Theory of masking with codewords in hardware: low-weight dth-order correlation-immune Boolean functions

Shivam Bhasin Claude Carlet Sylvain Guilley

May 21, 2013

Abstract

In hardware, substitution boxes for block ciphers can be saved already masked in the implementation. The masks must be chosen under two constraints: their number is determined by the implementation area and their properties should allow to deny high-order zero-offset attacks of highest degree. First, we show that this problem translates into a known trade-off in Boolean functions, namely finding correlation-immune functions of lowest weight. For instance, this allows to prove that a byte-oriented block cipher such as AES can be protected with only 16 mask values against zero-offset correlation power attacks of orders 1, 2 and 3. Second, we study dth-order correlation-immune Boolean functions $\mathbb{F}_2^n \to \mathbb{F}_2$ of low-weight and exhibit such functions of minimal weight found by a satisfiability modulo theory tool. In particular, we give the minimal weight for $n \leq 10$. Some of these results were not known previously, such as the minimal weight for $(n = 9, d = 4)$ and $(n = 10, d \in \{4, 5, 6\})$. These results set new bounds for the minimal number of lines of binary orthogonal arrays. In particular, we point out that the minimal weight $w_{n,d}$ of a dth-order correlation-immune function might not be increasing with the number of variables n.

Keywords: side-channel attacks, zero-offset high-order correlation power attacks, masking countermeasure, Boolean functions, dth-order correlation-immunity, orthogonal arrays, SMT.

1 Introduction

Masking is a technique to deceive attacks that exploit side-channel emanations leaking from cryptographic implementations. The main difficulty when designing a masking scheme is to pass through the substitution boxes (sboxes). Classical techniques are:

- Sboxes recomputation: before starting the encryption, the sbox S is replaced in memory by a masked version. An alternative is the mathematical computation of the sbox during the algorithm; this solution can totally avoid look-ups.

- Global look-up table: a huge sbox is implemented, with three inputs (the functional input, the input and the output masks).
- Random choice of premasked sboxes: a set of w masked sboxes is already implemented, and their selection is done randomly at each new encryption.

All these techniques come at a non-negligible cost: sboxes recomputation requires time and therefore incurs a non-negligible latency, the global look-up table is greedy in memory or silicon area, and the random choice of premasked sboxes needs to function with a limited amount of masks thus compromising with the entropy. In this article, we are interested in a masking solution without timing overhead and a limited significant area overhead. This rules out the two first options.

An architecture that implements the random choice of premasked sboxes in hardware has been presented in [22]. We summarize its main features. A small number of sboxes (*e.g.* $w = 16$ for the AES) are embedded already masked in the implementation. For cost-efficiency, the masks cannot be updated i.e. either ROM is used instead of RAM, or the masked sboxes are simply synthesized in logic gates. The gate count of a masked sbox (with constant masks) is similar to that of the unmasked sbox, since both are basically random functions or structured functions whose properties are preserved after masking. Such masking scheme is especially relevant for the ciphers that use many instances of the same sbox (*e.g.* AES (*i.e.* Advanced Encryption Standard [23]) or PRESENT [3]). For instance, AES uses 16 identical sboxes addressing 16 different bytes of the 128-bit plaintext in its round function. Therefore, it is natural to consider a countermeasure where the 16 sboxes are instantiated each with a different mask. At every encryption, the allocation of the sbox for each of the 16 plaintext bytes is done randomly. Compared to the sbox recomputation or the global look-up table options, only $w = 16$ masks are possible amongst the 256 possible ones. Still, it is interesting to assess the level of resistance such a countermeasure can achieve. Because all the w sboxes are evaluated in parallel, we exclude collision attacks or high-order attacks with the combination of leakage from individual sboxes. At least, we assume that the designer has taken all the actions to make those attacks impossible. Thus, the only viable attack that remains is the high-order zero-offset attack [30] (at order d, that consists in correlating a sensitive variable with the dth power of the centered side-channel leakage.

Our problem involves three parameters:

1. n: the sbox bitwidth, *e.g.* $n = 8$ bit for AES,
2. d: the targetted order of resistance, *e.g.* $d = 1, 2, 3$ or more,
3. w $(1 \leq w \leq 2^n)$: the number of masks, that coincides with the number of sboxes (*i.e.* the implementation cost).

The analysis will reveal (Theorem 1) that the problem consists in:

- finding the masks as a code $C \subseteq \mathbb{F}_2^n$ of dimension n, size w and dual-distance $d+1$; or equivalently
- finding a Boolean function f (the indicator of the C in $\mathbb{F}_2^n$) of weight w and that is dth-order correlation immune.

Those problems are classical ones, and actually consist in trade-offs: either *maximizing d for a given w* or *minimizing w for a given d.*

In this article, we intend to find correlation-immune Boolean functions f whose support (the reciprocal image of 1, *i.e.* $\mathrm{supp}(f) = f^{-1}(1)$) has a cardinality as small as possible. Therefore we are also interested in unbalanced f; thus, in the sequel, f is non-necessarily resilient. Moreover, for reasons which will clearly appear below, we are only interested in nonzero functions f.

Definition 1. *A Boolean function f is dth-order correlation-immune (d-CI for short) if its output value distribution does not change when at most d components of its input are fixed to arbitrary values.*

Let us recall that the Hamming weight $w_H(x)$ of a vector $x \in \mathbb{F}_2^n$ is the number of bits set to one. A classical characterization of d-CI functions involves the Fourier or the Walsh transform.

Definition 2. *The Fourier transform $\widehat{f}$ of a function f is defined as $\widehat{f} : a \in \mathbb{F}_2^n \mapsto \sum_x f(x)(-1)^{a\cdot x} \in \mathbb{Z}$.*

Definition 3. *The Walsh transform W_f of a function f is the Fourier transform of its character $f_\chi : x \mapsto (-1)^{f(x)}$.*

It has been proved in [31] that f is d-CI if and only if $\forall a \in \mathbb{F}_2^n, 0 < w_H(a) \leq d, W_f(a) = 0$ (or equivalently, if and only if $\forall a \in \mathbb{F}_2^n, 0 < w_H(a) \leq d, \widehat{f}(a) = 0$).

We recall that d-CI functions are functions whose supports are (simple) binary orthogonal arrays of strength d, that is, linear or nonlinear codes of dual distance at least $d+1$. A survey on orthogonal arrays can be found in [10]. The first contribution of this paper is the exhibition of new d-CI functions. We give the minimal weights of all d-CI functions of n variables for $n \in [\![1, 10]\!]$, and notably the previously unknown minimal weights of functions with those parameters: $(n = 9, d = 4)$ and $(n = 10, d \in \{4, 5, 6\})$.

The rest of the paper is organized as follows. The security analysis of the masking scheme based on the random choice of premasked sboxes is carried out in Sec. 2. This first section is aimed towards the definition of the criteria to find masks (n-input d-CI functions of weight w). Such functions are studied in Sec. 3 with three classical methods:

1. as indicators of binary linear codes (Sec. 3.3.2),
2. with the Maiorana-McFarland construction [5] (Sec. 3.3.3), and
3. using simple binary orthogonal arrays (Sec. 3.4).

These methods do not always allow to derive the minimal weight. Therefore we also conduct an exhaustive computer search using a satisfiability modulo theory (SMT) tool. Previously unknown d-CI functions of lowest weight are exhibited. Finally, the conclusions are drawn in Sec. 4. Some appendices provides technical proofs (App. A), the difficulty to build d-CI functions in certain cases as ($n \in \{8, 9, 10, 11\}, d = 2$), ($n \in \{9, 10, 11, 12\}, d = 3$) (App. B), and detail the difference between d-CI functions and general orthogonal arrays (App. C). In this appendix, we also present an open problem: the minimal weight $w_{n,d}$ of a d-CI function of n variables might not increase with n, as is the case for (non-simple) orthogonal arrays.

2 Boolean Masking

2.1 Presentation of Boolean Masking

The goal of cryptographic algorithms is to provide various functionalities, such as ensuring the confidentiality, the authenticity or the integrity of transmitted information. For instance, the confidentiality of blocks of data can be achieved by the encryption, using algorithms like AES. Those algorithms, considered blackboxes (that is, leaking no information on the computations made in them; the only information available to the attacker being the inputs and the outputs to the box), are purportedly strong: no attack (or only little efficient ones, *i.e.* concretely impractical) are known that would be able to extract the key by intercepting their input, output or both.

However, once the algorithm is implemented (either in software or in hardware), the so-called "side-channel attacks" become possible. These attacks consist in recording the unintentional emanations emitted from an implementation in a view to derive some information about the secret key. As a matter of fact, the key is mixed with the encrypted data, and any leakage that depends on this data can be used as an oracle: the attacker guesses a manageable part of the key (a "subkey"), and deduces an internal variable (usually referred to as a "sensitive variable"). If the sensitive variable is somehow leaked by the device, then the attacker can test exhaustively all the values of the subkey, and check whether the measured leakage is likely to originate from this variable. The correct subkey is the subkey hypothesis that maximizes the dependency between a modelled leakage and the measured leakage.

