Email: guido.bertoni@st.com*

reasoning formally about countermeasures continues to slip through the efforts of the cryptographic research community.

In this brief, we present a mathematical formalization for reasoning symbolically about such countermeasures. The main result of this work is the discovery of important mathematical rules that connect a successful correlation power attack to the Fourier expansion of the leakage under scrutiny. Originally, we started this research effort to extend some recent results [3] to cover a broader range of countermeasures (such as *Boolean masking* and *threshold implementations*). Eventually, however, we discovered an elegant yet effective way to analyze any countermeasure from the vulnerability standpoint. We admit that this is a significant claim that we hope to substantiate in the following pages. To frame this work in the current research context, we note that today there are mainly two "schools of thought" that address the same problem. On one side, some approaches try to decide whether a circuit is vulnerable through formal or static-type checking [4]–[6]. On the other side, a designer rushing to release its primitive to production (s)he is more prone to detect circuit's vulnerability through a simulation-based approach [7]. We believe these methods are very important during the mid to final stages of the design to verify the original protection claims. However, when we need to set those claims, we are at a loss in terms of mathematical tools to find them precisely. Eventually, we typically resort to more pragmatic approaches that, although increasing our confidence, might yield non-negligible overhead.

To address this issue, we start from classic results in the context of the analysis of correlation-immune Boolean functions [8], [9] (Section 2) and introduce a few novel theorems that precisely govern the correlation immunity when Boolean functions and leakages are manipulated through classic functional algebra (Section 3). We then show how to apply the introduced conceptual tools to confirm the protection claims of a well-known threshold implementation of Keccak (see Section 4). Eventually, we show how we can discover vulnerability even when glitches are present (Section 5) and we close with what we think are the current limitations and future developments (Section 6).

2 Notation and supporting theorems

In this section we introduce some basic facts about the Fourier expansion of Boolean and pseudo-Boolean functions[1].

Definition 1 (Fourier expansion of a pseudo-Boolean function)**.** The Fourier expansion of a function $f : \mathbb{F}_2^n \to \mathbb{R}$ is a pseudo-Boolean function

$$\mathcal{F}[f] \equiv \hat{f}(\gamma) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} f(x)\chi_\gamma(x)$$

where $\chi_\gamma(x) = (-1)^{\gamma \cdot x}$ is called *Fourier character* or parity function and forms an orthonormal basis for the vector space for all functions $f : \mathbb{F}_2^n \to \mathbb{R}$ [10]. The *spectral coordinate* $\gamma \in \mathbb{F}_2^n$ identifies a subset of the original n variables while $\hat{f}(\gamma)$ represents, informally, the contribution of the XOR of that subset on the overall function value.

Note that if $f(x) = (-1)^{F(x)}$ with $F : \mathbb{F}_2^n \to \mathbb{F}_2^1$ (i.e., a single output Boolean function), the above expression is conventionally called the *Walsh transform* of $F(x)$ (indicated with $\mathcal{W}[F, \gamma]$). Dually, f can be reconstructed from $\hat{f}$ with the inverse Fourier expansion:

$$\mathcal{F}^{-1}[\hat{f}] \equiv f(x) = \sum_{\gamma \in \mathbb{F}_2^n} \hat{f}(\gamma)\chi_\gamma(x) \tag{1}$$

Example 1. Let us consider the function $F : \mathbb{F}_2^2 \to \mathbb{F}_2^1$ defined as

$$F(x) = x_0 x_1.$$

Its Walsh transform can be computed as:

$$\begin{aligned}
\mathcal{W}[F, \gamma] &= 2^{-2} \sum_{x \in \mathbb{F}_2^2} (-1)^{x_0 x_1} (-1)^{\gamma_0 x_0 + \gamma_1 x_1} \\
&= 2^{-2} \sum_{x \in \mathbb{F}_2^2} (-1)^{x_0 x_1 + \gamma_0 x_0 + \gamma_1 x_1} \\
&= 2^{-2} \{1 + (-1)^{\gamma_0} + (-1)^{\gamma_1} - (-1)^{\gamma_0 + \gamma_1}\}
\end{aligned}$$

Now, to compute the contribution on f of a specific subset $S \subseteq \{x_0, x_1\}$, it suffices to evaluate $\mathcal{W}[F, \gamma]$ for $\gamma = [\gamma_0, \gamma_1]$, where $\gamma_i = 1$ if $x_i \in S$. For example, to compute the

[1] In this manuscript, we will use the following naming convention: a *pseudo-Boolean* function is a function whose type signature is $\mathbb{F}_2^n \to \mathbb{R}$ while a *Boolean* function is a function whose type signature is $\mathbb{F}_2^n \to \mathbb{F}_2^m$ for any n, m.

contribution of $\{x_0, x_1\}$ on f, we evaluate $\mathcal{W}[F,[1,1]] = 2^{-2}(1-1-1-1) = -1/2$. We can exploit this representation to derive other quantities associated with the degree of dependence of the function on a specific variable, as the following definition shows.

Definition 2 (Covariance and correlation of pseudo-Boolean functions (see [10]))**.** The covariance between $g : \mathbb{F}_2^n \to \mathbb{R}$ and the character function of a Boolean variable $\chi_{\gamma_i}(x) = (-1)^{x_i}$ is:

$$\sigma_{g\chi_{\gamma_i}} = \sum_{\gamma \neq 0} \hat{g}(\gamma)\hat{\chi}_{\gamma_i}(\gamma) = \hat{g}(\gamma_i) \tag{2}$$

Considering that for any character χ_{γ_i} it holds that $\sigma_{\chi_{\gamma_i}} = 1$, then it follows that the expected correlation is:

$$\rho_{g\chi_{\gamma_i}} = \frac{\sigma_{g\chi_{\gamma_i}}}{\sigma_g \sigma_{\chi_{\gamma_i}}} = \frac{\hat{g}(\gamma_i)}{\sigma_g} \tag{3}$$

The classic definition of *correlation immunity* builds above Eq. (3), i.e., a function g is m-th order correlation-immune if and only if $\hat{g}(\gamma_i) = 0$ for all $\gamma_i \in \mathbb{F}_2^n$ such that $1 \leq w_H(\gamma_i) \leq m$ where w_H is the number of bits of γ_i that are 1 (see [8], [9], [11]). However, this is too general for our purpose, as we care only about those variables that are sensitive[2]. Besides, we note that the concept of "order" that is conventionally used in countermeasure theory is different from the one used for correlation immunity. If not stated explicitly, when we refer to the protection order, we adhere to the conventional meaning used for countermeasures which is the order of the statistical moments used to mount the attack [12]. To be more precise, we will use this definition of vulnerability:

Definition 3 (m-th order vulnerability)**.** Given a spectral coordinate γ_s that characterizes only sensitive variables, we say that $g : \mathbb{F}_2^n \to \mathbb{R}$ is *vulnerable* at the m-th order in γ_s if and only if $\mathcal{F}[g^m](\gamma_s) \neq 0$.

