IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2020, No. 4, pp. 24–48. DOI:10.13154/tches.v2020.i4.24-48

On the spectral features of robust probing security

Maria Chiara Molteni[1] and Vittorio Zaccaria[2]

[1] Dipartimento di Informatica "Giovanni Degli Antoni"
Università degli Studi di Milano, Italy
maria.molteni@unimi.it
ORCID: 0000-0003-2901-2972

[2] Department of Electronics, Information and Bioengineering
Politecnico di Milano, Italy
vittorio.zaccaria@polimi.it
ORCID: 0000-0001-5685-9795

Abstract.
In this work we provide a spectral formalization of non-interference in the presence of glitches. Our goal is to present new theoretical and practical tools to reason about robust-d-probing security. We show that the current understanding of extended probes lends itself to probes that participate, during gadget composition, to the creation of additional extended probes. In turn, this enables a natural extension of non-interference definitions into robust ones to build a new reasoning framework that can formally explain some semi-formal results already appeared in the past and be used to synthesize new robust-d-SNI gadgets.

Keywords: Robust probing security and robust strong non-interference for hardware gadgets - Boolean functions · Random variables · Correlation-immunity · Spectral characterization · Walsh transform

1 Introduction

This paper deals with the problem of protecting a hardware and software implementation against side channel attacks. Conventional countermeasures are widely based on masking [ISW03] but creating a masked implementation is not trivial at all, especially considering advanced adversary models such as probing adversaries or, more recently, glitch-extended probing adversaries [MBR19]. The most studied case is the one of the multiplication in $\mathbb{F}_2^n$, which, over the years, has been proven extremely tricky to be protected against newer attack models, let alone higher order ones. For this reason, the subject has never ceased to stimulate research since its inception [ISW03].

One of the main problems addressed is *composability*, i.e., determining, given two d-probing secure gadgets[1], if their functional composition is still d-probing secure. It has been proved that this depends on the amount of *refreshing*[2] that is used [CPRR14] but more is true; there are inner gadget properties, related to the concept of non-interference [BBD+16], which can be used to determine whether their composition is d-probing secure. One of them is *strong* non interference (d-SNI)[BBD+16] which requires that the number of input shares derivable from a certain set of probes depends only on the number of internal

[1] These are secure in the sense that given d probes, it is impossible to derive information about the secret values encoded in the masks/shares.

[2] The term refresh indicates a procedure that aims to bring back the secret's shares into a uniformly random state, after a series of operations that might have invalidated uniformity.

Received: 2020-04-15 Accepted: 2020-06-15 Published: 2020-08-26

positions present in that set (whenever that set's size is less or equal to d). Demonstrating that a gadget is in the first place d-SNI might require lengthy ratiocination or automatic tools [BBD+16, BGI+18, BGR18], but once it has been done, composition can be studied with much simpler reasoning. This is however easier said than done as, even recently, some gadgets that were thought to be d-probing secure have been shown to be vulnerable to higher orders [MMSS19].

Trying to protect the gadget also from circuit glitches puts the problem to a whole new level. The main tool used to protect against glitches is the *threshold implementation* (TI) [NRS11]. A *threshold implementation* aims at ensuring that the logic cones[3] of a primitive do not depend on all the shares. Besides the overall correctness constraints, TI essentially boils down to ensuring that, *i)* if a gadget's input is fed with shares (computed from the secret) whose distribution is uniform, its outputs must be uniform as well and *ii)* each output share must be computed with a subset of the input shares.

The current trend tries to define a conceptual groundwork above which d-probing security and glitches are considered as a single challenge instead of different, seemingly orthogonal problems. The major conceptual evolution with respect to the original d-probing security model, is the *robust probing model* [FGP+17, MBR19]. In this attack model, glitches are seen as extended probes that can be used by an attacker to observe the input values of a given cone of logic. With this model, one can prove that some gadgets (e.g., multiplication) are not only d-probing secure in the conventional sense but can be made robust-d-SNI by adding a register layer at the outputs (see for example [MBR19]). More recently, stricter conditions on a gadget have been traced, to ensure composability in presence of glitches (t-PINI condition) [CS20, CGLS20].

Chronologically, the original efforts considered a hybrid of the Ishai-Sahai-Wagner (ISW) scheme [ISW03] with TI, culminating in the Consolidated Masking Scheme [RBN+15] (CMS for short). While the results were important in terms of decrease of randomness needed (in CMS with $d+1$ shares, one needs $(d+1)^2$ refresh values) it was shown recently that this cannot be extended past $d > 2$ (without even considering robust d-probing security [MMSS19]). Later proposals for a d-probing secure multiplication addressed a reduction in terms of refresh values [GMK17, GM18] (with a lower bound identified in [BBP+16]) but, after the considerations made in [MMSS19], it is not clear how much past $d > 1$ these can be made robust-d-probing secure, let alone robust-d-SNI without an output register[4].

1.1 Our contribution

This work revisits d-probing security fundamentals by providing a spectral formalization of non-interference that encompasses recently introduced advancements such as *robust d-probing security* [MBR19, MMSS19]. The overarching goal is to give an alternative yet comprehensive view of the problem which might be more amenable to proof mechanization, in the same vein as [BGI+18, MMSS19]. We thus take a detour from conventional information theoretical considerations (see, for example, [MBR19]) for a more algebraic approach which exploits the characterization of the spectrum of vector Boolean functions and its connections with correlation immunity [XM88, ZMB18]. Our approach aims to be more foundational than other approaches based on spectral characterization which are based on approximations and do not encompass composability [BGI+18]. In this sense, we derive formal conditions for d-probing security in the presence of glitches by further categorizing probes (e.g., *pure* vs *composed*) to enable compositional reasoning of vulnerability profiles. More importantly, we have found that, to conciliate with composability, the nature of an extended probe must afford an additional distinction, i.e., output vs internal, where output

[3] A logic cone is the whole of operations and inputs needed to compute one output (or one output's share).

[4] We call output register a register in which the function's outputs are recorded.

probes participate, during composition, in the creation of additional extended probes while internal do not. We thus discovered a new definition of robust non-interference which complies with existing observations in literature but has, from our point of view, a more intuitive meaning.

To corroborate the usefulness of our approach, we will show that the underlying tensor calculus is useful to reason formally about both conventional and robust d-probing security by giving new meaning to some results already appeared in the past [FGP+17]. On the other side, we show that it can enable the exploration of the design space of known gadgets by deriving an improved *consolidated masking scheme* [RBN+15] which is robust-3-probing secure and robust-3-SNI without the need of an additional register at the output (compared to [FGP+17]). While this is done only for $d = 3$, we are able to derive sufficient conditions for making a generalized CMS scheme into a robust-d-SNI one. Finally, we conclude with some deductions around the DOM multiplication scheme [GMK16] that can be made with our framework.

Before starting this research endeavor we felt that there was a lack of mathematical definitions of robust strong non-interference. This concern was raised before in the community; recently, in a paper published on TCHES [MBR19], the authors recognized that despite the existence of the concept of robust SNI, it remained unclear how to automate the verification of composability of hardware gadgets, as it was unclear how to define a single mathematical equation. They acknowledge that there is still room for more automated ways to reason about robust non-interference. To understand how our approach fills this gap, we would like to highlight how our work can benefit the community from the both the *research* and *development* standpoints.

The research standpoint. As it is known, one of the main goals of any research endeavor is to build inference rules to derive general solutions to common problem patterns. This is distinguished from solving those problems with an instance-by-instance approach or a tool. To show how our approach can be used for deriving such general rules, we refer the reader to Appendix A, where we present some general conclusions about the robust probing security of a common pattern found in cryptography, for any number of shares.

The development standpoint. Existing tools such as maskVerif [BBFG18] can be helpful in verifying if a fixed configuration instance of a gadget is probing secure or strong non-interferent; we call these *instance-by-instance* tools. Our approach can be used also on an instance-by-instance basis. More importantly, notwithstanding the efficiency of maskVerif, its developers argue that "more precise approaches remain important, when verification with more efficient methods fail"[BBFG18]. Given that our approach is not based on a syntactic model but on the exact theory of Boolean functions, it is probably the first to fit this purpose as previous works have only provided approximations [BGI+18] or partial solutions [MBR19]. We note that our approach provides the added benefit of a linear algebra based approach which is supported by many mathematical toolboxes. However, given the exponential size of correlation matrices, some analysis of computational complexity is in order. We refer the reader to Appendix B for an estimate of the time needed for computing the vulnerability profile for several known gadgets.

2 Probing security as a relation calculus

The methodology that we propose is heavily based on the Walsh transform of a vectorial boolean function. Besides introducing conventional concepts around it, we introduce the definition of the tensor product for the resulting matrices [Car10].

Definition 1 (Walsh transform of a vectorial function)**.** Given a vectorial Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, we define its Walsh transform as a $2^m \times 2^n$ matrix $\widehat{f}$ whose elements are:

$$\widehat{f}_{\omega,\alpha} = \sum_{x \in \mathbb{F}_2^n} (-1)^{\omega^\intercal f(x) \oplus \alpha^\intercal x} \tag{1}$$

$\omega \in \mathbb{F}_2^m, \alpha \in \mathbb{F}_2^n$ being the binary encoding of the row and column indices, called *spectral coordinates* (or sometimes *masks*).