For example, the plaintext X that is encrypted by a block cipher is usually mixed with a key K before being input in a substitution box (sbox) S. The sbox plays a role of "confusion", that allows to decorrelate the output Z from the plaintext X. At this stage, $Z = S(X \oplus K)$ is sensitive. Indeed, this variable depends on the key (usually, only one subkey enters each sbox), and can be predicted from X. So, the attacker is able to predict Z for every possible value of K.

In practice, the variable Z is leaking much because it is stored in a series of flip-flops (register), waiting for the next processing that consists in the iteration

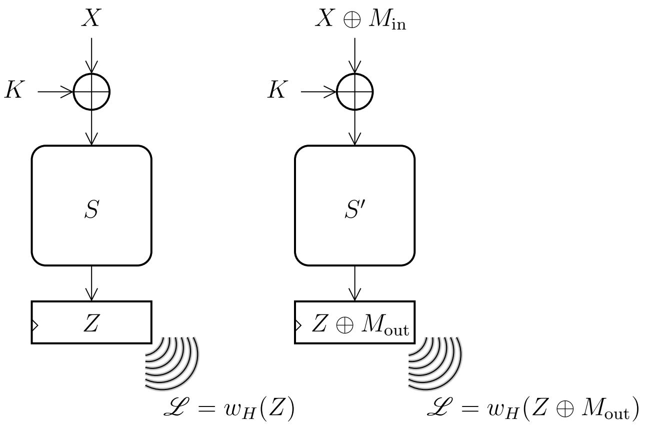
X
K
S
Z
$\mathscr{L} = w_H(Z)$
$X \oplus M_{\text{in}}$
K
S'
$Z \oplus M_{\text{out}}$
$\mathscr{L} = w_H(Z \oplus M_{\text{out}})$

Figure 1: Side-channel leakage $\mathscr{L}$ in a block cipher, unprotected (*left*) and masked (*right*).

of the same "round function". This next operation is carried out on the data just processed previously by the same round function. The register is a micrometric object, whose contents cannot be measured directly. Only the sum of each bit set in Z is available when using a mesoscopic sensor (*e.g.* an integrated antenna of 0.1 to 1 mm diameter). This leakage model is called the Hamming weight leakage function. It is widely accepted as a valid model for representative devices [4]. This model is suitable for FPGAs that are configured with a custom function, ASIC and microcontroller based implementataions. In other words, the Haming weight leakage model stays valid for a wide range of pratical cryptographic implementations.

The leakage model of a typical block cipher is illustrated in the left-hand side of Fig. 1: $\mathscr{L} = w_H(X)$. As it is easy to correlate this leakage model to actual measurements, many side-channel key retrieval attacks have been devised in theory and conducted successfully in practice. Therefore, it is important that cryptographic implementations be protected against side-channel attacks.

The Boolean masking is a protection where the internal data are manipulated XORed with a random mask that changes at every computation [16]. For instance, X is manipulated as $X \oplus M_{\text{in}}$. The key mixing (operation: $X \mapsto X \oplus K$) is compatible with the Boolean masking because it is a linear function. The application of the sbox S is complex, because it is not linear (*i.e.* $S(X \oplus K \oplus M_{\text{in}}) \neq S(X \oplus K) \oplus S(M_{\text{in}})$).

One method to apply sbox on masked data involves the re-computation of the sbox applied as follows:

- the masks M_{in} and M_{out} are randomly drawn;
- the sbox S is recomputed – this means that a new table S' is created, with this functionality: $x \mapsto S'(x) = S(x \oplus M_{\text{in}}) \oplus M_{\text{out}}$.

This way, the data can enter the sbox masked by M_{in} and remains masked at the output, as $Z \oplus M_{\text{out}}$. This countermeasure is illustrated in the right-hand side of Fig. 1. If we simply note M instead of M_{out}, the leakage model is now:

$$\mathscr{L} = w_H \left(Z \oplus M \right) \quad . \tag{1}$$

In some applications, the recomputation of the sbox S' is too long. For instance, a hardware AES with 128-bit key can be executed in about 11 clock periods, whereas a single sbox recomputation requires at least 256 clock periods. Therefore, it is relevant to precompute required set of masked sboxes S' once for all, and to cache them. Indeed, keeping all the possible S' certainly requires too much memory. Although not impossible (256^3 bytes, *i.e.* 16 MB, are needed for AES [19]), such "Global Look-up-Table" [24] is expensive. Please note that in hardware where implementations are parallel in general, several instances of a Sbox are used and all of them must be masked. A single AES GLUT would occupy about 90% of the available memory in a low-cost FPGA thus making a parallel AES implementation unfeasible. It explains why fewer number of masks is preferable. If w_{in} (resp. w_{out}) masks are used in input (resp. output) of the sbox, then the memory requirement is only $256 \times w_{\text{in}} \times w_{\text{out}}$ bytes.

Thus, in terms of implementation, it is more efficient to use for M a subset of all possible masks (we focus on $M = M_{\text{out}}$ in the sequel because M_{in} does not appear in the expression of the leakage model $\mathscr{L}$). So, unlike previous masking schemes, the mask M is not fully entropic. If $n \in \mathbb{N}^*$ is the input size of an sbox, then the masks belong to a subset of $\mathbb{F}_2^n$ called $\mathcal{M}$. A choice of masks is identified by the Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, that is defined as the indicator of $\mathcal{M}$ in $\mathbb{F}_2^n$, *i.e.* $f(x) = 1 \Longleftrightarrow x \in \mathcal{M}$.

From a designer point of view, security is not the only parameter. The key parameters for an implementations are area, performance, power-consumption and security of a design. Therefore often a designer is required to make appropriate trade-offs between the key parameters to make the implementation practical. For instance, lets focus on two complementary parameters i.e. area and security. Most of the methodologies to secure an implementation (including Boolean masking) rely on redundancy in the design *i.e.* increase of area. Area increase can further lead to increase in power-consumption and decrease in speed. In such a scenario, an elevated security with fewer number of mask has a clear interest. In the following, we analyze the security of the depleted masking countermeasure.

2.2 Security Analysis of the "Depleted" Masking Countermeasure

It is well-known that an attack can be mounted against the masking scheme just presented, since the leakage function conveys information on the sensitive variable Z. Indeed, the mutual information $\mathsf{I}[\mathscr{L}; Z]$ between $\mathscr{L}$ and Z is equal to zero if and only if M is uniformly distributed. However, the objective is not to achieve perfect security, but to make the exploitation as hard as possible, *i.e.*

to increase as much as possible the number of traces required for an attack to succeed. Now, an information-theoretic attack (such as a mutual-information analysis, *aka* MIA [1]) consists, in practice, in computing an estimation of a mutual information on noisy signals ($\mathscr{L}$ can only be measured approximately, because of algorithmic noise and imperfect acquisition conditions). Such an estimation requires a lot of measurements to be accurate (*i.e.* reliable).

The defender's strategy is to find masks that reduce the dependency between the leakage $\mathscr{L}$ and Z. To quantify the amount of dependency, we recall this result, that holds whatever the random variables $\mathscr{L}$ and Z:

$$\mathscr{L} \perp\!\!\!\perp Z \quad \Longrightarrow \quad \forall d \in \mathbb{N},\ \mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d | Z\right]\right] = 0 \ .$$

In this equation, the symbol "$\perp\!\!\!\perp$" expresses the statistical independence, and $\mathbb{E}$ (resp. Var) denotes the expectation (resp. variance) operator. If we call "classes" the values z taken by Z, then the term $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d | Z\right]\right]$ is also called the inter-class variance of $\mathscr{L}^d$; the expectation of $\mathscr{L}^d$ is computed in each class $Z = z$ and the variance is taken on Z.

Without countermeasure, the leakage $\mathscr{L}$ has, in average, a value that depends on the sensitive variable Z; indeed, as shown at the left of Fig. 1, $\mathscr{L} = w_H(Z)$. This means that the mean of $\mathscr{L}$ knowing $Z = z$ for each class $Z = z$ is scattered, or equivalently that $\mathbb{E}[\mathscr{L}|Z]$ depends on $Z = z$, or equivalently that the inter-class variance $\mathsf{Var}[\mathbb{E}[\mathscr{L}|Z]]$ is nonzero. For instance, this condition is required for the correlation power attack (CPA [4]) to work. The CPA computes the Pearson correlation coefficient ρ between the leakage and the sensitive variable. The CPA fails if $\rho(\mathscr{L}, Z) = \rho(\mathbb{E}[\mathscr{L}|Z], Z) = 0$, *i.e.* if $\mathbb{E}[\mathscr{L}|Z = z]$ does not vary with z. The goal of a first-order masking countermeasure is to equal the inter-class variance to zero, by balancing the leakage in each class. Thus $\mathsf{Var}[\mathbb{E}[\mathscr{L}|Z]] = 0$, but $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^2|Z\right]\right] \neq 0$. Second-order masking countermeasures not only balance, but also normalize the leakage, so that we have $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^d|Z\right]\right] = 0$ for $d \in \{1, 2\}$. Thus, by extrapolation, the quality of the countermeasure can be assessed as the largest d such that, for all $i \in [\![1, d]\!]$, $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^i|Z\right]\right] = 0$. The greater it is, the better the countermeasure. Such a countermeasure is said to resist dth-order attacks.