Theorem 1 (Spectrum of the Hamming weight)**.** *The Fourier expansion $\hat{H}_n$ of the Hamming*

[2] A side-channel might expose one or many intermediate Boolean values (*visible variables*) because they are effectively processed by the hardware. We call *sensitive variables* the values that are deterministic functions of any master key and public input. Visible variables are not always sensitive themselves because, in Boolean masking, those are combined with random masks to produce visible variables [3].

weight function $H_n : \mathbb{F}_2^n \to \mathbb{R}$ is

$$\hat{H}_n(\gamma) = \frac{n}{2}\delta_{\gamma,0} + \sum_{|\gamma'|=1} -\frac{1}{2}\delta_{\gamma,\gamma'} \tag{4}$$

where δ is the Kronecker delta function. The spectrum is thus 0-concentrated [10] on degree up to 1 because for all degrees $|\gamma|> 1$ it holds that $\hat{H}_n(\gamma) = 0$.

Proof. First, we observe that, given a Boolean variable x_i, we can use the expression $(1 - (-1)^{x_i})/2$ to compute its Hamming weight $H_1(x_i)$. The following derivation is thus possible

$$\begin{aligned}
&\hat{H}_n(\gamma) \\
&= \mathcal{F}[H_n(x)] = \mathcal{F}[\sum_{i=1}^{n} H_1(x_i)] \\
&= \mathcal{F}[\frac{n}{2} - \frac{\sum_{i=1}^{n} (-1)^{x_i}}{2}] = \mathcal{F}[\frac{n}{2} + \sum_{|\gamma'|=1} -\frac{1}{2}\chi_{\gamma'}(x)] \\
&= \frac{n}{2}\mathcal{F}[1] + \sum_{|\gamma'|=1} -\frac{1}{2}\mathcal{F}[\chi_{\gamma'}(x)] = \frac{n}{2}\delta_{\gamma,0} + \sum_{|\gamma'|=1} -\frac{1}{2}\delta_{\gamma,\gamma'}
\end{aligned}$$

□

Remark 1. The above result can be extended easily to include the case $H_n^{\alpha}(x) = \sum_{i=1}^{n} \alpha_i H_1(x_i)$ (i.e., each bit has a different weight α_i), however, for clarity of exposition, we will concentrate on the former. The extension of the results of this paper to the latter case is mechanical work.

3 A THEORY OF HIGHER ORDER VULNERABILITY

In this section, we set up a few novel mathematical tools which deal with how basic operations of the algebra of (pseudo-)Boolean functions act on the Fourier spectrum, i.e., composition and multiplication. The composition operation is useful to model the power consumption of a digital circuit computing a Boolean function f whose power model is expressed by a pseudo-Boolean function g:

Theorem 2 (Spectrum of the composition of a pseudo-Boolean function and a Boolean

function). *Given a function $h : \mathbb{F}_2^n \to \mathbb{R}$ such that*

$$h = g \circ f$$

where $g : \mathbb{F}_2^m \to \mathbb{R}$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the Fourier expansion of h is related to g and f by the following:

$$\hat{h}(\gamma) = \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') P_f(\gamma', \gamma), \ \gamma \in \mathbb{F}_2^n \tag{5}$$

with

$$P_f(\gamma', \gamma) = \langle \chi_{\gamma'} \circ f, \chi_\gamma \rangle$$

where $\langle \cdot, \cdot \rangle$ is the inner product of functions.

Proof.

$$\begin{aligned}\hat{h}(\gamma) &= 2^{-n} \sum_{x \in \mathbb{F}_2^n} h(x) \chi_\gamma(x) \\ &= 2^{-n} \sum_{x \in \mathbb{F}_2^n} \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') \chi_{\gamma'}(f(x)) \chi_\gamma(x) \\ &= 2^{-n} \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') \sum_{x \in \mathbb{F}_2^n} \chi_{\gamma'}(f(x)) \chi_\gamma(x) \\ &= \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') \langle \chi_{\gamma'} \circ f, \chi_\gamma \rangle \\ &= \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') P_f(\gamma', \gamma)\end{aligned}$$

□

From the previous theorem, we can derive two important corollaries.

Corollary 1 (Spectrum of the composition of a function and a linear, not necessarily invertible transform)**.** Let us assume that $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is a linear transform:

$$f(x) = Kx$$

where K is a $\mathbb{F}_2^m \times \mathbb{F}_2^n$ matrix. Then we have that

$$P_f(\gamma', \gamma) = \delta_{K^\top \gamma', \gamma} \tag{6}$$

where $\delta_{i,j}$ is the Kronecker delta. It follows that for a function $h = g \circ f$, the spectrum of h is related to the spectrum of g through the following relation:

$$\hat{h}(\gamma) = \sum_{\gamma' \in \mathbb{F}_2^m, K^\top \gamma' = \gamma} \hat{g}(\gamma') \tag{7}$$

Proof: see supplemental material.

When $f(x) = Mx$ and M is an invertible matrix, the following corollary holds:

Corollary 2 (Spectrum of the composition of a pseudo-Boolean function and an invertible linear transform)**.** Given two functions $g : \mathbb{F}_2^n \to \mathbb{R}$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ where $f(x) = Mx$ is a linear, invertible transform, the spectrum $\hat{h}$ of the function:

$$h = g \circ f$$

is related to $\hat{g}$ by the following formula[3]:

$$\hat{h}(\gamma) = \hat{g}(M^{-\top}\gamma). \tag{8}$$

Proof. In this case, there is a bijective mapping between γ and γ' so Eq. (6) becomes:

$$P_f(\gamma', \gamma) = \delta_{M^{-\top}\gamma, \gamma'}.$$

Consequently, Eq. (7) can be rewritten as $\hat{h}(\gamma) = \hat{g}(M^{-\top}\gamma)$. □