As it is known, the Walsh transform describes the correlation information between input variables' XOR-combinations and the corresponding output ones. Thus, they appear in the literature scaled by a coefficient 2^{-n}, under the name of *correlation matrices* [DGV95]:

$$W_f = 2^{-n} \widehat{f}$$

For correlation matrices, the following theorem is known to hold:

Theorem 1 (Correlation matrix as a map of probability distributions)**.** *Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and a probability distribution $p_X : \mathbb{F}_2^n \to \mathbb{R}$ for its input variable, the following holds:*

$$W_f F_{p_X} = F_{p_Y}$$

where p_Y is the distribution of the output values while F_g is the Fourier transform of any pseudo-boolean function $g : \mathbb{F}_2^n \to \mathbb{R}$ and defined as the following:

$$F_g(\gamma) = \sum_{x \in \mathbb{F}_2^n} g(x)(-1)^{\gamma^\intercal x}$$

(see [DGV95, DPGM16]).

Thus, the correlation matrix W_f of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is just a linear map $\mathbb{P}_n \to \mathbb{P}_m$ ($\mathbb{P}_x \subset \mathbb{R}^x$) that is endowed with composition:

Theorem 2 (Composition of correlation matrices)**.** *Given two functions $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $g : \mathbb{F}_2^m \to \mathbb{F}_2^q$, the following holds:*

$$W_{g \bullet f} = W_g W_f$$

Moreover, if f is a bijection, $W_{f^{-1}} = W_f^{-1}$. For a proof see [Car10, PGM11].

Given two independent variables $x_f \in \mathbb{F}_2^{n_f}$ and $x_g \in \mathbb{F}_2^{n_g}$, we can form the probability distribution of the vector $[x_f, x_g]$ with the product of distributions for which the following theorem can be proved:

Theorem 3 (Tensor product of correlation matrices)**.** *Given two functions $f : \mathbb{F}_2^{n_f} \to \mathbb{F}_2^{m_f}$ and $g : \mathbb{F}_2^{n_g} \to \mathbb{F}_2^{m_g}$, the correlation matrix of the function $h([x_f, x_g]) = [f(x_f), g(x_g)]$ is $W_h = W_g \otimes W_f$ where the symbol $\otimes$ is the Kronecker product (or tensor product) of matrices. We will say that W_h is a mapping from the space $\mathbb{P}_{n_f} \otimes \mathbb{P}_{n_g}$ to the space $\mathbb{P}_{m_f} \otimes \mathbb{P}_{m_g}$ where $\otimes$ is understood to be the tensor product of finite dimensional vector spaces over the reals* $\mathbf{FdVect}_{\mathbb{R}}$.

We use a graphical language known as *string diagrams* [Sel10] to reason intuitively on the effect of composing correlation matrices. The underlying assumption is that any equational statement derivable with the string diagram can be derived if and only if it is symbolically derivable from the axioms of the theory. Among the features of this language, we have that:

- each correlation matrix is drawn as a box (except for identities which are drawn as simple wires),

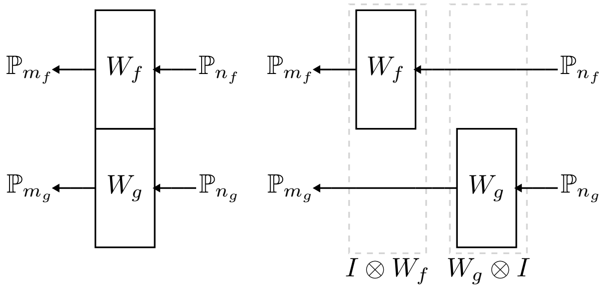


Figure 1: Example of compositional equality derived through a string diagram.

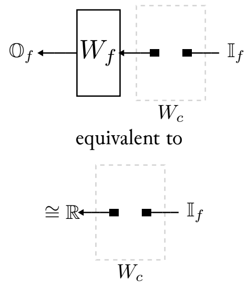


Figure 2: Compositional equality involving constant functions.

- composition is the horizontal juxtaposition,
- tensor product is the vertical one.

Figure 1 shows an example of compositional equality derived through a string diagram. The diagram on the left corresponds to the product $(W_g \otimes W_f)$ while the one on the right corresponds to $(1 \otimes W_f)(W_g \otimes 1)$(each factor is highlighted with a dotted box). Note that the underlying matrices have the property that simply moving boxes without crossing wires does not change the underlying formulas, i.e., $(W_g \otimes W_f) = (1 \otimes W_f)(W_g \otimes 1)$. Moreover, there always exist two mappings $B_{a,b} : \mathbb{P}_a \otimes \mathbb{P}_b \to \mathbb{P}_b \otimes \mathbb{P}_a$ and $B_{b,a}$ such that $B_{a,b}B_{b,a} = I$. $B_{a,b}$ is exactly the Walsh transform of a function that permutes variables a and b. A constant function c is such that its correlation matrix W_c is zero everywhere else than in its first column; in this case the image of W_c is always isomorphic to the base field $\mathbb{R}$ which is the unit of the tensor product of vector spaces in $\mathbf{FdVect}_{\mathbb{R}}$. Diagrammatically, we can show a constant function as a circuit breaker symbol that conveys figuratively some of the equalities about the unit of $\mathbf{FdVect}_{\mathbb{R}}$ (see for example Figure 2).

As correlation matrices can become exponentially big, we propose a compact representation for W_f:

$$\widetilde{W_f}(i,j) := (W_f(i,j) \neq 0) \tag{2}$$

which is shown in the following example.

Example 1. Consider a function $f : \mathbb{F}_2^4 \to \mathbb{F}_2^3$

$$f(a_0, a_1, r_0, r_1) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} a_0 + r_0 + r_1 \\ a_1 + r_0 + r_1 \\ a_1 + r_0 \end{bmatrix}$$

From its correlation matrix, we can derive through Eq. 2 the following relation matrix $\widetilde{W_f}(\phi, \psi)$ $(\phi = [\gamma_{f_2}\gamma_{f_1}\gamma_{f_0}], \psi = [\gamma_{r_1}\gamma_{r_0}\gamma_{a_1}\gamma_{a_0}])$:

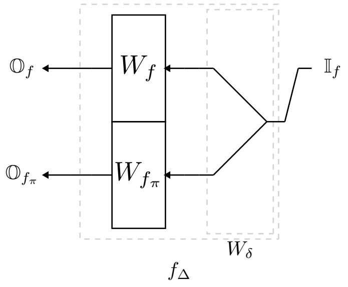


Figure 3: The vulnerability profile of a function corresponds to the tensor product of the regular Walsh transform of a function and of its probes f_π, multiplied by W_δ.

$$\begin{array}{cccccccccccccccccccc}
&&&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&\gamma_{r_1}\\
&&&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&\gamma_{r_0}\\
&&&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&\gamma_{a_1}\\
&&&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&\gamma_{a_0}\\
\gamma_{f_2}&\gamma_{f_1}&\gamma_{f_0}&&&&&&&&&&&&&&&&&\\
0&0&0&1&&&&&&&&&&&&&&&&\\
0&0&1&&&&&&&&&&&&&&1&&&\\
0&1&0&&&&&&&&&&&&&&&1&&\\
0&1&1&&&&1&&&&&&&&&&&&&\\
1&0&0&&&&&&&1&&&&&&&&&&\\
1&0&1&&&&&&&&&&&&1&&&&&\\
1&1&0&&&&&&&&&1&&&&&&&&\\
1&1&1&&&&&&&&1&&&&&&&&&
\end{array} \tag{3}$$

The above representation labels the columns (rows) with the corresponding combination of inputs (outputs) in binary form. As an example, the element $\widetilde{W_f}([011],[0011])$ (which is 1) represents an existing dependency between $f_0 \oplus f_1$ and $a_0 \oplus a_1$. Besides, note that we could write the index $([011],[0011])$ as $(3,3)$ by interpreting it in base 2. To derive a compact representation for the above relation matrix, we note that shares of the same variable can be grouped; for example, a_0 and a_1 could be two shares of a single sensitive variable a, r_0 and r_1 two random values, f_0 and f_1 are two shares of a single output value o and f_2 is the output associated with a potential internal probe p within the circuit realization of f.

These assumptions allow us to re-structure the original correlation matrix by compacting the spectral coefficients to account only for the number of shares of each original variable; we call this the *compact representation of a correlation matrix*:

$$\begin{array}{cccccccccccc}
&&0&0&0&1&1&1&2&2&2&\rho\\
&&0&1&2&0&1&2&0&1&2&\alpha\\
\pi&\omega&&&&&&&&&&\\
0&0&1&&&&&&&&&\\
0&1&&&&&&&&1&&\\
0&2&&&1&&&&&&&\\
1&0&&&&&1&&&&&\\
1&1&&&&1&&1&&&&\\
1&2&&&&&1&&&&&
\end{array} \tag{4}$$

where α, ρ, ω and π are called the *compact spectral indexes* of the input, randoms, output and probe respectively.