Back to the information-theoretic notion of security ($\mathsf{I}[\mathscr{L}; Z] \neq 0$ if M is not uniform) it shall be noted that in practice, the authors of the MIA [9] concur that MIA is seldom more efficient than correlation attacks (like CPA [4] or high-order CPA [30]). Moreover, the paper [29] indicates that MIA can be relevant only if the leakage model is badly known by the attacker. This is however not the case in this paper, where we assume that the designer of the countermeasure knows or is able to test the leakage model.

To allow for a trade-off between the security and the implementation cost, we envision a case where some masks are forbidden, whereas the others can be used. More precisely, the masks M follow a uniform distribution over a subset $\mathcal{M} \subseteq \mathbb{F}_2^n$.

Theorem 1. *Let $d \leq n$. A masking scheme with leakage as per Eqn.* (1) *resists* *to dth-order attacks if the indicator f of the masks $\mathcal{M}$ is a dth-order correlation-immune Boolean function.*

Proof. Theorem 1 had already be demonstrated for $d \leqslant 2$ in [20]. However, the proof was very *ad hoc* (with manual developments of the power of sums, and without spectral analysis tools). We simplify and generalize it below. □

Remark 1. *Incidentally, we notice that the notion of d-order attacks is independent of the condition $d \leq n$. Actually, when $d = n$, Theorem 1 states that f must be n-CI, which is equivalent (*cf *Example 1) to having all the masks. In this case,* $\mathsf{I}[\mathscr{L}; Z] = 0$ *and the countermeasure actually resists attacks of any order $d \in \mathbb{N}$.*

Remark 2. *It is noteworthy that theorem 1 links two different concepts of* correlation*. The attack uses* dth-order correlation power analysis*, whereas the defense (the countermeasure) employs* dth-order correlation-immunity*. It is a fortuitous coincidence that those two notions encounter one with the other in the framework of side-channel analysis.*

Remark 3. *We see that finding d-CI functions of low-weight is relevant in this case, because the memory size to save the masked sboxes is equal to the number of masks,* i.e. *to* $\mathsf{Card}[supp(f)]$.

We begin with one lemma:

Lemma 1. $\widehat{w_H^d}(z) = 0 \iff w_H(z) > d$.

Proof. See Appendix A, and in particular the property P5 of Lemma 12. □

Proof. (**of Theorem 1**) Given one $z \in \mathbb{F}_2^n$, the random variable $(w_H(Z \oplus M))^i \mid Z = z$ depends only on M. Its average is:

$$\begin{aligned}
& \mathbb{E}\left[(w_H(Z \oplus M))^i \mid Z = z\right] \\
= \quad & \frac{1}{\widehat{f}(0)} \sum_{m \in \mathcal{M}} w_H^i(z \oplus m) \\
= \quad & \frac{1}{\widehat{f}(0)} \sum_{m \in \mathbb{F}_2^n} f(m) \cdot w_H^i(z \oplus m) \quad \text{// Expression 1} \\
= \quad & \frac{f}{\widehat{f}(0)} \otimes w_H^i(z) \\
= \quad & \frac{1}{2^n} \frac{\widehat{\widehat{f}}}{\widehat{f}(0)} \cdot \widehat{w_H}^i(z) \ . \quad \text{// Expression 2}
\end{aligned}$$

Note that expression 2 comes from these properties: $\widehat{\phi \otimes \psi} = \widehat{\phi} \times \widehat{\psi}$, $\widehat{\phi} \otimes \widehat{\psi} = 2^n \widehat{\phi \times \psi}$, and thus $\phi \otimes \psi = \frac{1}{2^n} \widehat{\widehat{\phi} \times \widehat{\psi}}$. They relate the convolution product ($\otimes$) and the regular product ($\times$) when the Fourier transform is applied.

Now, security metric is equal to the variance on $z \in \mathbb{F}_2^n$ of the previous expression:

$$\frac{1}{2^{3n}} \sum_z \left(\widehat{\frac{\widehat{f}}{\widehat{f}(0)}} \cdot \widehat{w_H}^i(z) \right)^2 - \left(\frac{1}{2^n} \sum_z \frac{1}{\widehat{f}(0)} \sum_{m \in \mathbb{F}_2^n} f(m) \cdot w_H^i\left(z \oplus m\right) \right)^2 ,$$

using expressions 2 and 1. The first term simplifies when considering Parseval's relation: $\sum \widehat{\phi}^2 = 2^n \sum \phi^2$. The second term can also be simplified:

$$\begin{aligned} & \frac{1}{2^n} \sum_{z \in \mathbb{F}_2^n} \frac{1}{\widehat{f}(0)} \sum_{m \in \mathbb{F}_2^n} f(m) \cdot w_H^i\left(z \oplus m\right) \\ = \quad & \frac{1}{2^n} \frac{1}{\widehat{f}(0)} \sum_{m \in \mathbb{F}_2^n} f(m) \cdot \left(\frac{1}{2^n} \sum_{z \in \mathbb{F}_2^n} w_H^i\left(z \oplus m\right) \right) \\ = \quad & \frac{1}{2^n} \sum_{z \in \mathbb{F}_2^n} w_H^i\left(z\right) = \frac{\widehat{w_H}^i(0)}{2^n} \quad . \end{aligned} \tag{2}$$

Eventually, the inter-class variance $\mathsf{Var}\left[\mathbb{E}\left[\mathscr{L}^i|Z\right]\right]$ takes the following simple form:

$$\mathsf{Var}\left[\mathbb{E}\left[w_H^i(Z \oplus M) \mid Z\right]\right] = \frac{1}{2^{2n}} \sum_{z \neq 0} \left(\frac{\widehat{f}(z)}{\widehat{f}(0)} \cdot \widehat{w_H^i}\left(z\right) \right)^2 \quad . \tag{3}$$

If the masking scheme resists against dth-order attacks, then, by definition, for all $0 < i \leq d$, Eqn. (3) is null. Let us consider simply the case $i = d$. For all $z \neq 0$, $\widehat{f}(z) \cdot \widehat{w_H^d}\left(z\right) = 0$. The contrapositive of lemma 1 is the equivalence $\widehat{w_H^d}\left(z\right) \neq 0 \iff w_H\left(z\right) \leq d$. Thus, for all nonzero z with Hamming weight smaller or equal to d, $\widehat{f}(z)$ is equal to zero. This means that f is d-CI.

Conversely, if f is d-CI, then Eqn. (3) is null for all $0 < i \leq d$, because:

- when $w_H\left(z\right) \leq i$, $\widehat{f}(z) = 0$ because f is i-CI (weaker property of the dth-order correlation-immunity),
- when $w_H\left(z\right) > i$, $\widehat{w_H^i}\left(z\right) = 0$ (lemma 1).

□

Corollary 1. *Theorem 1 also holds if the leakage variable (Eqn. (1)) is centered,* i.e. $\tilde{\mathscr{L}} = \tilde{w_H}\left(Z \oplus M\right) = w_H\left(Z \oplus M\right) - n/2$ *is considered instead of* $\mathscr{L} = w_H\left(Z \oplus M\right)$.

Remark 4. *It has indeed been proved, for instance in [25], that the distinguishers are optimal if the leakage is centered before processing.*

Proof. The basic architecture of the proof of Theorem 1 still holds, albeit with some minor adjustments.

Let us assume that the equivalent of Eqn. (3): $\mathsf{Var}\left[\mathbb{E}\left[\left(\tilde{w_H}\left(Z \oplus M\right)\right)^i \mid Z\right]\right] = \frac{1}{2^{2n}} \sum_{z \neq 0} \left(\frac{\widehat{f}(z)}{\widehat{f}(0)} \cdot \widehat{\tilde{w_H}^i}\left(z\right)\right)^2$ is null for all $0 < i \leq d$. Then for all $0 < i \leq d$ and $z \neq 0$, $\widehat{f}(z) \cdot \widehat{\tilde{w_H}^i}\left(z\right) = 0$. Let us choose one $i \in [\![1, d]\!]$. According to the property P2 of lemma 10, we have that $\widehat{\tilde{w_H}^i}\left(z\right) \neq 0$ if $w_H\left(z\right) = i$. So, $\widehat{f}(z)$ if $w_H\left(z\right) = i$. But now, this is true for all $i \in [\![1, d]\!]$. So, this proves that f is d-CI.