Example 2 (Countermeasure against first order attacks)**.** We now present a small example to show the usefulness of Corollary 2. Assume the following leakage corresponding to:

$$L(x) = H_2(Mx) + \delta, \; M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \; x = \begin{pmatrix} S \\ T \end{pmatrix} \tag{9}$$

where S is a sensitive variable, T is a random mask and $\delta \sim \mathcal{N}(0, \sigma)$ random noise. We can derive the correlation with S following Eq. (3) and by considering its spectral

[3] We use the operator $-\top$ to indicate the inverse of the transpose of the considered matrix.

coordinate $\gamma_s = (1,0)^\top$ consequently,

$$\begin{aligned}
\rho_{g\chi_{\gamma_s}} &= \frac{\sigma_{g\chi_{\gamma_s}}}{\sigma_g} && \text{[by Eq. (3)]} \\
&\propto \hat{L}((1,0)^\top) && \text{[by Eq. (2)]} \\
&= \hat{H}(M^{-\top}(1,0)^\top) && \text{[by Eq. (8)]} \\
&= \hat{H}((1,1)^\top) \\
&= 0 && \text{[by Eq. (4)]}
\end{aligned}$$

Another way to check for zero correlation is to expand the Hamming weight through Eq. (4) into Eq. (9):

$$\begin{aligned}
&E\left[\delta + H_2(Mx)\right] \\
&= E\left[H_1(T)\right] + E\left[H_1(S+T)\right] \\
&= -\frac{E\left[(-1)^T\right]}{2} - \frac{E\left[(-1)^{S+T}\right]}{2} + 1
\end{aligned}$$

Given that $E[(-1)^T] = 0$ for a random mask T, it follows that, for a deterministic value $S = s$, the expected value of $L(x)$ is 1.

We turn now to the case where we could have multiple leakage points in our circuit. The following theorem allows us to compute the spectrum of a combining function which is the product of two (or more) leakages:

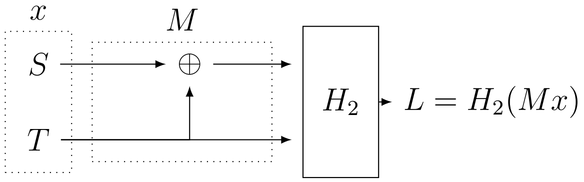

Figure 1. Circuit considered in example 2.

Theorem 3 (Spectrum of the product of two functions)**.** *Given two functions $f : \mathbb{F}_2^n \to \mathbb{R}$, $g : \mathbb{F}_2^n \to \mathbb{R}$, the Fourier expansion of their product is proportional to the convolution of their* *transforms:*

$$\mathcal{F}[f * g] = 2^n(\hat{f} \star \hat{g})$$

Proof.

$$\begin{aligned}
\mathcal{F}[f * g] &= \mathcal{F}[\sum_{\gamma_1 \in \mathbb{F}_2^n} \hat{f}(\gamma_1)\chi_{\gamma_1}(x) \sum_{\gamma_2 \in \mathbb{F}_2^n} \hat{g}(\gamma_2)\chi_{\gamma_2}(x)] \\
&= \mathcal{F}[\sum_{\gamma_1 \in \mathbb{F}_2^n} \hat{f}(\gamma_1)\chi_{\gamma_1}(x) \sum_{k \in \mathbb{F}_2^n} \hat{g}(\gamma_1 + k)\chi_{\gamma_1 + k}(x)] \\
&= \mathcal{F}[\sum_{k \in \mathbb{F}_2^n} \sum_{\gamma_1 \in \mathbb{F}_2^n} \hat{f}(\gamma_1)\hat{g}(\gamma_1 + k)\chi_k(x)] \\
&= \mathcal{F}[2^n \sum_{k \in \mathbb{F}_2^n} (\hat{f} \star \hat{g})(k)\chi_k(x)] \\
&= \mathcal{F}[2^n \mathcal{F}^{-1}[\hat{f} \star \hat{g}]] \\
&= 2^n(\hat{f} \star \hat{g})
\end{aligned}$$

□

Remark 2. The convolution $\hat{f} \star \hat{g}$ is a fundamental building block behind any CPA attack. The product of two leakages $f * g$ is in fact correlated with the sensitive variable γ_s if the covariance with character χ_{γ_s} is not 0, i.e., if its Fourier expansion is not null in γ_s (see Eq. (2) and Definition 3):

$$\begin{aligned}
\mathcal{F}[f * g](\gamma_s) &= 2^n(\hat{f} \star \hat{g})(\gamma_s) \\
&= \sum_{\gamma_1 \in \mathbb{F}_2^n} \hat{f}(\gamma_1)\hat{g}(\gamma_1 + \gamma_s) \\
&= \sum_{a_1, a_2 \in \mathbb{F}_2^n, a_1 + a_2 = \gamma_s} \hat{f}(a_1)\hat{g}(a_2) \\
&\neq 0
\end{aligned}$$

Considering a single leakage f, one could easily extend the above condition to the p-th power of f, which is thus vulnerable when the following holds:

$$\begin{aligned}
\mathcal{F}[f^p] &= 2^{(p-1)n} \underbrace{(\hat{f} \star \hat{f} \ldots \star \hat{f})}_{p \text{ times}}(\gamma_s) \\
&= \sum_{a_1 + \cdots + a_p = \gamma_s} \prod_{i=1}^{p} \hat{f}(a_i) \\
&\neq 0
\end{aligned}$$

This observation leads us to the following theorem which, rather unsurprisingly, subsumes the XOR-condition introduced in [3] where M describes the matrix associated with the *visible variables*.

Theorem 4 (Vulnerability conditions for a leakage that is the Hamming weight of a linear combination of variables)**.** *The leakage $L = H(Mx) + \delta$, where M is an $n \times n$ invertible transform, is vulnerable at the p-th order if and only if there exists a set Γ satisfying:*

$$\begin{aligned} \Gamma \subset \mathbb{F}_2^n = \{\gamma_i\} \wedge \\ |\Gamma|= p \wedge \\ \forall \gamma_i, |\gamma_i|= 1 \wedge \\ \sum_i M^\top \gamma_i = \gamma_s \end{aligned} \tag{10}$$

where γ_s is the coordinate in the Fourier spectrum corresponding to the sensitive variable.