2.1 The vulnerability profile of a function

Figure 3 shows a typical wiring diagram of the mapping betwen the Fourier transform of its input and output distributions. In particular, it is related to the boolean function f and its potential probes f_π. The tensor product

$$f_\Delta = (W_{f_\pi} \otimes W_f) W_\delta \tag{5}$$

encodes all the vulnerability data associated to f; in practice, each row of this matrix corresponds to a convolution of a combination of rows in W_f and in W_{f_π} and we know that, if there is some input variable combination for which this convolution is not zero, we have a dependency between a combination of outputs (either outputs of f or its probes) and a subset of input variables [ZMB18, XM88]. We call this data *the vulnerability profile of f*. It is a special case of *fan*, a notion that will be useful to deal with glitches and extended probes as well:

Definition 2 (fan of a family of matrices)**.** The *fan* of a family of matrices $M = \{M_i\}_{i=1\dots n}$ is a matrix:

$$\Delta M = (\bigotimes_i M_i) W_\delta^{n-1}$$

where W_δ is the correlation matrix associated with the duplication function.

2.2 Composition of vulnerability profiles

It is possible to derive the vulnerability profile of a composition of two functions by studying the composition of two fans:

$$k_\Delta \bullet h_\Delta = \Delta\{W_{h_\pi}, W_{k_\pi h}, W_{kh}\}$$

which is the fan of the composition of the original functions:

$$k_\Delta \bullet h_\Delta = (k \bullet h)_\Delta$$

and it is possible to show that it is associative; figure 4 shows the string diagram associated to it where we exploited tensor product equivalences to create a compact yet equivalent representation.

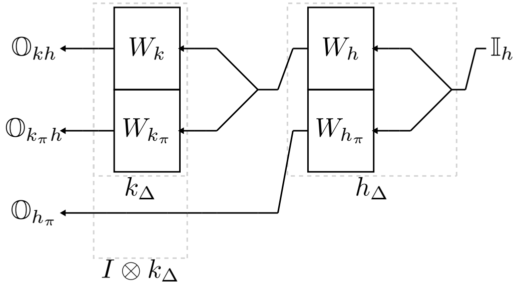


Figure 4: The composition of two vulnerability profiles as a map in the probability space.

This way of modeling vulnerability allows to reason around d-probing security and d-non-interference in a composable way. Recall that a function f is d-non interferent (d-NI) if, when given a total of s outputs and internal probes, $s \leq d$ implies a dependency with maximum s input shares. A function f is strongly d-non interferent (d-SNI) if $s \leq d$ implies a dependency with maximum i input shares, where i is the number of internal probes, among those placed [BBD+16].

Let us for example reconsider a case discovered in [CPRR14] that proves that, in general, the composition of d-NI and d-SNI functions is not d-NI. Figure 5 shows the structure of a function h which is a composition of two functions f and g; the assumptions are that f is d-NI and g is d-SNI. In particular, f refreshes its input a with two random bits r_f:

$$o_f(a_0, a_1, a_2, r_0, r_1) = [a_0 \oplus r_0 \oplus r_1, a_1 \oplus r_0, a_2 \oplus r_1]$$

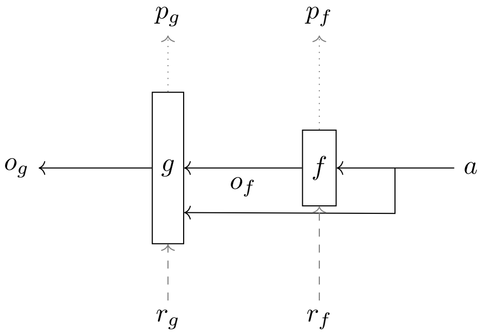


Figure 5: The composition pattern of f (d-NI) and g (d-SNI) derived from [CPRR14].

and it is assumed to have been probed at location $p_f = a_0 \oplus r_0$. On the other hand, $g(a, b, r_g)$ is the ISW multiplication [ISW03] which consumes 3 random bits r_g for the secret computation. Also in this case, it is assumed a single probe $p_g = a_2 \wedge b_1$.

The string diagram in Figure 4 can describe the vulnerability profile of the circuit by considering $h(a, r_f, r_g) = [f(a, r_f), (a, r_g)]$ and $k(a, r_f, r_g, o_f) = g(a, o_f, r_g)$ where the space of the input distributions is $\mathbb{I}_h = \mathbb{A} \otimes \mathbb{R}_f \otimes \mathbb{R}_g$ while for output distributions we have $\mathbb{O}_{kh} = \mathbb{O}_g, \mathbb{O}_{k_\pi h} = \mathbb{P}_g, \mathbb{O}_{h_\pi} = \mathbb{P}_f$.

Figure 6 shows the compact representation of the vulnerability profile. First of all, we are interested only in the first 4 columns, as these are the ones that represent relationships between the outputs and the shares of a not masked by any random value. We note that there is a potential dependency in row $[1, 1, 0]$, column $[0, 0, 3]$, exactly the one found in [CPRR14], which says that one needs only two probe values to get three shares; h is thus not even 2-NI, showing that d-NI and d-SNI do not compose into a d-NI function. It is possible to show through the compact representation of vulnerability profiles that, for this composition pattern, if f is d-SNI and g is d-NI (d-SNI) then the composition is d-NI (d-SNI).

2.3 Extended probes

Extended probes change the attack model in the sense that they allow the attacker to observe all the inputs of a gadget by probing its output wires. We will show that the fan linear algebra introduced above is still suitable for computing probing security profiles with a little more sophistication. In part this is due to the fact that one has to model in a composable way the information flow from inputs to outputs. Before going into the details let us classify the probes used in this model (we will drop the term *extended* as it is implicit in this discussion and we will introduce some symbol-coding to identify probes):

- a *pure* probe (notation symbol $\circ$) w_π over a wire computing the combinatorial function $w(x)$, modeled as a Boolean function that has a *stable* non-zero correlation with all the inputs of w (and their combinations). A stable correlation means that *any* transient effect is observable through that probe[5]. We will exploit the spectral characteristics of the *and* operator (which has a non-zero correlation with all of its operands and their combination) to model such probes:

$$w_\pi(x) = \bigwedge_{x_i \in \text{support}(w)} x_i$$

[5] This definition works only for combinatorial functions. For a register, instead, any pure probe on its output will have zero correlation with its inputs.

$$
\begin{array}{ccc|cccccccccc|l}
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots\rho_g \\
 & & & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & \dots\rho_f \\
 & & & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & \dots\alpha \\
\pi_f & \pi_g & \omega_g & & & & & & & & & & & \\
0 & 0 & 0 & 1 & & & & & & & & & & \\
0 & 0 & 1 & & & & & & & & & & & \\
0 & 0 & 2 & & & & & & & & & & & \\
0 & 0 & 3 & 1 & & & 1 & & & & & & & \\
0 & 1 & 0 & 1 & 1 & & & 1 & 1 & 1 & & 1 & 1 & \\
0 & 1 & 1 & & & & & & & & & & & \\
0 & 1 & 2 & & & & & & & & & & & \\
0 & 1 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 0 & 0 & & & & & & 1 & & & & & \\
1 & 0 & 1 & & & & & & & & & & & \\
1 & 0 & 2 & & & & & & & & & & & \\
1 & 0 & 3 & & & & & & 1 & 1 & & & & \\
1 & 1 & 0 & & 1 & 1 & 1 & 1 & 1 & 1 & & & 1 & \\
1 & 1 & 1 & & & & & & & & & & & \\
1 & 1 & 2 & & & & & & & & & & & \\
1 & 1 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 &
\end{array}
\tag{6}
$$

not d-NI

Figure 6: Vulnerability profile of [CPRR14] (we use greek letters to indicate the spectral coordinate associated with each function variable, i.e., α is the spectral coordinate associated with variable a and so on). Gray areas indicate where the composition $g \bullet f$ is allowed to have non-zero values to meet d-NI hypotheses.

- a *composed* probe (notation symbol ∞) w_κ over a wire computing $w(x) = (w^a \bullet w^b)(x)$ is a probe that can be factored into a pure probe over the intermediate values of w:

$$w_\kappa(x) = (w^a_\pi \bullet w^b)(x)$$

  where $w^b(x)$ is different from the identity.

Probes might be orthogonally classified in *output* probes and *internal* ones; this orthogonal characterization is relevant when talking about the composition of blocks:

- potential *output* probes (notation symbol $\uparrow$) are grouped in sets whose size corresponds to the actual outputs of the function. Among them we will distinguish one set of pure probes and zero or more sets of composed probes. We will use the symbol ω to indicate the overall number of sets ($\omega \geq 1$). Output probes are important during composition of functions because they will produce new probes (either pure or composed).
- *Internal* probes might be pure or composed; compared to output ones, these will not produce new probes when composing functions but will participate in the computation of the probing profile of the result.

In this context, we define an extended fan that encompasses the probes of the function

$$f_\nabla = \Delta\{\underbrace{\overset{\uparrow\circ}{W}_{f_\pi}, \overset{\uparrow\infty}{W}_{f^1_\kappa}, \dots, \overset{\uparrow\infty}{W}_{f^{\omega-1}_\kappa}}_{\text{output}}, \underbrace{\overset{\circ|\infty}{W}_{f^1_i}, \dots, \overset{\circ|\infty}{W}_{f^\nu_i}}_{\text{internal}}, \overset{\uparrow}{W}_f\}$$

Provided that one has all the matrices involved, f_∇ describes the overall security profile of the function. An important observation (which will be useful later) is that if one considers a register r there is a single set of pure output probes that one can build, i.e., constant ones.