The converse is shown as in the proof of Theorem 1, with the following modification: Property P1 of lemma 10 is invoked instead of lemma 1. □

2.3 Example of Application

The results can be applied on algorithms for which the datapath is segmented in words of n-bits. For instance, the AES [23] manipulates bytes, hence $n = 8$. Then, the security level shall be determined. First-order resistance is a minimum for most evaluation schemes, but for practical applications, a resistance $d > 2$ is welcome, as second-order attacks are now known for more than 12 years [17]. Higher-order attacks, *e.g.* attacks of order $d = 3$, are more difficult to realize successfully provided the noise level is sufficient [18] (*e.g.* hardware implementations are noisy owing to the large activity of parallel operations that can be seen as noise). So, for a strong security level of today, we set $d = 3$. Then, the designer looks for the minimally sized masks set. A look-up in Tab. 3 shows that this set can be depleted from 256 down to 16 masks. In this case, the entry of the table is not gray, which means that the masks are actually the codewords of a linear code[1] namely $[8, 4, 3]$. The implementation overhead is straightforward: 16 different masks can be used, hence a factor 16 in resources usage. Also, in a system-on-chip, the masks are drawn from a true random number generator (TRNG). For the randomness to be of good quality, this module is typically limited in throughput. Thus, it is also beneficial for the overall secure system (TRNG and AES) to limit the amount of required randomness per encryption.

In the case of AES, the datapath is itself made up of 16 bytes arranged as a 4×4 matrix. In this matrix, each byte is processed similarly (notably, the non-linear operation SB, called SubBytes, is composed of 16 identical sboxes). Therefore, before the application of the countermeasure, 16 identical sboxes were required to compute one round of AES in one clock period. To apply the masking scheme, 16 different S' are computed one for each mask in $\mathcal{M}$. The main idea is to reuse the 16 different S' for 16 bytes. In hardware, where the 16 bytes are processed in parallel, reusing S' can be achieved simply by rotataion (as illustrated in Fig 2).

The masking applies as follows:

[1] Put differently, the mask values, denoted $\mathcal{M}$ in Sec. 2.1, coincide with the codewords, denoted C in Sec. 3.3.1.

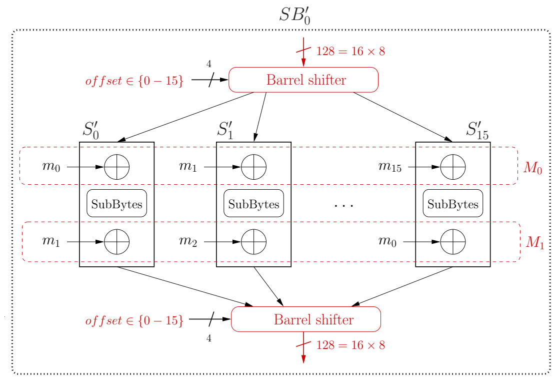

Figure 2: Organization of 16 different S' in RSM countermeasure.

- At the beginning of the encryption, the mapping between the 4×4 bytes and the 16 masked sboxes is chosen according to a randomly generated offset;

- Then, to protect also the sbox input, one output mask M_{out} can be chosen as the input mask M_{in} for the neighbor sbox; thus at the next round, all the sboxes S' are rotated in a circular manner by one position. This allows to change the masks for the next round, as in the "rotating sboxes masking" (RSM [20, 22]) countermeasure for AES.

Thus, in this case, the countermeasure is at constant cost. In the previous implementation of this RSM countermeasure (refer to [20, 22]), only a protection against first- and second-order attacks was sought. As we now know that a 3-CI function of 8 variables and of minimal weight 16 exists, we can improve (at a constant cost) the security of RSM on AES to resist also third-order attacks, simply by choosing for the mask values the codewords of $[8, 4, 3]$.

3 Low-weight d-CI Boolean Functions

In this section, we first give general results about d-CI functions, then detail some (non-optimal) constructive techniques (linear codes indicator, Maiorana-McFarland construction and orthogonal arrays), and finally present an exhaustive search technique to find d-CI functions. Eventually, a summary of the minimal weight of d-CI functions of $n \leq 10$ variables is given with constructions.

3.1 General Results about d-CI Functions

We recall in this section known facts, for the article to be self-contained.

Example 1. *The only $\mathbb{F}_2^n \to \mathbb{F}_2$ Boolean functions that are n-CI are the constant functions. Indeed, let us assume that $\forall a \neq 0$, $\widehat{f}(a) = 0$. Thus $\forall a, \widehat{f}(a) = \widehat{f}(0)\delta(a)$. Now, $\widehat{\widehat{f}} = 2^n f$ and $\widehat{\delta} = 1$, so: $\forall a, f(a) = \widehat{f}(0)/2^n$. But $f(a) \in \{0, 1\}$. Thus either $\widehat{f}(0) = 0$ or $\widehat{f}(0) = 2^n$, which is equivalent to having either $f = 0$ or $f = 1$.*

Example 2. *There exists a Boolean function that is $(n-1)$-CI and of weight 2^{n-1},* i.e. *a function that is $(n-1)$-resilient. As a matter of fact, the function $f(x) = \bigoplus_{i=1}^{n} x_i$ is $(n-1)$-CI and has weight $\widehat{f}(0) = 2^{n-1}$. Indeed, $\forall a \neq 0$, $\widehat{(-1)^f}(a) = \sum_x (-1)^{1\cdot x \oplus a \cdot x} = \sum_x (-1)^{\neg a \oplus x}$, that is equal to zero if and only if $\neg a \neq 0 \iff a \neq 1 \iff w_H(a) \neq n$. So, for all a of Hamming weight strictly less than n, f has a null Fourier transform, which means that f is $(n-1)$-CI.*

Lemma 2. *The support of a d-CI ($d > 0$) function has a cardinality divisible by 2^d. Incidentally, the weight of a d-CI function is divisible by more than 2^d, depending on the algebraic degree*[2] *of f.*

Proof. The proof is a specific case where $l = 0$ in Theorem 6 of [6]. Let us note a typographic mistake in this theorem: if the algebraic degree of f is noted $m \geq 1$, the weight of f is divisible by $2^{d+\lfloor \frac{n-d-1}{m} \rfloor}$ and not $2^{n-1} - 2^{d+\lfloor \frac{n-d-1}{m} \rfloor}$ as erroneously written in [6]. We notice that the algebraic degree is bounded by $n - d$ [26]. □

3.2 General Results about d-CI Functions of Lowest-Weight

In this section, we give some properties of nonzero d-CI functions of lowest weight. We denote:

- $\mathcal{D}_{n,d}$ the set of d-CI nonzero functions of n variables,
- $w_{n,d}$ the lowest weight of d-CI nonzero functions of n variables (elements of $\mathcal{D}_{n,d}$). According to Lemma 2, 2^d divides $w_{n,d}$.

Lemma 3. *Let $0 < d \leq n$. Then $w_{n,d-1} \leq w_{n,d}$.*

Proof. As $\mathcal{D}_{n,d} \subseteq \mathcal{D}_{n,d-1}$,

$$w_{n,d} = \min_{f \in \mathcal{D}_{n,d}} \mathsf{Card}[\mathrm{supp}(f)] \geq \min_{f \in \mathcal{D}_{n,d-1}} \mathsf{Card}[\mathrm{supp}(f)] = w_{n,d-1} \enspace .$$

□

Lemma 4. *Let $d < n$. Then $w_{n+1,d} \leq 2w_{n,d}$.*

Proof. Let $f \in \mathcal{D}_{n,d}$. Then, the new function $\tilde{f}(x_1, \cdots, x_n, x_{n+1}) = f(x_1, \cdots, x_n) \in \mathcal{D}_{n+1,d}$, and has weight twice that of f because its truth table is the concatenation of twice that of f. □

[2] A Boolean function f can be uniquely written in an algebraic normal form as $f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u \prod_{j=1}^{n} x_j^{u_j}$. The algebraic degree of f is defined as $d^\circ f = \max\{w_H(u)\,;a_u \neq 0\}$.

Lemma 5. *Let $d, n > 1$. Then $w_{n-1,d-1} \leq \frac{1}{2} w_{n,d}$.*

Proof. Let $f \in \mathcal{D}_{n,d}$. Let us denote $(a, b) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2$ and $(x, y) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2$. As f is d-CI, if $0 < w_H(a, b) \leq d$, $\sum_{(x,y)} (-1)^{f(x,y) \oplus a \cdot x \oplus b \cdot y} = 0$. So we have:

- When $b = 0$, for all $0 < w_H(a) \leq d$, $\sum_x (-1)^{f(x,0) \oplus a \cdot x} + \sum_x (-1)^{f(x,1) \oplus a \cdot x} = 0$;
- When $b = 1$, for all $0 < w_H(a) \leq d-1$, $\sum_x (-1)^{f(x,0) \oplus a \cdot x} - \sum_x (-1)^{f(x,1) \oplus a \cdot x} = 0$.