Proof. By Definition 3, we know that the following must hold:

$$\begin{aligned} \mathcal{F}[L^p](\gamma_s) &= \sum_{a_1+\cdots+a_p=\gamma_s} \prod_{i=1}^{p} \hat{L}(a_i) \\ &= \sum_{\gamma_1+\cdots+\gamma_p=M^{-\top}\gamma_s} \prod_{i=1}^{p} \hat{H}_n(\gamma_i) \\ &\neq 0 \end{aligned}$$

Assuming that there are solutions to the condition expressed below the sum, we must ensure that all the factors of the product are different from 0 and that products do not cancel in the sum. Since $\hat{H}_n$ is 0-concentrated on degree up to 1 (see Eq. (4)), this is possible only when $|\gamma_i| \in \{0, 1\}$. In particular, if, for all γ_i, $|\gamma_i|= 1$ then products do not cancel because they have all the same sign (this case is particularly important and is captured by the following definition). □

Definition 4 (Minimum vulnerability order)**.** We say that p is the **minimum vulnerabil-**

ity order if for any $\bar{p} \leq p$ equation Eq. (10) does not have solutions; for such a p, it holds[4] that $|\gamma_i| = 1, \forall \gamma_i$. It follows that, if there exists a minimum vulnerability order, it is such that:

$$H_n(M^\top \gamma_s) = p \tag{11}$$

as the last condition in Eq. (10) implies. *In the following part of the paper, we will always deal, unless stated otherwise, with the minimum vulnerability order of a countermeasure.*

Example 3. Let us consider again Example 2; The above Eq. (11) mandates that if there is any vulnerability, the minimum vulnerability order p is such that:

$$p = H(\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}) = H(\begin{pmatrix} 1 \\ 1 \end{pmatrix}) = 2$$

Indeed, the equation

$$\gamma_1 + \cdots + \gamma_p = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

has the following solutions for $p = 2$:

$$\gamma_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \gamma_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

so we say that L *is vulnerable at the second order*. To verify this finding, we expand the Hamming weight in the expression of L through Eq. (4):

$$\begin{aligned} & E\left[(\delta + H_2(Mx))^2\right] \\ & = E\left[(H_1(S+T) + H_1(T) + \delta)^2\right] \\ & = E\left[\left(1 + \delta - \frac{(-1)^{S+T}}{2} - \frac{(-1)^T}{2}\right)^2\right] \\ & = \frac{E\left[(-1)^S\right]}{2} + \sigma^2 + \frac{3}{2} \end{aligned}$$

One could thus readily see that there is a term $\frac{(-1)^S}{2}$ that is directly correlated with the sensitive variable. By solving for $(-1)^S$ and averaging over T, one could directly derive an unbiased estimator for $(-1)^S$:

$$C(L) = 2 * L^2 - 3 - 2\sigma^2 \ \text{ s.t. } \ E[C(L)] = (-1)^S \tag{12}$$

[4] In fact a p-order solution Γ with a $|\gamma_i| = 0$ does not actually increase the vulnerability order as the remaining γ_i constitute a solution for order $p - 1$.

3.1 Non-uniformity of distributions of masks/shares

Interestingly enough, the applicability of Theorem 3 extends to the problem of the non-uniformity of random distributions of shares. As noted by previous authors [13]–[15], it is an issue that might increase the vulnerability of an implementation. Let us reconsider Example 2 where T now is a random variable in $\mathbb{F}_2^1$ with the following non-uniform *probability mass function* (PMF)[5]:

$$\pi_T(t) = \delta_{\top,t} p_1 + \delta_{\bot,t}(1-p_1)$$

The joint probability distribution of $x = (S, T)$ is thus

$$\pi_x(x) = \pi_x([s,t]) = \delta_{\bar{S},s}\pi_T(t)$$

where $\bar{S}$ is the deterministic but unknown sensitive value. Not surprisingly, π_x is a pseudo-Boolean function as well, thus we can compute the average of the leakage as:

$$E_{x\sim\pi_x}[L(x)] = E[\pi_x(x)L(x)]$$

following some basic fact of the theory of Boolean functions [10]. This observation is fundamental, because the pseudo-Boolean function π_x arising from the non-uniformity of T *becomes, in fact, an additional leakage*. More importantly, averaging $L(x)$ in the non-uniform scenario is similar to *a bi-variate attack on the uniform scenario*. In fact, we can rewrite the above expression into[6]

$$\begin{aligned}
E[\pi_x(x)L(x)] &= E[\delta_{\bar{S},S}\pi_T(T)L(x)] \\
&= \frac{1}{2}E[\pi_T(T)L(x)] \\
&= \frac{1}{2}E[(H(T)p_1 + (1-p_1)(1-H(T)))L(x)] \\
&= \frac{1}{2}E[\underbrace{(H(T)(2p_1-1)+1-p_1)}_{\propto H(T)}L(x)]
\end{aligned}$$

where one can observe that the leakage $L(x)$ is multiplied by a quantity proportional to $H(T)$. If we compute the spectral expansion of this product (in the spirit of Remark 2)

[5] We use the Kronecker delta to represent the probability density of each of the elements of the domain of the random variable.

[6] Basically, $\delta_{\top,t}$ becomes the Hamming weight $H(T)$ when included in the expectation computation, while $\delta_{\bot,t}$ becomes $1 - H(T)$.

we can find that it is correlated with the sensitive variable, thus vulnerable, only when $p_1 \neq 1/2$.

3.2 Extension to non-invertible transforms

In this section, we extend Theorem 4 to consider p different leakages of the form $L_i = H \circ Q_i$ (where Q_i is not necessarily invertible) as the following corollary shows:

Theorem 5 (Vulnerability conditions for a set of (linear) leakages)**.** *The product of p leakages $\boldsymbol{L} = [L_1, \dots L_i \dots L_p]$ of the form:*

$$L_i(x) = H(Q_i x), \forall i$$

(where Q_i is a linear non-invertible transform) is said p-vulnerable in γ_s if there exists a set Γ:

$$\begin{gathered} \Gamma \subset \mathbb{F}_2^n = \{\gamma_i\}, |\Gamma| = p, \sum_i \gamma_i = \gamma_s \wedge \\ \forall \gamma_i, \gamma_i \neq 0 \wedge \exists \gamma' \textit{ s.t. } Q_i^\top \gamma' = \gamma_i \wedge |\gamma'| = 1 \end{gathered} \tag{13}$$

Proof. The proof of the above statement is implied by the conditions under which the following Fourier expansion of the product is different from zero:

$$\begin{aligned} \mathcal{F}[\prod_i L_i](\gamma_s) &= \sum_{\gamma_1 + \dots + \gamma_p = \gamma_s} \prod_{i=1}^{p} \hat{L}(\gamma_i) \\ &= \sum_{\gamma_1 + \dots + \gamma_p = \gamma_s} \prod_{i=1}^{p} \sum_{\gamma', K^\top \gamma' = \gamma_i} \hat{H}(\gamma') \\ &\neq 0 \end{aligned}$$

□

3.3 Extension to single output non-linear functions

Building up from the previous theorems, in this section we show how to detect the vulnerability for a general class of non-linear functions which, we believe, covers important practical cases. In fact, we consider the following function composition:

$$h = g \circ f \tag{14}$$

where $f : \mathbb{F}_2^n \to \mathbb{F}_2^1$ and $g : \mathbb{F}_2^1 \to \mathbb{R}$. We will show that even this seemingly simple case can be used effectively to model multivariate vulnerability.