Table 1: Algebraic composition rules for probes.

g	f	g • f
↑	↑	↑
↑⚭	↑	↑⚭
↑○	↑	↑⚭
↑○	↑⚭	↑⚭
↑○	↑○	↑○
○	↑	⚭
○	↑⚭	⚭
○	↑○	○
-	○, ⚭	○, ⚭

$$r_{\nabla} = \Delta\{\overset{\uparrow\circ}{W_c}, \overset{\uparrow}{I}\} \tag{7}$$

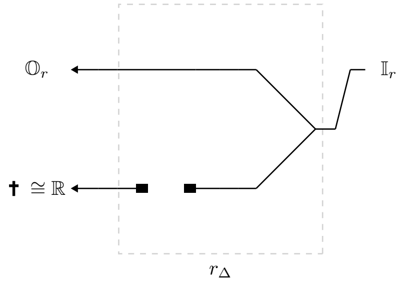


Figure 7: The vulnerability profile of a register in terms of maps over the Fourier transform of input and output distributions.

This will have an important implication because, when computing the composition of blocks, the zero matrix will become a circuit breaker, essentially forcing all successive functions to map to it as well. Let us consider the composition of two fans:

$$g_{\nabla} \bullet f_{\nabla} = h_{\nabla}$$

For it to be associative, the new fan h_{∇} will be such that:

- its *internal* probes (○ or ⚭) will be all those internal to f plus those produced by composing:
  - internal probes in g (○ or ⚭) with f's outputs and
  - internal pure probes in g (○) with output probes in f (↑⚭ or ↑○).
- its *output* probes (↑⚭ or ↑○) will be generated by combining
  - pure output probes in g (↑○) with f's outputs and its output probes (↑⚭ or ↑○).
  - composed output probes in g (↑⚭) with f's outputs.

Table (1) shows all the composition rules.

Example 2. Assume that both f and g have only a set of pure (output) probes:

$$f_{\nabla} = \Delta\{\overset{\uparrow\circ}{W}_{f_\pi}, \overset{\uparrow}{W}_f\}$$

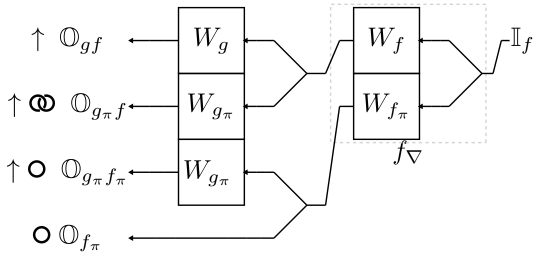


Figure 8: The vulnerability profile of a composition of functions when considering extended probes.

$$g_\nabla = \Delta\{\overset{\uparrow\circ}{W}_{g_\pi}, \overset{\uparrow}{W}_{g}\}$$

then, $g_\nabla \bullet f_\nabla$ will be

$$g_\nabla \bullet f_\nabla = \Delta\{\overset{\uparrow\circ\!\circ}{W}_{g_\pi f}, \overset{\uparrow\circ}{W}_{g_\pi f_\pi}, \overset{\circ}{W}_{f_\pi}, \overset{\uparrow}{W}_{gf}\}$$

Diagrammatically, one could picture the above vulnerability profile as in Figure 8. If we compare this with the non-extended case (Figure 4), we see an additional pure output probe whose correlation matrix is

$$\overset{\uparrow\circ}{W}_{g_\pi f_\pi}$$

This probe practically connects the outputs of the resulting vulnerability profile to the inputs of f (with maximum correlation).

Example 3. Let us now consider the case where, between g and f, we put a register r. The pure composition of these three blocks is shown in Figure 9. However, if we consider the vulnerability profile of the register (Figure 7), we get a more explicative diagram in Figure 10 which faithfully translates into correlation matrices and corresponding Fourier transform of the probability distributions. In practice, the probes are isomorphic to the one that would be produced by

$$g_\nabla \bullet r_\nabla \bullet f_\nabla = \Delta\{\overset{\uparrow\circ\!\circ}{W}_{g_\pi f}, \overset{\circ}{W}_{f_\pi}, \overset{\uparrow}{W}_{gf}\}$$

i.e, to the composition of vulnerability with probes acting as regular probes, not extended ones.

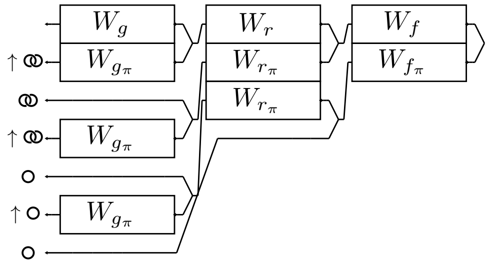
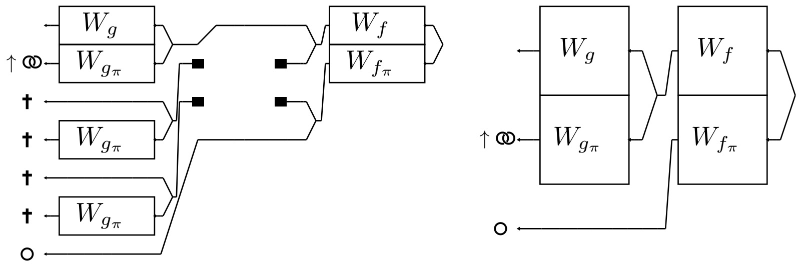


Figure 9: The vulnerability profile of a composition of three functions when considering extended probes.



(a) (b)

Figure 10: (a) shows the vulnerability profile of a composition of two functions when a register is considered in the middle. Probes that come after the "circuit breaker" map to the unit of $\mathbf{Vect}_{\mathbb{R}}$ and thus do not add any information so they have been drawn with a white circle. The Fourier transform of the output distribution is isomorphic to the one produced by the diagram in (b).

2.4 Definition of robustness

Given a vulnerability profile f_∇, we propose the following robustness definitions.

Definition 3 (robust-d-probing-secure vulnerability profile)**.** A vulnerability profile f_∇ is d-probing secure when given a total of d outputs (either conventional or output probes) and internal probes (either composed or pure), there is no dependency with all the shares of a secret.

Definition 4 (robust-d-NI vulnerability profile)**.** A vulnerability profile f_∇ is robust-d-NI when given a total of s outputs (either conventional or output probes) and internal probes (either composed or pure), $s \leq d$ implies a dependency with maximum s input shares.

Definition 5 (robust-d-SNI vulnerability profile)**.** A vulnerability profile f_∇ is robust-d-SNI when given a total of s outputs (either conventional or output probes) and internal probes (either composed or pure), $s \leq d$ implies a dependency with maximum i input shares, where i is the number of internal probes.

Example 4. To show how the above definition of robust-d-probing security matches with the existing understanding, let us rederive the considerations exposed in [FGP$^+$17, MBR19] concerning the compositionality of a second-order secure multiplier when considering glitches. The example considers inputs (x_0, x_1, x_2) and (y_0, y_1, y_2) and consists of two stages separated by a register r; the first stage (let us call it f) contains 9 products $x_i y_j$ some of them (cross-domain products) are remasked:

$$\begin{array}{lll} f_{0,0} = x_0 y_0 & f_{0,1} = x_0 y_1 \oplus r_1 & f_{0,2} = x_0 y_2 \oplus r_2 \\ f_{1,0} = x_1 y_0 \oplus r_1 & f_{1,1} = x_1 y_1 & f_{1,2} = x_1 y_2 \oplus r_3 \\ f_{2,0} = x_2 y_0 \oplus r_2 & f_{2,1} = x_2 y_1 \oplus r_3 & f_{2,2} = x_2 y_2 \end{array} \tag{8}$$

The second stage (let us call it g) compresses the triplets:

$$\begin{array}{l} g_0 = f_{0,0} \oplus f_{0,1} \oplus f_{0,2} \\ g_1 = f_{1,0} \oplus f_{1,1} \oplus f_{1,2} \\ g_2 = f_{2,0} \oplus f_{2,1} \oplus f_{2,2} \end{array} \tag{9}$$

The question is whether outputs $g_0 \dots g_2$ should be saved into a register s to preserve composability in the sense of robust-d-probing security. We thus compute the two vulnerability profiles:

$$s_\Delta \bullet g_\Delta \bullet r_\Delta \bullet f_\Delta = \Delta\{\overset{\circ}{W}_{f_\pi}, \overset{\infty}{W}_{g_\pi f}, \overset{\uparrow}{W}_{gf}\} \tag{10}$$

and

$$g_\Delta \bullet r_\Delta \bullet f_\Delta = \Delta\{\overset{\circ}{W}_{f_\pi}, \overset{\uparrow\infty}{W}_{g_\pi f}, \overset{\uparrow}{W}_{gf}\} \tag{11}$$

We note that, when the register s is not present, probe $g_\pi f$ is an output probe ($\uparrow\infty$) and will participate in creating new probes in the following compositions. Instead, when the register s is present (Eq. 10), the *outputs* are only the conventional outputs of $g \bullet f$ while $g_\pi f$ is just an internal composed probe (∞).

Considering again Eq. 11, if we take just one output in $g_\pi \bullet f$ (e.g., the wire z_0) and no internal probes ($i = 0$), one would get a dependency with $t_{0,0}$ which in turn depends[6] on one share of x and y. This shows that the case without output register is not robust-d-SNI because, for $i = 0$, there should not be any dependency over input shares. Note that this observation has already been done in the past [FGP+17]; however, we argue that ours is one the first attempts to formalize this point mathematically.