Thus, for all $0 < w_H(a) \leq d$, we have $\sum_x (-1)^{f(x,0) \oplus a \cdot x} = 0$ (sum of the two previous relations). Therefore, $x \mapsto f(x, 0)$ is $(d-1)$-CI, and of weight half that of f. □

3.3 Constructions of d-CI Functions with Codes

3.3.1 Relationship between Correlation-Immune Functions and Codes

By definition, the dual distance of a code $C \subseteq \mathbb{F}_2^n$ is equal to the maximal number $d_C^\perp$ such that there is no monomial of degree relative to Y strictly smaller than $d_C^\perp$ in $D_C(X+Y, X-Y)$, where D_C is the distance enumerator polynomial of C: $D_C(X, Y) = \frac{1}{\mathsf{Card}[C]} \sum_{x,y \in C} X^{n - w_H(x \oplus y)} Y^{w_H(x \oplus y)}$. When the code C is linear (*i.e.* C is a linear subspace of the vector space $\mathbb{F}_2^n$), its minimum distance is equal to $\min\{w_H(x), \forall x \in C^*\}$, and the distance enumerator, which equals then the weight enumerate W_C (defined as $W_C(X, Y) = \frac{1}{\mathsf{Card}[C]} \sum_{x \in C} X^{n - w_H(x)} Y^{w_H(x)}$), satisfies $W_C(X+Y, X-Y) = \mathsf{Card}[C] W_{C^\perp}(X, Y)$, where the dual distance of C is minimum distance of $C^\perp$.

Lemma 6. *Let f be a nonzero Boolean function $\mathbb{F}_2^n \to \mathbb{F}_2$. f is d-CI if and only if the set $C = supp(f)$ is a subcode of $\mathbb{F}_2^n$ of dual distance equal to or greater than $d+1$.*

Proof. This lemma is an immediate application of the MacWilliams' identity [12]:

$$\begin{aligned} D_C(X+Y, X-Y) &= \frac{1}{\mathsf{Card}[C]} \sum_{x,y \in C} \sum_{z \in \mathbb{F}_2^n} X^{n - w_H(z)} Y^{w_H(z)} (-1)^{z \cdot (x \oplus y)} \\ &= \frac{1}{\mathsf{Card}[C]} \sum_{z \in \mathbb{F}_2^n} X^{n - w_H(z)} Y^{w_H(z)} \left(\sum_{x \in C} (-1)^{z \cdot x} \right)^2 . \end{aligned}$$

The result comes from the fact the coefficient of $X^{n-i} Y^i$ is a sum of squares that can be null only if all the terms are null. □

Remark 5. *When $C = \mathbb{F}_2^n$, the dual distance C is strictly greater than n; we consider in the sequel it is equal to $n + 1$.*

3.3.2 Construction of d-CI Boolean Functions based on Linear Codes

Lemma 7. *Let n and $1 \leq d < n$. Let $k_{max}(n, d)$ be the largest dimension of a binary linear code $[n, k_{max}(n, d), d + 1]$. The lowest weight of d-CI functions is upper bounded by $2^{n-k_{max}(n,d)}$.*

Proof. The dual of the linear code $[n, k_{\max}(n, d), d + 1]$ has parameters $[n, n - k_{\max}(n, d)]$ and dual distance $d+1$. According to Lemma 6, the indicator of this code is d-CI. Its weight is two to the power of its dimension, *i.e.* $2^{n-k_{\max}(n,d)}$. Therefore, the lowest weight of d-CI functions is smaller than or equal to this quantity. □

Corollary 2. *The weight $w_{n,d}$ reaches its minimum 2^d if and only if a binary linear maximum distance separable (MDS) code of length n and of minimum distance $d + 1$ exists.*

Proof. In Lemma 7, we have $k_{\max}(n, d) \leq n - d$ because of the Singleton bound [12]. This bound is tight for linear codes over finite fields (but not by binary codes, in general, see below) and attained by the MDS codes. If such binary MDS codes exist then there exist d-CI functions of weight $2^{n-k_{\max}(n,d)} = 2^d$. Conversely, the maximal degree of a d-CI function f is $n - d$ [26]. Thus f is the indicator of an affine space of dimension $n - d$ [12]. Without loss of generality, we assume it is a vectorial space. So it is a linear code of parameters $[n, d]$ and of dual distance at least $d + 1$, whose dual has parameters $[n, n - d, d + 1]$ (*i.e.* reaches the Singleton bound, thus is MDS). Therefore the weight $w_{n,d} = 2^d$ is attained only by the cosets of linear MDS codes. □

Remark 6. *The Singleton bound exists for unrestricted (that is, linear or non-linear) codes: given its length n and its minimum distance, say d_1, the size of such a code is at most 2^{n-d_1+1}. The fact that every code of dual distance at least $d + 1$ has a size divisible by 2^d, and therefore has size at least 2^d, can be viewed as an equivalent of the Singleton bound dealing with the dual distance. It is nice to see that the same (linear) MDS codes attain both bounds. This remark is also given as Theorem 4.21, due to Delsarte (1973), in [10, §4.5, page 79].*

Example 3. *As the repetition code $[n, 1, n]$ and its dual $[n, n-1, 2]$ are MDS, we have $w_{n,n-1} = 2^{n-1}$ and $w_{n,1} = 2$. Incidentally, those are the only cases where Corollary 2 can be used, since all binary MDS codes are trivial (*i.e.* *either the empty code, the parity-check code, the universe code — all $\mathbb{F}_2^n$ — or the repetition code).*

Remark 7. *For all $1 < d < n - 1$, $w_{n,d} \geq 2^{d+1}$. Indeed, lemma 2 states that a nonzero d-CI function has weight at least 2^d. When $n = d + 2$ and $n \geq 4$, the bound is reached,* i.e. *the lowest weight $w_{n,n-2}$ of d-CI functions is exactly $2^{d+1} = 2^{n-1}$. The reason is that the maximum algebraic degree is $n - (n - 2) - 1 = 1$. Thus these are affine functions and naturally the weight will be 2^{n-1}.*

3.3.3 Construction of d-CI Boolean Functions based on Maiorana-McFarland Construction

Maiorana-McFarland's construction [7, §7.5.1] allows to build Boolean resilient functions. The dimension n is split into $n = r + s$, where $r \neq 0$ and $s \neq 0$. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is written as $f(u, v) = u \cdot \phi(v) \oplus g(v)$, where $\phi : \mathbb{F}_2^s \to \mathbb{F}_2^r$ and $g : \mathbb{F}_2^s \to \mathbb{F}_2$ are arbitrary functions.

The Walsh transform of a Maiorana-McFarland function f in $x \in \mathbb{F}_2^n$, noted $x = (a, b) \in \mathbb{F}_2^r \times \mathbb{F}_2^s$, is equal to:

$$\begin{aligned} W_f(a,b) &= \sum_{u,v} (-1)^{u\cdot\phi(v)\oplus g(v)\oplus a\cdot u\oplus b\cdot v} \\ &= \sum_{v} (-1)^{g(v)\oplus b\cdot v} \sum_{u} (-1)^{u\cdot(\phi(v)\oplus a)} \\ &= 2^r \sum_{v\in\phi^{-1}(a)} (-1)^{g(v)\oplus b\cdot v} \quad , \end{aligned} \tag{4}$$

because $\sum_u (-1)^{u\cdot(\phi(v)\oplus a)}$ is equal to 0 but when $\phi(v) \oplus a = 0$. If every element in $\phi(\mathbb{F}_2^s)$ has Hamming weight strictly greater than d, then f is (at least) d-resilient.

The same template $f(u, v) = u \cdot \phi(v) \oplus g(v)$ can be used to design correlation-immune functions. Let us illustrate it with the following assumptions:

- g is null;
- for every v, $\phi(v)$ is either the null vector[3] or has a Hamming weight strictly greater than d, and
- $\phi^{-1}(0)$ is an affine space $w + E$ such that $E^\perp$ has minimum distance at least $d + 1$.

Then the Maiorana-McFarland function is at least d-CI. The reason is that:

- If $a \neq 0$, then $\phi^{-1}(a) = \emptyset$ unless $w_H(a) > d$. Thus, for all a such that $1 \leq w_H(a) \leq d$, $W_f(a, b) = 0$ (whatever b).
- If $a = 0$, then $b \neq 0$ since $(a, b) \neq (0, 0)$. Thus $\sum_{v\in\phi^{-1}(a)} (-1)^{g(v)\oplus b\cdot v} = \sum_{v\in w+E} (-1)^{b\cdot v} = (-1)^{b\cdot w} \mathsf{Card}[E] \delta_{E^\perp}(b) = 0$ because $w_H(b) \leq d$ and $E^\perp$ has minimum distance strictly greater than d by hypothesis.