For now, let us reconsider Eq. 5; in particular we note that the function $P_f(\gamma', \gamma) = \langle \chi_{\gamma'} \circ f, \chi_\gamma \rangle$ is formally equal to the Fourier expansion of:

$$Z^f_{\gamma'}(x) = \chi_{\gamma'}(f(x)) \tag{15}$$

we thus rewrite Eq. (5) as:

$$\hat{h}(\gamma) = \sum_{\gamma' \in \mathbb{F}_2^m} \hat{g}(\gamma') \hat{Z}^f_{\gamma'}(\gamma), \ \gamma \in \mathbb{F}_2^n. \tag{16}$$

We now assume $m = 1$; in practice we focus on the case where f is a single output function, so we can rewrite Eq. 16 as:

$$\begin{aligned}\hat{h}(\gamma) &= \hat{g}(0)\hat{Z}^f_0(\gamma) + \hat{g}(1)\hat{Z}^f_1(\gamma) \\ &= 2^{-n} \sum_{x \in \mathbb{F}_2^n} \hat{g}(0)(-1)^{\gamma x} + \hat{g}(1)(-1)^{f(x)+\gamma x} \\ &= \hat{g}(1)\mathcal{W}[f, \gamma]\end{aligned}$$

which indicates that the spectrum of the composition of two functions $g : \mathbb{F}_2^1 \to \mathbb{R}$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2^1$ is the Walsh spectrum of f scaled by the Fourier coefficients of g. In particular, if $g = H_1$ then, as per Eq. 4, $\hat{g}(1) = -1/2$. We now exploit this finding to extend Theorem 5 to the following:

Corollary 3. Given a set of leakages $\mathbf{L} = [L_1, \dots L_i \dots L_q]$, where

$$L_i(x) = H_1(f_i(Q_i x)), \forall i$$

where $f_i : \mathbb{F}_2^n \to \mathbb{F}_2^1$ and Q_i is a linear non-invertible transform, the product is vulnerable in γ_s if there exists a solution to[7]:

$$\begin{aligned}
&\mathcal{F}[\prod_{i=1}^{p} L_i] \\
&\propto \sum_{a_1+\cdots+a_p=\gamma_s} \prod_{i=1}^{p} \mathcal{W}[f_i \circ Q_i, a_i] \\
&= \sum_{a_1+\cdots+a_p=\gamma_s} \prod_{i=1}^{p} \sum_{\gamma_{ij}\in\mathbb{F}_2^m, Q_i^\top \gamma_{ij}=a_i} \mathcal{W}[f_i, \gamma_{ij}] \\
&\neq 0
\end{aligned}$$

4 A practical case study: Keccak

To show a practical application of the previous findings, let us now consider the architecture of a countermeasure devised for the Keccak algorithm as presented by its inventors [1]. We selected Keccak because it is a real-world standard (SHA-3) yet the core mapping is a simple 3-to-1 bit non-linear function of degree two which makes it amenable to the available space of this brief.

The considered version offers first order protection against SCA by using a Threshold Implementation (TI) with three shares. The focus point of our analysis is the output of each share; in fact we investigate whether there exists a leakage product configuration that correlates with a sensitive variable. In particular, we are going to consider the following leakages[8]:

$$L_c = \quad H(\chi_3(a_0, a_1, a_2, b_1, b_2)) + \delta_1 \tag{17}$$

$$L_a = \quad H(\chi_3(b_0, b_1, b_2, c_1, c_2)) + \delta_2 \tag{18}$$

$$L_b = \quad H(\chi_3(c_0, c_1, c_2, a_1, a_2)) + \delta_3 \tag{19}$$

where a, b and c are understood as variables in $\mathbb{F}_2^3$. We also note that each χ_3 can be

[7] With a small abuse of notation, we use Q_i not only to indicate the matrix associated with the linear transformation but also the transformation itself.

[8] With respect to the original notation we omit the implicit index i and use just the offset to indicate the actual bit of the share considered (e.g., $a_{i+1} \rightarrow a_1$).

rewritten vectorially as:

$$\chi_3(v) = v_0 + (v_1 + 1)v_2 + (v_2 v_3) + (v_1 v_4)$$

thus, defining an $\mathbb{F}_2^9$ vector of shares $y = [a, b, c]$, each L_* above can be rewritten in matrix form as[9]

$$L_* = H(\chi_3(R_* y)) \tag{20}$$

where R_* is a $\mathbb{F}_2^{5\times 9}$ matrix that selects the right bits for each leakage. While the latter equation is similar to the one considered in Corollary 3, we still need to express the shares in terms of the original vector x containing the sensitive variables. Here we follow the share composition proposed in [1], where a and b are uniformly random variables in $\mathbb{F}_2^3$, while c is derived from the sensitive variable s: $c = a + b + s$. We can express this computation in matrix form if we define $x = [s, a, b]^T$ and thus rewrite y as

$$y = Mx, \quad M = \begin{bmatrix} I_3 & I_3 & I_3 \\ 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix} \tag{21}$$

where I_3 is the $\mathbb{F}_2^{3\times 3}$ identity matrix and 0_3 is the $\mathbb{F}_2^{3\times 3}$ null matrix. Eq. 20 can be eventually rewritten as

$$L_* = H_1(\chi_3(R_* M x)) = H_1(\chi_3(Q_* x)) \tag{22}$$

where $Q_* = R_* M$ is thus a leakage specific matrix (see supplemental material for the complete data on R_* and Q_*). Eventually, after deriving the Walsh transform of χ_3, we check whether the condition dictated by Corollary 3 holds for any combination of the three leakages. In theory, we can formulate the corollary as a predicate over (γ_s, p) and solve it with an SMT solver (e.g., Z3 [16]). In practice, however, if one limits itself to three variables and $q \leq 4$, a full search works as well. Table 1 shows the only vulnerabilities found for $q \leq 4$ (as the reader can see, no vulnerabilities with less than 3 variables have been found).