Before closing we note that, in a general case such as Eq. 11, one has always $W_{gf} \preceq W_{g_\pi f}$ because extended probes over g are always more powerful of g itself; in this case, robust-d-probing security is thus determined by W_{f_π} and $W_{g_\pi f}$ alone:

$$g_\Delta \bullet r_\Delta \bullet f_\Delta = \Delta\{\overset{\circ}{W}_{f_\pi}, \overset{\infty\uparrow}{W}_{g_\pi f}\} \tag{12}$$

we will exploit this consideration in the following sections.

3 Revisiting the probing security of CMS

The acronym CMS stems from the title of the proposing article [RBN+15] and identifies an evolution of the ISW scheme [ISW03] meant to provide, at the same time, d-probing security and protection against glitches by borrowing ideas from the TI scheme [NRS11].

A CMS scheme with $s = 4$ shares is organised as in Figure 11. Every output share c_i is computed in a *logic cone* which involves s pairs $(a_i, b_h), h \in \{0 \ldots s-1\}$; *adjacent* cones share only a random bit while internal bits within a cone preserve uniformity, as is usual in a TI scheme. The computation is typically decomposed in three layers: non-linear ($\mathcal{N}$), refresh ($\mathcal{R}$) and compression ($\mathcal{C}$), the latter two separated by a register to mitigate the propagation of glitches to the outputs.

While the original proposal identified a scheme that was d-probing secure up to $d = 2$, a simple generalization of the scheme to $d = 3$ has shown that, as it is, it cannot be made probing secure anymore [MMSS19]. Figure 11 shows the scheme for $d = 3, s = 4$ and a triplet of probes that reveals the four shares of b; note that this vulnerability exists even if one does not consider extended probes.

Considering robust probing security, we can say something more. First of all, note that we are in the case covered by Eq. 12 where:

$$f = \mathcal{R} \bullet \mathcal{N}, g = \mathcal{C}$$

We thus know that the only probes that determine robust-d-probing security are:

- the pure probes at the output of the refresh layer, i.e., f_π;
- the composed probes at the output of the compression layer $g_\pi \bullet f$.

[6] Recall that $a \wedge b$ is correlated with both a, b and $a \oplus b$, as it correlation matrix shows.

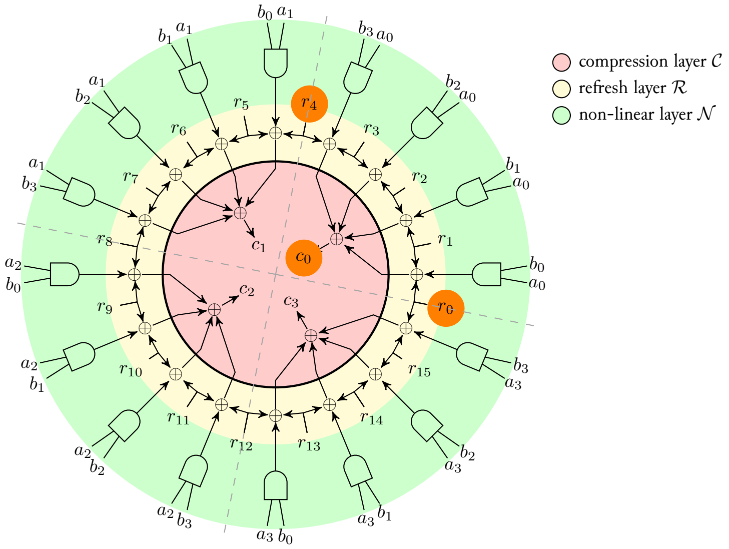


Figure 11: The four-share CMS scheme considered in [MMSS19]. The scheme is decomposed in three layers, non-linear ($\mathcal{N}$), refresh ($\mathcal{R}$) and compression ($\mathcal{C}$). To preserve output shares from the propagation of glitches, a register (thick line) layer is inserted between compression and refresh. Orange circles correspond to regular probes that break the d-probing security.

Composed probes are just four and the number of shares that they cover is important to determine probing security; to show this, let us assign them a label:

$$S_c = \{c_0, c_1, c_2, c_3\}$$

and show, in a table, which pairs (a_i, b_h) are covered by which extended probe:

	b_0	b_1	b_2	b_3
a_0	c_0	c_0	c_0	c_0
a_1	c_1	c_1	c_1	c_1
a_2	c_2	c_2	c_2	c_2
a_3	c_3	c_3	c_3	c_3

Note that, given one of these output probes, e.g., c_0, one needs to recover only the two random bits that separate it from adjacent cones $c_1$and c_3. These two bits can be derived only by using just two pure probes of f_π.

It has been observed that *non-completeness* might be useful in this case to reach robust-d-probing security for $d = 3$ (see [MMSS19]). To find it, we note that a combination[7] of output probes $\Pi \in \mathcal{P}(S_c)$ is such that it needs R_Π additional randoms depending on adjacency. For example, $\Pi = \{c_0\}$ would need $R_\Pi = 2$ pure probes in f_π, while $\Pi = \{c_0, c_2\}$ would need $R_\Pi = 4$ because they form two partitions in terms of adjacency; in fact, given T_Π as the number of such partitions we have $R_\Pi = 2T_\Pi$.

The organization of the input shares can be seen as a surjective mapping from the set of pairs of input shares to the set of output cones:

$$\lambda : S_a \times S_b \to S_c$$

[7] With the symbol $\mathcal{P}(S_c)$ we denote the power set of S_c

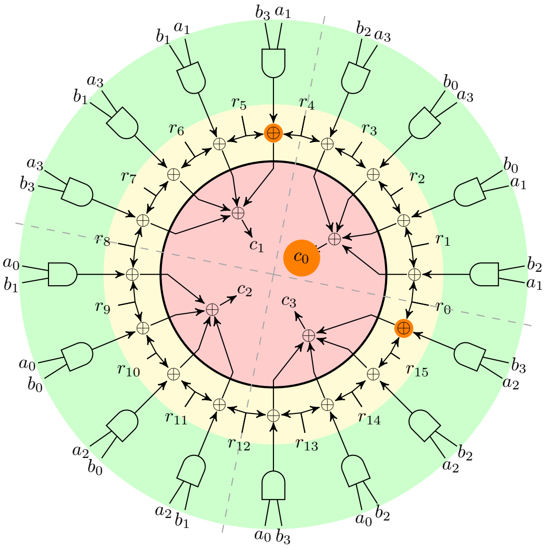


Figure 12: The robust-3-probing secure CMS scheme found with our formalization. Highlighted in orange the probes which make the above scheme not robust-3-SNI.

Define $\delta_a^\lambda(\Pi)$ as the maximum number of shares of a that a specific probe configuration Π covers ($\delta_b^\lambda(\Pi)$ is analogously defined). To find a configuration that is robust-3-probing secure we can state the following problem:

> *Find a mapping λ such that, if the total number of probes is less or equal to three, the number of shares that one can get is always less or equal to three, i.e.:*
>
> $$\begin{aligned} \forall \Pi \in \mathcal{P}(S_c). \quad & |\Pi| + R_\Pi \leq 3 \implies \delta_a^\lambda(\Pi) \leq 3 \quad \wedge \\ & |\Pi| + R_\Pi \leq 3 \implies \delta_b^\lambda(\Pi) \leq 3 \end{aligned}$$

We formalized the problem as an *satisfiability modulo theory* one and solved it through Microsoft's z3 SMT solver. The solver provided the following solution λ (also depicted in Figure 12):

	b_0	b_1	b_2	b_3
a_0	c_2	c_2	c_3	c_3
a_1	c_0	c_1	c_0	c_1
a_2	c_2	c_2	c_3	c_3
a_3	c_0	c_1	c_0	c_1

We verified this solution by computing the vulnerability profile as in Eq. 12. We computed the underlying correlation matrices W_{f_π} and $W_{g_\pi f}$ by using a sparse representation while the complete fan is computed by convolving the rows of the above two matrices. The used sparse representation of the correlation matrix is a modified version of a List of Lists representation (LIL): each stored list refers to a specific row of the correlation matrix, and the elements of every list are the column coordinates of the nonzero element in the correlation matrix row. We do not need to store the value of nonzero elements, because the presence of this nonzero elements is the only thing that matters. To give an idea of the space required for correlation matrices for $d = 3$, W_{f_π} is a $2^{16} \times 2^{28}$ correlation matrix with less than 2^{24} elements different from 0. However, one needs to store data associated with only 16 rows because the remaining part can be computed with convolution (if needed) by definition of vector Walsh transform. In this use case, the overall operations involved in

$$\begin{array}{cc|ccccccccccccccccccccccccc|c}
 & & 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 & \ldots\rho \\
 & & 0&0&0&0&0&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3&4&4&4&4&4 & \ldots\beta \\
 & & 0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4 & \ldots\alpha \\
\omega_{f_\pi} & \omega_{g_\pi f} & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\ldots & \ldots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
0 & 3 & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\ldots & \ldots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
1 & 2 & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\ldots & \ldots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
2 & 1 & 1&1&1&\textcircled{1}& &1&1&1&\textcircled{1}& &1&1&1&\textcircled{1}& &1&1&1&\textcircled{1}& &&&&& & \\
\ldots & \ldots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
3 & 0 & 1&1&1&1& &1&1&1&1& &1&1&1&1& &1&1&1&1& &&&&& & \\
\ldots & \ldots & &&&&&&&&&&&&&&&&&&&&&&&& &
\end{array} \tag{13}$$

Figure 13: The vulnerability profile of the robust-3-probing secure CMS scheme found with our formalization. This has been computed only for a sum of the hamming weight of the output spectral coordinates (i.e., the sum of probes) equal to 3. Red circles indicate where the vulnerability profile fails to be robust-3-SNI because for $\omega_{f_\pi} = 2$ there can be a dependency with up to $\alpha = 2$ or $\beta = 2$.

computing the convolutions for determining strong-non interference for $d = 3$ are about 1.35×10^6, for $d = 4$ about 4.92×10^7 and for $d = 5$ about 2.63×10^9.