3.4 Deriving d-CI Boolean Functions from Binary Orthogonal Arrays

In this section, we recall known facts developed in [10], that help transport properties of orthogonal arrays (OAs) without multiple rows to d-CI functions.

[3] It is necessary to have zero in the image of ϕ for f to be of weight strictly smaller than 2^{n-1}.

Definition 4. *An orthogonal array $OA(w, n, s = 2, d)$ is a $w \times n$ binary (because $s = 2$) array of w rows and n columns, such that every subarray of size $w \times d$ contains the elements of $\mathbb{F}_2^d$ an equal number of times. The OA is said to be of strength d.*

Remark 8. *Orthogonal arrays can have identical lines. In this paper, we are interested in* simple *OAs,* i.e. *OAs whose lines are different.*

Theorem 2. *If C is a binary (n, w) code with dual distance $d + 1$, then the corresponding OA is $OA(w, n, 2, d)$. Conversely, the code corresponding to an $OA(w, n, 2, d)$ is a (n, w) code with dual distance greater than or equal to $d + 1$. If the orthogonal array has strength d but not $d + 1$, then the dual distance of the code is precisely $d + 1$.*

Proof. It is the Theorem 4.9 of [10, p. 70]. □

Corollary 3. *The minimum number of rows w of an $OA(w, n, 2, d)$ is smaller than or equal to the minimum weight of a d-CI function. They are identical if and only if there exists a minimal $OA(w, n, 2, d)$ whose rows are unique.*

Proof. This is a direct application of Theorem 2, with the link between d-CI functions and codes of dual distance $d + 1$ given by Lemma 6. Since not all OAs are simple, the minimum number of rows w of an OA is smaller or equal to the minimum weight of a d-CI function. This distinctive feature is discussed in App. C. There is equality if and only if there exists a minimal $OA(w, n, 2, d)$ whose rows are unique. □

It is proved in [10, §4.5] that the number of rows of an OA can be lower-bounded by a linear programming problem (Delsarte LP bound). The numerical values are given in Tab. 1 for all $n \leq 13$.

Lemma 8. *A lower bound for the minimal weight of a d-CI function can also be found in Tab. 1. Indeed, Corollary 3 states that the minimal weight of a d-CI function is greater than the minimal number of rows in an OA, that is in turn greater than the optimal solution of Delsarte linear programming problem.*

3.5 A Search for Correlation-Immune Boolean Functions with SMT

Satisfiability modulo theory (SMT) tools solve SAT problems within a domain-specific theory. Like SAT-solvers, SMT require the problem to be written as clauses. Unlike SAT-solvers, higher-level constructions can be used.

A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is d-CI if it satisfies:

$$\begin{aligned} &\forall a \in \mathbb{F}_2^n, 1 \leqslant w_H(a) \leqslant d, \\ &\widehat{f}(a) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{a \cdot x} = 0 \qquad \Longleftrightarrow \\ &\sum_{x \in \mathbb{F}_2^n} f(x) \wedge (a \cdot x) = \frac{1}{2}\widehat{f}(0) \ . \end{aligned} \tag{5}$$

Table 1: Lower bounds on $w_{n,d}$ obtained by the Delsarte LP algorithm.

d n	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2												
2	2	4											
3	2	4	8										
4	2	6	8	16									
5	2	8	12	16	32								
6	2	8	16	32	32	64							
7	2	8	16	48	64	64	128						
8	2	10	16	64	88	112	128	256					
9	2	12	20	96	128	192	224	256	512				
10	2	12	24	96	192	320	384	512	512	1024			
11	2	12	24	96	192	512	640	1024	1024	1024	2048		
12	2	14	24	112	176	768	1024	1536	1792	2048	2048	4096	
13	2	16	28	128	224	1024	1536	2560	3072	3584	4096	4096	8192

Indeed, $(-1)^{a\cdot x} = 1 - 2(a \cdot x)$, and $(a \cdot x)$ can be seen both as an integer living in $\{0, 1\}$ or as an element of $\mathbb{F}_2$.

Concretely, we are looking for the existence of functions of a given weight $\widehat{f}(0) = w$. As recalled in Lemma 2, w must be a multiple of 2^d. There are 2^n unknown literals, noted $f(x)$, $x \in \mathbb{F}_2^n$, and the problem consists in satisfying simultaneously $\sum_{i=0}^{d} \binom{n}{i}$ clauses:

$$\left(\sum_{x\in\mathbb{F}_2^n} f(x) = w\right) \wedge \bigwedge_{\substack{a\in\mathbb{F}_2^{n\,\star},\\ w_H(a)\leqslant d}} \left(\sum_{\substack{x\in\mathbb{F}_2^n,\\ a\cdot x=0}} f(x) = \frac{1}{2}w\right) . \qquad (6)$$

For example, with $n = 3$ and $d = 2$, we get the following condition, expressed in conjunctive normal form (CNF):

$$\begin{array}{rll}
 & (\, f(7)+f(6)+f(5)+f(4)+f(3)+f(2)+f(1)+f(0) = w \;) & \\
\wedge & (\, f(7)+f(5)+f(3)+f(1) = w/2\;) & [\text{a=1}] \\
\wedge & (\, f(7)+f(6)+f(3)+f(2) = w/2\;) & [\text{a=2}] \\
\wedge & (\, f(7)+f(6)+f(5)+f(4) = w/2\;) & [\text{a=4}] \\
\wedge & (\, f(6)+f(5)+f(2)+f(1) = w/2\;) & [\text{a=3}] \\
\wedge & (\, f(6)+f(4)+f(3)+f(1) = w/2\;) & [\text{a=5}] \\
\wedge & (\, f(5)+f(4)+f(3)+f(2) = w/2\;)\,. & [\text{a=6}]
\end{array}$$

The algorithm to find d-CI Boolean functions of low-weight w is given in Alg. 1: the possible values for w can tested in ascending order until the Eqn. (6) becomes satisfiable.

Algorithm 1: Method to find the lowest weight of d-CI functions.

```
input  : $n$, the number of variables of the function $f$
input  : $d$, the order of correlation-immunity
output: $w_{n,d}$, the minimal weight of $d$-CI functions of $n$ variables
for $w \in \{i \times 2^d; i \in [\![1, 2^{n-d}]\!]\}$ do // See Lemma 2 for the steps of $w$
    if Eqn. (6) is unsatisfiable then
        // No $d$-CI $f$ of weight $w$ exists
    else
        return $w$ // Yields at the same time the lowest weight
        and an example of $f$
    end
end
```

SMT implement various theories. The one that specifically fits our needs is `QF_BV`, that handles arithmetic and logic operations on bitvectors of fixed size.

The SMT actually feature many advantages compared to SAT-solvers. Our problem of Eqn. (6) implies sums and equality tests. They can be expressed as two cardinality constraints in SAT-solvers [20], in an obviously suboptimal way, whereas these operations are captured natively and at high-level in SMT. Therefore SMT tools produce optimized CNF formulas whose satisfiability is easier to check than hand-written CNF formulas. For instance, one SAT-solver (`cryptominisat` [27]) exhausted all the 16 GB RAM of a computer when trying to solve the problem $n = 8$ and $d = 4$ with cardinality constraints for the equality. Now, an SMT succeeds within one to two minutes with a couple of tens of MB of RAM only. The reason is that the SMT finds high-level simplifications of the problem. Furthermore, with SMT, the problem is formulated in human-readable code, as the solution.

The SMT program (written in LISP and compliant with SMT-LIB2 format) can be generated automatically. For instance, the lowest weight Boolean functions from $\mathbb{F}_2^3$ to $\mathbb{F}_2$ of second-order correlation-immunity ($n = 3$, $d = 2$) have Hamming weight 4. A script that finds a solution is given in Tab. 2.

The Hamming weight computation is based on a divide-and-conquer approach, for instance explained in [11]. The parallel with the CNF expression given at page 17 is clear.

3.6 Summary of Minimally Weighted d-CI Functions

Table 3 lists, (in bold and in italic), the exact minimal cardinality of $\mathrm{supp}(f)$, where $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is d-CI. It lists values of n from 1 to 13, with complete results for n up to 10.