[9] We use an asterisk (*) to mean either a, b or c.

Table 1
Vulnerabilities and combination functions of Keccak TI / 3 shares

q	type	exposed variables	combining function
3	tri-variate	s_0	$L_a * L_b * L_c$
3	tri-variate	$s_0 + s_1$	$L_a * L_b * L_c$
3	tri-variate	$s_0 + s_2$	$L_a * L_b * L_c$
3	tri-variate	$s_0 + s_1 + s_2$	$L_a * L_b * L_c$

5 GLITCH-INDUCED SIDE-CHANNEL LEAKAGE

In this section, we extend the proposed methods to detect vulnerability when a particular kind of glitch is present, i.e., a glitch due to a non-instantaneous transition of the input signals of the circuit, as the following definition shows:

Definition 5 (Functional glitch)**.** A *functional glitch* of a pseudo-Boolean function $f : \mathbb{F}_2^n \to \mathbb{R}$ over a time interval T is an unintended value $f(x_t),\ t \in (0, T), x \in \mathbb{F}_2^n$ where x_t is different from x_0 and x_T (i.e., the intended values of the function at the beginning or the end of the time period).

This definition is useful to model the case where the arrival-times of the input signals of the (synchronous) circuit are different, i.e., *inputs might settle at different times within the clock period*. We also assume that the circuit's output line is driving a high capacitance wire or the input of a register, implying that the leakage (power consumption) of intermediate nodes is negligible.[10] In this context, we can model inputs through an *input transition vector*:

Definition 6 (Input transition vector)**.** An *input transition vector* of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{R}$ over a time interval $[0, T]$ is a vector that combines the function's input value at the beginning of the time interval (i.e., x_0) and at the end of the same (x_T):

$$z = (x_0, x_T)^\top \ \ z : \mathbb{F}_2^{2n}$$

[10] While we acknowledge that this not the most general model, we also point out that simultaneous arrival-times are very difficult to achieve thus a leakage associated with this phenomenon will always be present.

where x_0, x_T are variables in $\mathbb{F}_2^n$. We also define some projection operators:

$$\lceil z \rceil = x_0, \ \lfloor z \rfloor = x_T$$

$\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are called, respectively, the *upper slice* operator and the *lower slice* operator.

The concrete value of the input of the function at the beginning and end of the interval can always be described using its input transition vector z. For example, the following equivalences hold:

$$f(x_0) = f(\lceil z \rceil), \ f(x_T) = f(\lceil P_T z \rceil)$$

where P_T is a permutation that swaps $x_0 \leftrightarrows x_T$.

Assumption 1. We assume here that a 1 bit input signal x can change only once during the time interval T. This allows us to model each intermediate value x_t of the input signal at time t as either x_0 or x_T. This allows us to represent x_t as the concrete value of a specific permutation P_t of z:

$$x_t = \lceil P_t z \rceil. \tag{23}$$

Note that this representation is not suitable when x changes multiple times. For example, if x is meant to be constant across the time interval ($x_0 = x_T$) but changes at $x_{\frac{T}{3}}$ and $x_{\frac{T}{2}}$, the latter two values are inexpressible in terms of Eq. (23).

Our approach considers a fixed set of permutations. While this might seem as a limitation, we note that there are particular design phases (e.g, after physical layout) were an estimate of the delays (and thus of signal switches) can be done. Otherwise, if the number of signals is reasonably low, one could explore the space of possible permutations to find a vulnerability.

Example 4. Let us consider a function $f(q, s) : \mathbb{F}_2^2 \to \mathbb{R}$ with the corresponding input transition vector:

$$z = (q_0, s_0, q_T, s_T)^\top$$

and assume that the concrete input signal undergoes the following transitions:

$$(q_0, s_0)^\top \to (q_T, s_0)^\top \to (q_T, s_T)^\top.$$

It is evident that these signal values can be described just as permutations of the transition vector:

$$\lceil z \rceil \rightarrow \lceil P_q z \rceil \rightarrow \lceil P_s P_q z \rceil$$

where P_s, P_q are, respectively, the permutation matrix exchanging $s_0 \rightarrow s_T$ and $q_0 \rightarrow q_T$. Note that a functional glitch is actually caused by the presence of concrete values (like $\lceil P_q z \rceil$) which do not correspond to neither the initial nor the final value of the function's input.

5.1 Leakage modeling criteria

Previously, we have shown that a functional glitch of a function f at a certain time t can be modeled as:

$$f(x_t) = f(\lceil P_t z \rceil).$$

To check whether such leakage is vulnerable, we characterize the spectrum of the composition of a normal function and an upper slice operator.

Theorem 6. *Given a function $f : \mathbb{F}_2^n \rightarrow \mathbb{R}$, the spectrum of a function $h : \mathbb{F}_2^{2n} \rightarrow \mathbb{R}$ such that:*

$$h(z) = f(\lceil z \rceil)$$

is the following:

$$\hat{h}(\gamma) = \begin{cases} \hat{f}(\lceil \gamma \rceil) & \textit{for } \lfloor \gamma \rfloor = 0 \\ 0 & \textit{otherwise} \end{cases} \tag{24}$$

where γ_0 is the indicator associated with the subset of variables x_0.

Proof: see supplemental material.

Assuming that the attacker can probe the leakage function $f(x_t)$ in each time slot t, she will see samples whose convolution is potentially correlated with a sensitive variable. The case when f is linear can be analyzed through Corollary 2 and Theorem 4. This allows us to introduce the following theorem.

Theorem 7 (Vulnerability conditions for a leakage that is the Hamming weight of a linear combination of variables which change value once within a given time period)**.** *Let us*

Table 2
Signal transitions considered in Example 5.

time slot	0	1	2	3
P_t	I	P_S	$P_T P_S$	$P_U P_T P_S$
$L(z,t)$	$f(s_0,t_0,u_0)$	$f(s_1,t_0,u_0)$	$f(s_1,t_1,u_0)$	$f(s_1,t_1,u_1)$

consider the following leakage function, parameterized by the time-slot t in which the measure takes place:

$$L(z,t) = H_n(M\lceil P_t z \rceil)$$

where P_t is the permutation matrix describing the current state of the variables at time slot t (see Assumption 1). We say that the leakage is minimally vulnerable at the p-th power if there exists a set Γ and a corresponding choice of the time slots t_i satisfying:

$$\begin{aligned} \Gamma = \{\gamma_i\} \wedge \\ |\Gamma|= p \wedge \\ \forall \gamma_i, |M^{-\top}\lceil \gamma_i \rceil|= 1 \wedge \lfloor \gamma_i \rfloor = 0 \wedge \\ \sum_i P_{t_i}^{\top} \gamma_i = \gamma_s \end{aligned} \tag{25}$$

where γ_s is the coordinate in the Fourier spectrum corresponding to the sensitive variable[11].