The compact representation of the resulting vulnerability profile can be seen in Figure 13; In this matrix, $\omega_{f_\pi}, \omega_{g_\pi f}$ are the *compact spectral index* of pure internal probes f_π and output composed probes $g_\pi f$ respectively. Similarly, α, β, ρ are the compact spectral index of the shares of a, b and the refresh random bits r. Note that, for the spectral indexes $\alpha = 4$ or $\beta = 4$ and $\rho = 0$ the correlation between the extended probes f_π and $g_\pi f$ is 0; this solution is thus robust-3-probing secure.

One could check whether for $s > 4$ there can be suitable solutions to the above problem. However, we have found that for $s \geq 6$ the underlying formulas are not satisfiable anymore, leaving us conjecture that the above scheme hits an upper bound for $s = 5$.

3.1 Achieving Robust Strong non-Interference for CMS

Figure 13 shows that the solution is not robust-3-SNI (we have marked in red the correlation matrix positions that violate d-SNI properties). This is because, for two internal probes f_π one can get up to three shares of either a, b or both[8].

Indeed, as shown in Figure 12, if an output probe is placed on c_0, and two internal probes are placed in the refresh layer of adjacent cones, e.g., after the operation $r_{15} \oplus (a_2 \cdot b_3) \oplus r_0$ and $r_4 \oplus (a_1 \cdot b_3) \oplus r_5$, one can recover information about three shares of a (a_1, a_2, a_3) and three shares of b (b_0,b_2,b_3) with only two internal pure probes. The rationale is that, whenever we try to attack the input shares of a cone, we need to remove the two protecting randoms through two internal probes in adjacent cones. Since any adjacent cone will work with different shares, these will be attacked as well.

One countermeasure would be to increase the number of randoms that protect adjacent cones; for example, by adding one random for each pair of adjacent cones we would have that if an output probe is placed on one output c_i, no matter how the two internal probes are placed in the scheme, the output is always protected by two random and one would need obviously two other internal probes. Note that, even by adding output probes from adjacent cones, these will still be protected by four random so one is forced to use internal probes. In Figure 14, we show a proposal for such a scheme for $s = 4$ where additional random q_j are applied pair-wise to cones. Computing the vulnerability profile for such scheme yields Figure 15. As can be seen, two internal probes now do not imply a correlation with any share. This construction can be generalized into a sufficient condition:

[8] Note that we have shown columns up to all combinations of a and b not covered by random values.

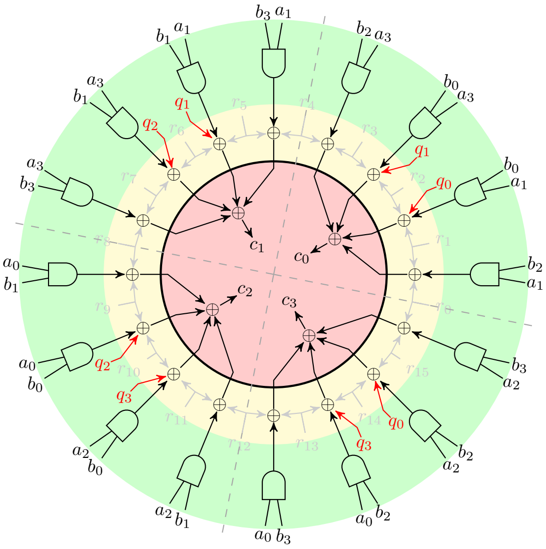


Figure 14: The robust-3-SNI CMS scheme proposed in this paper. Additional random are identified with the label q_j. Other randoms have been grayed out to avoid crowding the image.

Proposition 1. *Let s be the number of shares ($s \geq 4$); any generalized CMS scheme can become robust-$(s-1)$-SNI by adding $s \cdot (\lfloor \frac{s}{2} \rfloor - 1)$ randoms to the refresh layer such that each pair of adjacent cones shares $\lfloor \frac{s}{2} \rfloor - 1$ of them.*

Proof. In this scheme, each internal probe reveals one share of a, one share of b, and at most one of those randoms that are shared with the two adjacent cones. Moreover, each output probe is masked by $z = 2 + 2 \cdot (\lfloor \frac{s}{2} \rfloor - 1)$ randoms (for example, in Figure 14, c_1 is covered by 2 + 2 randoms, i.e., r_4, r_8, q_1, q_2).

Let us assume that we have $|\Pi| \geq 1$ output probes and i internal probes (Π is the configuration of the output probes and T_Π is the number of their partitions, see section 3). Assume that $|\Pi| + i \leq s - 1$. If this scheme was not robust-$(s-1)$-SNI, it would mean that i internal probes would provide information on more than $s - |\Pi|$ shares. This additional information could be obtained only when removing all of the $T_\Pi \cdot z$ randoms that cover Π. However, under the above assumptions, this is impossible because i internal probes provide less than $s - |\Pi|$ (i.e., less than $s-1$) randoms while the needed ones ($T_\Pi \cdot z$) are always greater or equal to $s-1$ as the following derivation shows:

$$\begin{aligned} & T_\Pi(2 + 2 \cdot (\lfloor \tfrac{s}{2} \rfloor - 1) && \geq \\ & 2 + 2 \cdot (\lfloor \tfrac{s}{2} \rfloor - 1) && \geq \\ & 2 + 2 \cdot (\tfrac{s-1}{2} - 1) && = \\ & s - 1 && \end{aligned} \tag{14}$$

In the case where $|\Pi| = 0$, every set of i provides just i input shares.

□

4 Analysis of the robust probing security of DOM-indep

As another example of application of our framework, we analyze the robust-d-probing security of another multiplication gadget referred to Domain Oriented Masking with independent shares (DOM-indep). Domain Oriented Masking is an alternative shared multiplication scheme which aims to be d-probing secure by using $d(d+1)/2$ random

$$
\begin{array}{cc|ccccccccccccccccccccccccc|l}
 & & 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 & \dots\rho \\
 & & 0&0&0&0&0&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3&4&4&4&4&4 & \dots\beta \\
 & & 0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4 & \dots\alpha \\
\omega_{f_\pi} & \omega_{g_\pi f} & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\dots & \dots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
0 & 3 & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\dots & \dots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
1 & 2 & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\dots & \dots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
2 & 1 & &&&&&&&&&&&&&&&&&&&&&&&& & \\
\dots & \dots & &&&&&&&&&&&&&&&&&&&&&&&& & \\
3 & 0 & 1&1&1&1& &1&1&1&1& &1&1&1&1& &1&1&1&1& &&&&& & \\
\dots & \dots & &&&&&&&&&&&&&&&&&&&&&&&& &
\end{array}
\tag{15}
$$

Figure 15: Vulnerability profile of CMS scheme with $s = 4$ when using additional randoms q_j.

bits [GMK16]. It is the basis above more sophisticated schemes have been built (such as DOM-dep or *DOM with dependent shares* [GM18]). The generic structure of DOM-indep is as follows:

$$
\begin{aligned}
c_0 &= \mathbf{a_0 b_0} + (a_0b_1 + r_0) + (a_0b_2 + r_1) + (a_0b_3 + r_3)\dots \\
c_1 &= (a_1b_0 + r_0) + \mathbf{a_1 b_1} + (a_1b_2 + r_2) + (a_1b_3 + r_4)\dots \\
c_2 &= (a_2b_0 + r_1) + (a_2b_1 + r_2) + \mathbf{a_2 b_2} + (a_2b_3 + r_5)\dots \\
c_3 &= (a_3b_0 + r_3) + (a_3b_1 + r_4) + (a_3b_2 + r_5) + \mathbf{a_3 b_3}\dots \\
\dots
\end{aligned}
$$

Bold multiplication terms are called inner-domain terms and do not require to be masked with randoms while, for the remaining cross-domain terms, the same random is reused to mask terms with mirrored indices. Parentheses indicate that terms are saved into registers before being *compressed* into the output share.

The current understanding of the DOM scheme is that, at least in the implementation that considers dependent shares (DOM-dep), it is not robust-d-SNI [MMSS19]. We will show here that also DOM-indep is not robust-d-SNI. To do so, we will study how inner pure probes f_π and output composed probes $g_\pi f$ behave.