The figures in **bold** have been obtained by the algorithm 1. The resolution of the satisfiability problem can last several days. Let us comment on the most computing-intensive entry of Tab. 3 computed by SMT. The minimal weight when $n = 10$ and $d = 6$ is $w_{10,6} = 512$. Because of the LP bound (Lemma 8),

Table 2: Script that finds (if it exists) a 2-CI Boolean function of 3 variables and weight 4.

```
(set-logic QF_BV)
(set-info :smt-lib-version 2.0)
(set-option :produce-models true)

(declare-fun f () (_ BitVec 8)); The 2-CI function, our unknown (8 literals)

; Sub-functions for the Hamming weight
(define-fun w_H0 ((x (_ BitVec 8))) (_ BitVec 8)
                  (bvadd (bvand         x       #x55 )
                         (bvand (bvlshr x #x01) #x55 )))
(define-fun w_H1 ((x (_ BitVec 8))) (_ BitVec 8)
                  (bvadd (bvand         x       #x33 )
                         (bvand (bvlshr x #x02) #x33 )))
(define-fun w_H2 ((x (_ BitVec 8))) (_ BitVec 8)
                  (bvadd (bvand         x       #x0f )
                         (bvand (bvlshr x #x04) #x0f )))

; The complete Hamming weight (noted w_H)
(define-fun w_H ((x (_ BitVec 8))) (_ BitVec 8) (w_H2 (w_H1 (w_H0 x))) )

; Our problem (7 clauses: 1 for w_H(a)=0, 3 for w_H(a)=1 and 3 for w_H(a)=2)
(assert (= (w_H        f      ) #x04)); [a=#x00]
(assert (= (w_H (bvand f #xaa)) #x02)); [a=#x01]
(assert (= (w_H (bvand f #xcc)) #x02)); [a=#x02]
(assert (= (w_H (bvand f #xf0)) #x02)); [a=#x04]
(assert (= (w_H (bvand f #x66)) #x02)); [a=#x03]
(assert (= (w_H (bvand f #x5a)) #x02)); [a=#x05]
(assert (= (w_H (bvand f #x3c)) #x02)); [a=#x06]

(check-sat); Answers ‘sat’
(get-value (f)); Gives one solution, here ‘#x96’ (10010110 in binary)
```

we knew that $w_{10,6} \geq 320$ (see Tab. 1). Furthermore, this figure must be a multiple of 2^6 (see Lemma 2), and smaller than or equal to $w_{10,7} = 512$. Thus, there are only four possible solutions: $5 \times 2^6 = 320$, $6 \times 2^6 = 384$, $7 \times 2^6 = 448$, or $8 \times 2^6 = 512$. We led the computation on a 2.33 GHz server with 4 MB cache and 16 GB of RAM. A program similar to that of Tab. 2 (but optimized) revealed that:

- the problem with weight 320 is found unsatisfiable in 1.5 hour,
- the problem with weight 384 is found unsatisfiable in about 3 days and 4 hours (the script was optimized),
- the problem with weight 448 is found unsatisfiable in about 1 day and 9 hours,
- the problem with weight 512 is found satisfiable in 1 hour (but this step was not necessary, since one solution for $d = 7$ was known and of the same weight – hence Lemma 3 applied).

The optimization consisted in implementing Hamming weight calculations according to the vector size. Instead of keeping the data on 2^n bits, a new Hamming weight function uses 2^{n-1} bit inputs[4].

The figures in italic font have been deduced from various lemmas. They are marked by a letter, that refers to the explanations below:

[a] The weights for $w_{n,n} = 2^n$ for $d = n$ result from Example 1;

[b] The weights for $d = n - 1$ result from Lemma 2 ($w_{n,n-1}$ is a nonzero multiple of 2^{n-1}), and from Example 2 (one solution of weight 2^{n-1} exists). Those weights $w_{n,n-1} = 2^{n-1}$ are also given in Example 3.

[c] The weights for $d = n - 2$ results from Remark 7.

[d] According to Lemma 5, the weight for $n = 10$ and $d = 7$ is greater than twice $w_{9,6}$. Thus $w_{10,7} \geq 256 \times 2$ ($w_{9,6} = 256$) was found by the SMT. But also, $w_{10,7} \leq w_{10,8} = 2^{n-1} = 512$ because of Remark 7. Thus $w_{10,7} = 2^{n-1}$.

[e] Similarly, by recursion, we can show that $w_{n,n-3} = 2^{n-1}$ starting from $n = 7$ onwards.

[f] Similarly, by recursion, we can show that $w_{n,n-4} = 2^{n-1}$ starting from $n = 10$ onwards.

[g] The weight for $n = 13$ and $d = 3$ is lower bounded by Delsarte LP to 28 (*cf.* Tab. 1). In addition, this value must be a multiple of $2^3 = 8$, it is thus greater of equal to 32. Now, code $[13, 13 - \log_2 32 = 8, 4]$ has distance 4 [13], hence it is the support of one solution (Lemma 7).

[4] The script is available online at: http://perso.enst.fr/guilley/dCI/dCI_z3_opt.py.

[h] The weight for $n = 11$ and $d = 6$ is lower bounded by $2 \times w_{10,5} = 512 = 2^9$ (Lemma 5). Now, code $[11, 11 - \log_2 512 = 2, 7]$ has distance 7 [13], hence it is the support of one solution.

[i] The weight for $n = 12$ and $d = 7$ is lower bounded by $2 \times w_{11,6} = 1024 = 2^{10}$ (result above and Lemma 5). Now, code $[12, 12 - \log_2 1024 = 2, 8]$ has distance 8 [13], hence it is the support of one solution.

The values contributed by the authors are underlined. In particular, the case of $n = 9$ can have an application in cryptography when the sbox fanin is 9 bits, as is the case for MISTY [14] and KASUMI [15] (used in GSM, GPRS and UMTS mobile communications systems).

Eventually, unknown values are indicated by question marks. They correspond to the lack of mathematical proofs for the lowest weight and to the failure of the SMT to converge fast enough (within a couple of days). A single question mark indicates that the exact lower value for OA is tabulated in a Tab. 12.1 of [10]. A triple question mark indicates that the value is completely unknown.

The truth tables of the functions can be found in an online document at: http://perso.enst.fr/guilley/dCI/index.html.

4 Conclusion

In this article, a masking scheme based on randomly selected “hardcoded” masked sboxes is analyzed under the view of high-order zero-offset attacks. We have explained that the indicator of the masks shall meet two contradictory properties: low weight and high correlation-immunity order. Our results quantify the trade-off between order of resistance (that corresponds to the order of correlation-immunity) and the number of values taken by the mask. In particular, we explain that it is possible with only 16 masks to protect AES against attacks of orders 1, 2 and 3, while the state-of-the-art was a protection at orders 1 and 2 only. We have identified, thanks to an SMT, correlation-immune functions of lowest possible weight. In particular, we provided all the minimal weights for functions up to 10 variables, along with a construction for the functions.

Acknowledgments

The authors thank Patrick Solé for interesting discussions and relevant advices.

References

[1] Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.X., Veyrat-Charvillon, N.: Mutual Information Analysis: a Comprehensive Study. J. Cryptology **24**(2), 269–291 (2011)

Table 3: Minimal value $w_{n,d}$ of the cardinal of supp(f), where $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is d-CI. The cells in gray highlight functions of minimal weight not power of two, that are discussed in Appendix B.

d n	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**2**												
2	**2**	**4**											
3	**2**	**4**	**8**										
4	**2**	**8**	**8**	**16**									
5	**2**	**8**	**16**	**16**	**32**								
6	**2**	**8**	**16**	**32**	**32**	**64**							
7	**2**	**8**	**16**	**64**	**64**	**64**	**128**						
8	**2**	**12**	**16**	**64**	**128**	**128**	**128**	**256**					
9	**2**	**12**	**24**	**128**	**128**	**256**	**256**	**256**	**512**				
10	**2**	**12**	**24**	**128**	**256**	**512**	*512*^{d}	*512*^{c}	*512*^{b}	*1024*^{a}			
11	**2**	**12**	**24**	???	???	*512*^{h}	*1024*^{f}	*1024*^{e}	*1024*^{c}	*1024*^{b}	*2048*^{a}		
12	**2**	**16**	**24**	???	???	???	*1024*^{i}	*2048*^{f}	*2048*^{e}	*2048*^{c}	*2048*^{b}	*4096*^{a}	
13	**2**	**16**	*32*^{g}	???	???	?	???	???	*4096*^{f}	*4096*^{e}	*4096*^{c}	*4096*^{b}	*8192*^{a}

[2] Bierbrauer, J., Gopalakrishnan, K., Stinson, D.R.: Bounds for Resilient Functions and Orthogonal Arrays. In: Y. Desmedt (ed.) CRYPTO, *Lecture Notes in Computer Science*, vol. 839, pp. 247–256. Springer (1994)

[3] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: CHES, *LNCS*, vol. 4727, pp. 450–466. Springer (2007). Vienna, Austria

[4] Brier, É., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: CHES, *LNCS*, vol. 3156, pp. 16–29. Springer (2004). Cambridge, MA, USA