Proof: see supplemental material.

Remark 3. There is a striking similarity between Theorem 7 and Theorem 4 which can be justified by the fact that each permutation P_{t_i} is a special case of invertible linear transform. In fact, the former can be considered an extension of the latter.

Note that the above theorem allows to detect a possible vulnerability that can be associated with combinations of input signals at the beginning and the end of the considered time period as the following example will show.

[11] As usual, this holds when H_n is a weighted sum as well (see Remark 1).

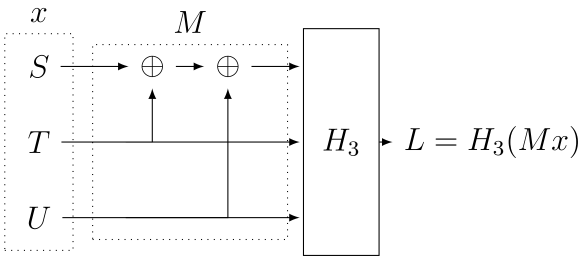

Figure 2. Circuit under scrutiny in Example 5

Example 5. Let us extend Example 2 to include a second order countermeasure and to show how the glitch model applies to it. In practice, we add a second mask U to protect the sensitive variable S (see Figure 2). For all the input variables we consider a single transition over the time interval $[0, 1]$ thus the actual input transition vector will be:

$$z = [s_0, t_0, u_0, s_1, t_1, u_1]^\top .$$

The order of transitions can give rise to different propagation sequences; Table 2 just describes one of them, i.e., the case where S changes at time-slot 1, followed by T and U.

If we want, for example, to detect vulnerabilities at the second order in the first two time slots, we check whether the combinatorial problem in Eq. (25) has any solution (Γ, γ_s) with the following data:

$$p = 2, P_0 = I, P_1 = \begin{pmatrix} 0&0&0&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{pmatrix}, M^{-\top} = \begin{pmatrix} 1&0&0 \\ 1&1&0 \\ 1&0&1 \end{pmatrix}$$

where M and P_t are known from Table 2 and Figure 2. The solution(s) might be multiple; among the interesting ones we find with brute force there is:

$$\Gamma = \{[111000]^T, [111000]^T\}, \gamma_s = [100100]^T$$

where γ_s corresponds to the spectral coordinate of $s_0 + s_1$ which is thus vulnerable. It can be shown (see supplemental material) that the average leakage conditioned to the sensitive variables is:

$$E\left[\left(L\left(z,3\right)+L\left(z,2\right)+L\left(z,1\right)+L\left(z,0\right)+\delta\right)^{2}\right] \tag{26}$$

$$=\sigma^{2}+\frac{E\left[(-1)^{s_1+s_0}\right]}{2}+\frac{83}{2} \tag{27}$$

which indeed depends on the transition of the sensitive variable S over the time interval.

5.2 Functional glitches in a non-linear circuit

The above theory of functional glitches can be extended to non-linear circuits such as those considered in Section 3.3. Let us consider a single non-linear function that emits a leakage through H_1:

$$L(z,t)=H_1(f(\lceil P_t z\rceil)).$$

We can check whether this function is vulnerable to functional glitches by checking if there is any set of spectral coordinates

$$\gamma_1,\dots,\gamma_p$$

for which the following expansion is different from zero:

$$\begin{aligned}
&\mathcal{F}[\textstyle\prod_i L(z,t_i))](\gamma_s)\\
&=\textstyle\sum_{a_1+\cdots+a_p=\gamma_s}\prod_{i=1}^{p}\hat{L}_{t_i}(a_i)\\
&=\textstyle\sum_{a_1+\cdots+a_p=\gamma_s}\prod_{i=1}^{p}\hat{L}_0(P_{t_i}^{-\top}a_i) && \text{[by Cor. 2]}\\
&=\textstyle\sum_{P_{t_1}^{\top}\gamma_1+\cdots+P_{t_p}^{\top}\gamma_p=\gamma_s}\prod_{i=1}^{p}\hat{L}_0(\gamma_i) && \text{[by subst.]}\\
&\propto\textstyle\sum_{P_{t_1}^{\top}\gamma_1+\cdots+P_{t_p}^{\top}\gamma_p=\gamma_s}\prod_{i=1}^{p}\mathcal{W}[f,\lceil\gamma_i\rceil] && \text{[by Th. 6]}\\
&\quad\wedge\ \forall\gamma_i,\lfloor\gamma_i\rfloor=0\\
&\neq 0
\end{aligned} \tag{28}$$

where $\hat{L}_0=\mathcal{F}[L(z,0)]$.

Example 6. To find a few more interesting vulnerabilities for the Keccak primitive considered in Section 4, let us analyze leakage L_a (see Eq. (22)) when only s_1 changes to s_1' over an interval characterized by two time slots (t_0 and t_1). The input transition vector is an 18-wide vector[12], while $P_{t_0}=I$ and P_{t_1} permutes $s_1\leftrightarrow s_1'$. The question we want to

[12] each leakage depends on 9 variables, the first 3 of which are the sensitive ones.

answer is whether L_a's value correlates with the change $s_1 \rightarrow s_1'$, i.e., with

$$\gamma_s = [\underbrace{0\overset{s_1}{1}0000000}_{\lceil\gamma_s\rceil}\underbrace{0\overset{s_1'}{1}0000000}_{\lfloor\gamma_s\rfloor}]^\top.$$

We know that the correlation with γ_s is not zero only if we find a pair of spectral coordinates (γ_1, γ_2) for which Eq. (28) is satisfied[13]. On the one hand, the subscript of the sum in Eq. (28) requires that any pair should satisfy:

$$\gamma_1 + P_{t_1}\gamma_2 = \gamma_s \;\; (P_{t_0} = I \text{ in this example}) \tag{29}$$

which, substituting γ_s and after some simple algebraic manipulation, can be rewritten as:

$$\gamma_1 = \gamma_2 \wedge \mathsf{bit}(\gamma_1, 1).$$

Both pair components should thus *i)* be equal and *ii)* have the second bit set (in the following we will use the symbol γ to refer to the same value $\gamma_1 = \gamma_2$). On the other hand, Eq. (28) is satisfied only if $\lceil\gamma\rceil$ belongs to the support of $\mathcal{W}[\chi_3 \circ Q_a]$ (see Corollary 3). As can be seen in Table 3, there are candidates for γ belonging to such support (those marked with the asterisk) which make Eq. (28) satisfiable, thus the considered leakage correlates with a transition on the sensitive variable s_1.