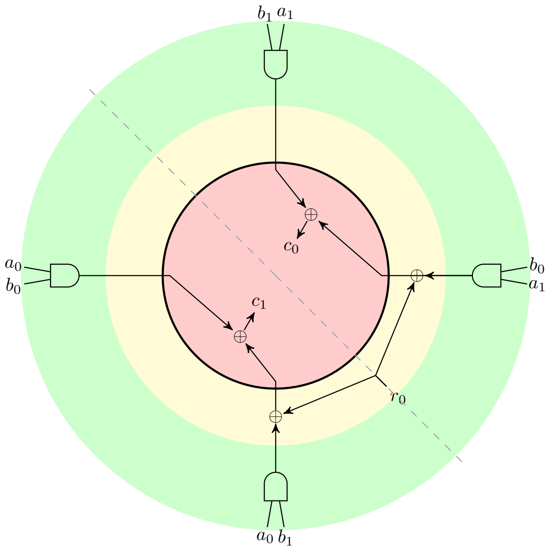


Figure 16: The DOM scheme for $d = 1, s = 2$

The reasoning is simple; consider Figure 16. For it to be robust-1-SNI, taking an output composed probe and no internal pure probes should not provide any information on input shares. However, an extended output probe on c_1, for example, allows to observe one of its inputs, i.e., a_0b_0 which is not covered by any random. Given that a_0b_0 correlates with either share a_0 or b_0 we would have one input share observable with zero internal probes, which goes against robust-1-SNI premises.

The vulnerability profile is shown in Figure 17 (left) where inner probes are accounted with ω_i while outputs and output probes are accounted with ω_o. We can see that it is not robust-1-SNI, as for $(\omega_i = 0, \omega_o = 1)$ there is a dependency with at least one share of α and β; however the gadget is still robust-1-probing secure.

		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	ρ
		0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	β
		0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	α
ω_i	ω_o																			
0	0	1																		
0	1	1	1		1	1					1	1		1	1		1	1		
0	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	0	1	1		1	1					1	1		1	1					
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	3	1	1		1	1	1	1	1		1	1	1	1	1		1		1	
4	4	1	1					1	1						1					

		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	ρ
		0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	β
		0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	α
ω_i	ω_o																			
0	0	1																		
0	1										1	1					1	1		
0	2	1		1				1		1										
1	0	1	1		1	1					1	1		1	1		1	1		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
6	0	1	1					1	1						1					
6	1				1		1				1		1				1		1	
6	2	1	1					1	1						1					

Figure 17: Vulnerability profiles of DOM without (left) and with (right) output register for $d = 1$ $(s = 2)$.

Adding an output register we obtain Figure 17 (right) where we can see that it is actually robust-1-SNI. Note that the one with the register has more inner probes because the original non-registered outputs have become internal. Figure 18 shows (part of) the vulnerability profiles for $d = 2$ which confirm that adding a register at the outputs makes the gadget robust-2-SNI.

We verified that the same happens for $d = 3$. We note that, in this case, the gadget DOM-indep uses 6 randoms, while the robust-3-SNI variant we propose for CMS uses 20 (without output register); this suggests that there exist a trade-off between registers and randomness when dealing with robust non interference. The ratio of random usage between DOM and our CMS construction is:

$$\frac{2\left(\frac{s^2}{2}+\left(\frac{s}{2}+1\right)s\right)}{(s-1)\;s}$$

		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	ρ
		0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	0	0	0	0	β
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	α
ω_i	ω_o																					
0	0	1																				
0	1	1	1			1	1												1	1		
0	2	1	1	1		1	1	1		1	1	1						1	1	1	1	
0	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	
1	0	1	1			1	1											1	1			
1	1	1	1	1		1	1	1		1	1	1						1	1	1	1	
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	0	1	1	1		1	1	1		1	1	1						1	1	1		
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
…	…																					

		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	ρ
		0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	0	0	0	0	β
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	α
ω_i	ω_o																					
0	0	1																				
0	1																					
0	2																					
0	3	1			1									1			1					
1	0	1	1			1	1											1	1	1		
1	1	1	1			1	1											1	1	1	1	
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	0	1	1	1		1	1	1		1	1	1						1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
…	…																					

Figure 18: Part of the vulnerability profiles of DOM without (left) and with (right) output register for $d = 2$ $(s = 3)$.

which, asymptotically, provides a 2× size factor. We conjecture that this is the cost one has to pay for sparing registers when building a robust-d-SNI gadget.

5 Conclusions

This work provided an alternative yet comprehensive view of robust probing security which, we argue, addresses more clearly the issues associated with composability of robust-probing secure gadgets. To achieve our goal, we introduced further distinctions for dealing with extended probes; in particular, these must be admitted to participate in a unique way during composition much like conventional outputs. We believe we have provided sufficient evidence that this new mathematical framework could work for analysis and synthesis of such gadgets.

Further work is needed to make the underlying computations more efficient as they are based on computation of the Walsh spectrum which incurs exponential cost. We believe that sparse matrix representation might be a tool worth investigating to improve correlation matrix computation. Another possible further extension of this work could be modeling t-PINI as well as inquiring about the minimum number of randoms required to achieve robust-d-strong-non-interference and/or investigating whether the ring structure of multiplication gadgets can be replaced by potentially more efficient refresh layers.

References

[BBD$^+$16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong Non-Interference and Type-Directed Higher-Order Masking. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 116–129, New York, NY, USA, 2016. ACM.

[BBFG18] Gilles Barthe, Sonia Belaïd, Pierre-Alain Fouque, and Benjamin Grégoire. maskVerif: automated analysis of software and hardware higher-order masked implementations. Technical Report 562, 2018.

[BBP$^+$16] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness Complexity of Private Circuits for Multiplication. Technical Report 211, 2016. C.

[BGI$^+$18] Roderick Bloem, Hannes Gross, Rinat Iusupov, Bettina Könighofer, Stefan Mangard, and Johannes Winter. Formal Verification of Masked Hardware Implementations in the Presence of Glitches. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, Lecture Notes in Computer Science, pages 321–353. Springer International Publishing, 2018.

[BGR18] Sonia Belaïd, Dahmun Goudarzi, and Matthieu Rivain. Tight Private Circuits: Achieving Probing Security with the Least Refreshing. Technical Report 439, 2018.

[Car10] Claude Carlet. Vectorial Boolean Functions for Cryptography. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 398–470. Cambridge University Press, Cambridge, 2010.

[CGLS20] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware private circuits: From trivial composition to full verification. Cryptology ePrint Archive, Report 2020/185, 2020. https://eprint.iacr.org/2020/185.

[CPRR14] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-Order Side Channel Security and Mask Refreshing. In Shiho Moriai, editor, *Fast Software Encryption*, Lecture Notes in Computer Science, pages 410–424. Springer Berlin Heidelberg, 2014.

[CS20] G. Cassiers and F. Standaert. Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Transactions on Information Forensics and Security*, 15:2542–2555, 2020.

[DGV95] Joan Daemen, René Govaerts, and Joos Vandewalle. Correlation matrices. In Bart Preneel, editor, *Fast Software Encryption*, Lecture Notes in Computer Science, pages 275–285. Springer Berlin Heidelberg, 1995.

[DPGM16] Brandon Dravie, Jérémy Parriaux, Philippe Guillot, and Gilles Millérioux. Matrix representations of vectorial boolean functions and eigenanalysis. *Cryptography and Communications - Discrete Structures, Boolean Functions and Sequences*, 8(4):555–577, October 2016.

[FGP$^+$17] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable Masking Schemes in the Presence of Physical Defaults and the Robust Probing Model. Technical Report 711, 2017. B.

[GM18] Hannes Gross and Stefan Mangard. A unified masking approach. *Journal of Cryptographic Engineering*, 8(2):109–124, June 2018.

[GMK16] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Technical Report 486, 2016.

[GMK17] Hannes Gross, Stefan Mangard, and Thomas Korak. An Efficient Side-Channel Protected AES Implementation with Arbitrary Protection Order. In Helena Handschuh, editor, *Topics in Cryptology – CT-RSA 2017*, Lecture Notes in Computer Science, pages 95–112. Springer International Publishing, 2017.

[GSM17] Hannes Gross, David Schaffenrath, and Stefan Mangard. Higher-Order Side-Channel Protected Implementations of Keccak. Technical Report 395, 2017.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, Lecture Notes in Computer Science, pages 463–481. Springer Berlin Heidelberg, 2003.

[MBR19] Lauren De Meyer, Begül Bilgin, and Oscar Reparaz. Consolidating Security Notions in Hardware Masking. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 119–147, May 2019.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-Resistant Masking Revisited. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 256–292, February 2019.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *Journal of Cryptology*, 24(2):292–321, April 2011.

[PGM11] Jérémy Parriaux, Philippe Guillot, and Gilles Millérioux. Towards a spectral approach for the design of self-synchronizing stream ciphers. *Cryptography and Communications*, 3(4):259–274, December 2011. C.

[RBN$^+$15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating Masking Schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215, pages 764–783. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[Sel10] P. Selinger. A Survey of Graphical Languages for Monoidal Categories. In Bob Coecke, editor, *New Structures for Physics*, volume 813, pages 289–355. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[XM88] G. Z. Xiao and J. L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[ZMB18] V. Zaccaria, F. Melzani, and G. Bertoni. Spectral Features of Higher-Order Side-Channel Countermeasures. *IEEE Transactions on Computers*, 67(4):596–603, April 2018.

A On enabling general reasoning about non-interference

The aim of this appendix is to show how the proposed formalization can enable the analysis and derivation of new inference rules around (robust) non-interference. For this purpose, consider the circuit represented in Figure 5. This circuit is one of building blocks used for computing the inverse in $GF(2^8)$ [CPRR14]. It is known that this type of circuit is d-SNI when both g and f are d-SNI [BBD+16]. Figure 19 shows the corresponding correlation matrices diagram when not considering robustness against glitches (namely probes are non extended probes).

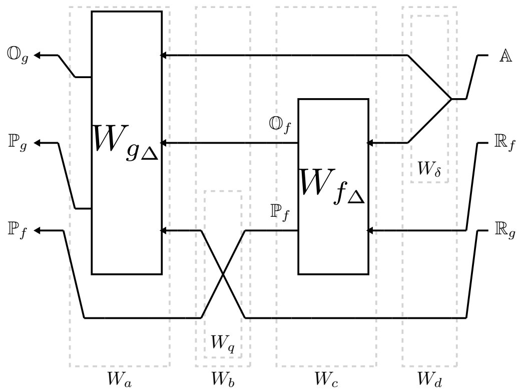


Figure 19: Map between Fourier transforms of probability distributions implied by the considered example composition pattern.

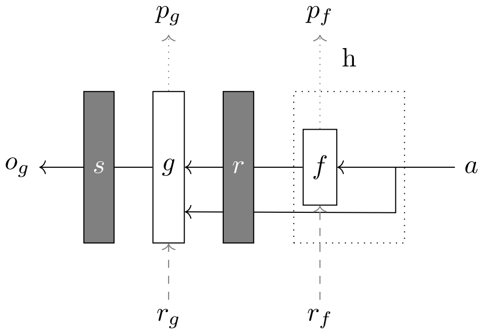


Figure 20: The considered example composition pattern, gray boxes are registers.

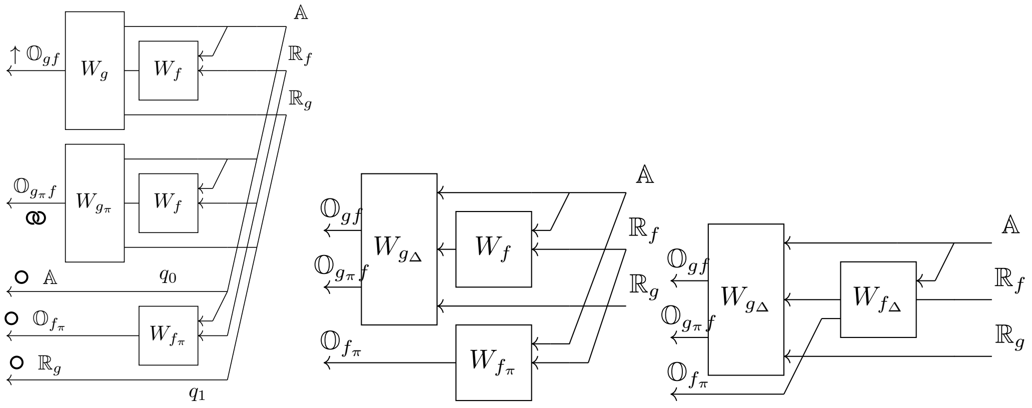
Consider now the same circuit with a register between f and g (Figure 20). This can be seen as the composition of two blocks g and h, where h is the block that duplicates the input and propagates one of the paths through f. We already know that, for robust non-interference, this composition has the vulnerability profile shown in Eq. (10) and rewritten here to consider h:

$$s_\Delta \bullet g_\Delta \bullet r_\Delta \bullet h_\Delta = \Delta\{\overset{\circ}{W}_{h_\pi}, \overset{\circ\circ}{W}_{g_\pi h}, \overset{\uparrow}{W}_{gh}\}$$

where

$$W_h = (I \otimes W_f \otimes I)(I \otimes I \otimes W_\delta) \tag{16}$$

$$W_{h_\pi} = (I \otimes W_{f_\pi} \otimes I)(I \otimes I \otimes W_\delta) \tag{17}$$



(a) Initial **(b)** Intermediate **(c)** Final

Figure 21: Pruning of the a vulnerability profile considering equivalences and dominance relations of the correlation matrix calculus.

while I is the identity, and W_δ is the correlation matrix of the duplication function $\delta = x \mapsto (x, x)$.

Graphically, the above equations correspond to Figure 21a. From there, one can reason by pruning redundant paths. For example, the distribution of $\mathbb{A}$ reaches the output through the extended probe $g_\pi f$, so path q_0 can be pruned off without loss of generality (same thing for path q_1) obtaining 21b. Considering the commutativity across duplication points we can move f_π and then substitute the definition of fan (Eq. 5) for both f and g, one can obtain 21c, which is exactly the vulnerability profile of the non-robust case shown in Figure 19 where probes now are extended probes. This means that the reasoning about the non robust case can be directly applied also to this case, i.e., if f_Δ is d-SNI and g_Δ is d-SNI (d-NI) then the composition in Figure 20 is robust-d-SNI (robust-d-NI). Note that the derivation of the same general conclusions through a classic approach would have possibly required a much more involved demonstration.

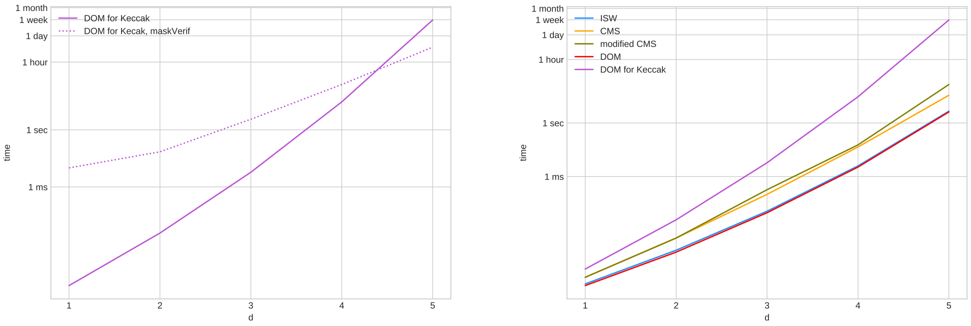
B Computational complexity and scalability of the proposed approach

The vulnerability profile of a function f is computed starting from its correlation matrices W_f and W_{f_π} (i.e. W_{f_Δ}). The complete computation of these matrices could become quickly impracticable due to the large number of their elements, which, in turn, is exponentially related to the number of inputs, outputs and probes analysed.

To reduce the time and space computational complexity of this operation, we store only the rows that refer to single outputs and probes, and compute on-demand the remaining rows by using convolution. Besides, we exploit the fact that correlation matrices are sparse (as explained in Section 3), to speed up the convolution itself. In the following, we show some estimates of computation time when such sparsity is taken into account.

Let us consider the non linear function χ of Keccak, implemented using the DOM multiplication gadget to make it probing secure at the d-th order [GSM17]. We recall that the internal state of Keccak is divided into groups of five bits, called *rows*, and function χ is applied row by row. Given x_0, x_1, x_2, x_3, x_4 (the row bits), χ is defined as:

$$y_i = \chi(x_i, x_{i+1}, x_{i+2}) = x_i + (\overline{x}_{i+1} x_{i+2})$$



(a) Scalability computed for χ of Keccak with DOM, and comparison with time needed to apply maskVerif tool [BBFG18] to the same algorithm.

(b) Scalabity computed for known algorithms.

Figure 22: Estimated time needed to compute the vulnerability profile for well known algorithms.

where indices are computed modulus 5. To make it probing secure at the d-th order, each element x_i is split into $d+1$ shares $x_i^0, x_i^1, \dots x_i^d$, and a share y_i^j of the output y_i is computed as follows [GSM17]:

$$y_i^j = \Big(x_i^j + \big(\overline{x}_{i+1}^j x_{i+2}^j + \sum_{h>j}(x_{i+1}^j x_{i+2}^h + r_{j+\frac{h(h-1)}{2}}) + \sum_{h<j}(x_{i+1}^j x_{i+2}^h + r_{h+\frac{j(j-1)}{2}})\big)\Big)$$

Note that parentheses indicate that terms are saved into registers; thus there is a register before the compression layer in DOM (reg_1), one that stores multiplication results (reg_2) and one that stores the final output's share y_i^j (reg_3).

Define as χ_d the function that computes all the shares of y_i. In the corresponding circuit, we assume an extended probe on each wire that ends into a register: we thus have $n_1 = (d+1)d$ probes placed before reg_1, $n_2 = d+1$ before reg_2 and $n_3 = d+1$ before reg_3.

In the correlation matrix of χ_{d_Δ} each row referring to a probe before reg_1 has only $a_1 = 8$ nonzero elements, while for a probe before reg_2 or reg_3 the nonzero elements are respectively $a_2 = 2^{d+1}$ and $a_3 = 2^{d+2}$. Each remaining row is computed by convolution of p single-probe rows, and it has, in average, $av(p)$ elements different from 0:

$$av(p) = \frac{\sum\limits_{h=0}^{p}(\sum\limits_{k=0}^{p-h} n_1^k a_1^k \cdot n_2^{p-k-h} a_2^{p-k-h}) n_3^h a_3^h}{(n_1+n_2+n_3)^p}$$

where $2 \le p \le n_1+n_2+n_3$. Figure 22(a) reports the estimated time needed to compute the correlation matrix of χ_{d_Δ} (solid line), in comparison with the time needed to execute maskVerif [BBFG18] to show that χ with DOM algorithm is robust-d-NI (dotted line). Figure 22(b) reports the estimated time for other known gadgets to compute their correlation matrices. In both cases, the value of d is varying between 1 and 5 and we assume to work with a 8 processors, 4GHz machine with a 1 integer operation per clock cycle throughput[9].

[9] Luckily, the computation of multiple rows of the correlation matrix can be done in parallel.