Table 3

Support of the Walsh transform of $\chi_3 \circ Q_a$. Rows with an asterisk correspond to the values of γ that comply with Eq. (29).

$\lceil\gamma\rceil$	$\mathcal{W}[\chi_3 \circ Q_a]$		$\lceil\gamma\rceil$	$\mathcal{W}[\chi_3 \circ Q_a]$	
00000100	1/4		00000101	1/4	
00000110	1/4		00000111	1/4	
00101100	1/4		00101101	1/4	
00101110	-1/4		00101111	-1/4	
01010100	-1/4	*	01010101	1/4	*
01010110	-1/4	*	01010111	1/4	*
01111100	1/4	*	01111101	-1/4	*
01111110	-1/4	*	01111111	1/4	*

[13] We look only for two coordinates because we have two time slots, i.e., $p = 2$.

Table 4
Vulnerabilities found for Keccak three shares. L_* is the leakage measured at t_0 while L'_* is the one measured at t_1. Note that the sum of both leakages is just the energy consumed in the whole time interval.

p	type	exposed variables	combining function
2	bi-variate	$s_1 + s'_1$	$(L_a + L'_a)^2$
2	bi-variate	$s_2 + s'_2$	$(L_a + L'_a)^2$
2	bi-variate	$s_0 + s'_0$	$(L_b + L'_b)^2$
2	bi-variate	$s_1 + s'_1$	$(L_b + L'_b)^2$
2	bi-variate	$s_2 + s'_2$	$(L_b + L'_b)^2$

Eventually, if we repeat the process for all the three shares, we find the vulnerabilities expressed in Table 4 which are all at the second order (as the smallest p we've found is 2) and concern only L_a and L_b.

6 Conclusions

In this brief, we have proposed a spectral model for reasoning symbolically about a countermeasure against side-channel attacks. The proposed framework allows one to detect, in a mathematically sound way, whether a general class of countermeasures holds up to a specific protection order. The symbolic treatment allowed us to derive precise combination functions useful for attacks, as well as confirm certain claims about a threshold implementation of Keccak. In addition, we have shown that the framework allows one to tackle the problem of glitch-based leakages, and we give some practical applications for this case. Further development of this work might include extending it to a broader set of nonlinear functions and to multiple changes of a signal during a time interval.

7 Acknowledgments

This project has been partly funded by EU's H2020 (grant n. 644052-HECTOR).

References

[1] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Building power analysis resistant implementations of Keccak," Tech. Rep., 2010.

[2] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards, 1st edition*. Springer Publishing Company, Incorporated, Oct. 2010.

[3] E. Bisi, V. Zaccaria, and F. Melzani, “Symbolic Analysis of Higher-Order Side Channel Countermeasures,” *IEEE Transactions on Computers*, pp. 1–7, Dec. 2016.

[4] A. G. Bayrak, F. Regazzoni, D. Novo, and P. Ienne, “Sleuth: automated verification of software power analysis countermeasures,” in *Cryptographic Hardware and Embedded Systems - CHES 2013*, Swiss Federal Institute of Technology, Lausanne. Berlin, Heidelberg: Springer, Aug. 2013, pp. 293–310.

[5] A. Moss, E. Oswald, D. Page, and M. Tunstall, “Compiler Assisted Masking,” in *Cryptographic Hardware and Embedded Systems - CHES 2012*. Berlin, Heidelberg: Springer, Sep. 2012, pp. 58–75.

[6] G. Barthe, F. Dupressoir, B. Grégoire, C. Kunz, B. Schmidt, and P.-Y. Strub, *EasyCrypt: A Tutorial*. Cham: Springer, 2014, pp. 146–166. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10082-1_6

[7] O. Reparaz, “Detecting flawed masking schemes with leakage detection tests,” in *Fast Software Encryption - FSE 2016*, 2016, pp. 204–222. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-52993-5_11

[8] T. Siegenthaler, “Correlation-immunity of nonlinear combining functions for cryptographic applications (Corresp.),” *IEEE Transactions on Information Theory*, vol. 30, no. 5, pp. 776–780, Sep. 1984.

[9] G.-Z. Xiao and J. L. Massey, “A spectral characterization of correlation-immune combining functions,” *IEEE Transactions on Information Theory*, vol. 34, no. 3, pp. 569–571, May 1988.

[10] R. O’Donnell, *Analysis of Boolean Functions*. Cambridge: Cambridge University Press, Jun. 2014.

[11] C. Carlet, “Boolean Functions for Cryptography and Error-Correcting Codes,” in *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Y. Crama and P. L. Hammer, Eds. Cambridge: Cambridge University Press, 2009, pp. 257–397.

[12] A. Moradi and O. Mischke, “How far should theory be from practice? Evaluation of a countermeasure,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Ruhr-Universitat Bochum, Bochum, Germany. Berlin, Heidelberg: Springer Berlin Heidelberg, Oct. 2012, pp. 92–106.

[13] B. Bilgin, J. Daemen, V. Nikov, S. Nikova, V. Rijmen, and G. Van Assche, “Efficient and First-Order DPA Resistant Implementations of Keccak,” in *Smart Card Research and Advanced Applications*. Cham: Springer, Cham, Nov. 2013, pp. 187–199.

[14] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, “Pushing the Limits - A Very Compact and a Threshold Implementation of AES.” *EUROCRYPT*, vol. 6632 LNCS, no. Chapter 6, pp. 69–88, 2011.

[15] S. Nikova, V. Rijmen, and M. Schläffer, “Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches,” *Journal of Cryptology*, vol. 24, no. 2, pp. 292–321, 2010.

[16] L. De Moura and N. Bjørner, “Z3: An Efficient SMT Solver,” in *Smart Card Research and Advanced Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 337–340.