[5] Camion, P., Carlet, C., Charpin, P., Sendrier, N.: On Correlation-Immune Functions. In: J. Feigenbaum (ed.) CRYPTO, *Lecture Notes in Computer Science*, vol. 576, pp. 86–100. Springer (1991)

[6] Carlet, C.: On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions. In: Sequences and their Applications (SETA), Discrete Mathematics and Theoretical Computer Science, pp. 131–144. Springer-Verlag (2001). Bergen, Norway

[7] Carlet, C.: Boolean Functions for Cryptography and Error Correcting Codes: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering. pp. 257–397. Cambridge University Press, Y. Crama and P. Hammer eds (2010). Preliminary version available at http://www.math.univ-paris13.fr/~carlet/chap-fcts-Bool-corr.pdf

[8] Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: CHES, 10th International Workshop, *Lecture Notes in Computer Science*, vol. 5154, pp. 426–442. Springer (2008). Washington, D.C., USA

[9] Hedayat, A.S., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays, Theory and Applications. Springer series in statistics. Springer, New York (1999). ISBN 978-0-387-98766-8

[10] Lauradoux, C., Dalke, A.: Hamming weight (2009). Research report available at:
http://perso.citi.insa-lyon.fr/claurado/ham/overview.pdf

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

[12] Markus Grassl: Code Tables: Bounds on the parameters of various types of codes. Universität Karlsruhe, http://www.codetables.de/

[13] Matsui, M.: New Block Encryption Algorithm MISTY. In: E. Biham (ed.) FSE, *Lecture Notes in Computer Science*, vol. 1267, pp. 54–68. Springer (1997)

[14] Matsui, M., Tokita, T.: Data Transformation Apparatus and Data Transformation Method (2006). US patent 7096369

[15] Messerges, T.S.: Power Analysis Attacks and Countermeasures for Cryptographic Algorithms. Ph.D. thesis, University of Illinois at Chicago, USA (2000). 468 pages

[16] Messerges, T.S.: Using Second-Order Power Analysis to Attack DPA Resistant Software. In: CHES, *LNCS*, vol. 1965, pp. 238–251. Springer-Verlag (2000). Worcester, MA, USA

[17] Moradi, A.: Statistical tools flavor side-channel collision attacks. In: D. Pointcheval, T. Johansson (eds.) EUROCRYPT, *Lecture Notes in Computer Science*, vol. 7237, pp. 428–445. Springer (2012)

[18] Moradi, A., Mischke, O.: How Far Should Theory be from Practice? Evaluation of a Countermeasure. In: CHES (2012). Leuven, Belgium

[19] Nassar, M., Guilley, S., Danger, J.L.: Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks. In: INDOCRYPT, *LNCS*, vol. 7107, pp. 22–39. Springer (2011). Chennai, Tamil Nadu, India. DOI: 10.1007/978-3-642-25578-6_4

[20] Nassar, M., Guilley, S., Danger, J.L.: Formal Analysis of the Entropy / Security Trade-off in First-Order Masking Countermeasures against Side-Channel Attacks — Complete version. Cryptology ePrint Archive, Report 2011/534 (2011). http://eprint.iacr.org/2011/534

[21] Nassar, M., Souissi, Y., Guilley, S., Danger, J.L.: RSM: a Small and Fast Countermeasure for AES, Secure against First- and Second-order Zero-Offset SCAs. In: DATE, pp. 1173–1178 (2012). Dresden, Germany. (TRACK A: "Application Design", TOPIC A5: "Secure Systems")

[22] NIST/ITL/CSD: Advanced Encryption Standard (AES). FIPS PUB 197 (2001). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[23] Prouff, E., Rivain, M.: A Generic Method for Secure SBox Implementation. In: S. Kim, M. Yung, H.W. Lee (eds.) WISA, *Lecture Notes in Computer Science*, vol. 4867, pp. 227–244. Springer (2007)

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

[25] Siegenthaler, T.: Correlation-immunity of nonlinear combining functions for cryptographic applications. IEEE Transactions on Information Theory **30**(5), 776–780 (1984)

[26] Soos, M., Nohl, K., Castelluccia, C.: Extending SAT Solvers to Cryptographic Problems. In: O. Kullmann (ed.) SAT, *Lecture Notes in Computer Science*, vol. 5584, pp. 244–257. Springer (2009)

[27] University of Sydney: Magma Computational Algebra System. http://magma.maths.usyd.edu.au/magma/

[28] Veyrat-Charvillon, N., Standaert, F.X.: Mutual Information Analysis: How, When and Why? In: CHES, *LNCS*, vol. 5747, pp. 429–443. Springer (2009). Lausanne, Switzerland

[29] Waddle, J., Wagner, D.: Towards Efficient Second-Order Power Analysis. In: CHES, *LNCS*, vol. 3156, pp. 1–15. Springer (2004). Cambridge, MA, USA

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

A Properties of the Fourier Transform of the Powers of the Hamming Weight Function

The quantity $\widehat{w_H^d}(z)$ is central to the computations about the RSM masking scheme described in Sec. 2. For the safe of simplicity, we begin with the study of $\widehat{\tilde{w_H}^d}(z)$ in App. A.1; The function $\tilde{w_H}$ is the centered Hamming weight, defined as $\tilde{w_H}(x) \doteq w_H(x) - n/2$, where x is a bitvector belonging to $\mathbb{F}_2^n$. Afterwards, we continue with the study of $\widehat{w_H^d}(z)$ in App. A.2.

A.1 Properties of $\widehat{\tilde{w_H}^d}(z)$

First of all, this lemma is proved:

Lemma 9. *(Invariance in bits reordering).* $\widehat{\tilde{w_H}^d}(a)$ *depends only on* $w_H(a)$.

Proof. It is well-known that the Walsch transform of a symmetric function is symmetric (that is, invariant under permutation of the input coordinates). □

Some numerical values are given in Tab. 4. Then, we state Lemma 10.

Lemma 10. $\widehat{\tilde{w_H}^d}(z)$ *has the following properties:*

P1: $\widehat{\tilde{w_H}^d}(z) = 0$ *if* $w_H(z) > d$*;*

P2: $\widehat{\tilde{w_H}^d}(z) = (-1)^d 2^{n-d} d!$ *if* $w_H(z) = d$*;*

P3: $\widehat{\tilde{w_H}^d}(z) = 0$ *if* $|d - w_H(z)| \in 2\mathbb{N} + 1$ *(*i.e. *d and $w_H(z)$ have different parities); In particular,* $\widehat{\tilde{w_H}^{w_H(z)+1}}(z) = 0$.

P4: When $\widehat{\tilde{w_H}^d}(z)$ *is nonzero, it has sign* $(-1)^d$ *(or equivalently sign of* $(1)^{w_H(z)}$*).*

Proof. We recall that $\tilde{w_H}(z) = -\frac{1}{2}\sum_{i=0}^{n-1}(-1)^{z_i}$. Thus the Fourier transform of the dth power of the centered Hamming weight is $\widehat{\tilde{w_H}^d}(z) = \left(-\frac{1}{2}\right)^d \sum_{x\in\mathbb{F}_2^n}\left(\sum_{i=0}^{n-1}(-1)^{x_i}\right)^d(-1)^{x\cdot z}$. Now, the term $\left(\sum_{i=0}^{n-1}(-1)^{z_i}\right)^d$ can be developed as a sum of weighted quantities that have the form: $(-1)^{\bigoplus_{i\in I,\mathsf{Card}[I]\in\{d,d-2,\dots\}}x_i}$; the weights themselves are multinomial coefficients – their value is irrelevant for this proof, but we simply note that they are all positive. If z has a Hamming weight strictly greater than d, then in $x\cdot z$, there will always be a component of x, say x_j, such that $j\notin I$. Therefore $\sum_x(-1)^{\bigoplus_{i\in I}x_i}(-1)^{x\cdot z} = \sum_{x_0,x_1,\cdots,x_{j-1},x_{j+1},\cdots,x_{n-1}}(-1)^{\bigoplus_{i\in I}x_i}(-1)^{x\cdot z}\times\sum_{x_j}(-1)^{x_j} = 0$ (because $\sum_{x_j}(-1)^{x_j} = 0$). This proves that $\widehat{\tilde{w_H}^d}(z) = 0$ if $w_H(z) > d$ (property P1).

Now, let us examine the case where $w_H(z) = d$. We find some quantities with $(-1)^{\bigoplus_{i\in I}x_i}$ where I is the set of indices where z is nonzero. This quantity is thus equal to $(-1)^{x\cdot z}$. There are $d!$ of them, that are each of unitary weight. Now, $\sum_x(-1)^{x\cdot z\oplus x\cdot z} = \sum_x 1 = 2^n$. Therefore, if d is equal to $w_H(z)$, $\widehat{\tilde{w_H}^d}(z) = \left(-\frac{1}{2}\right)^d d!2^n \neq 0$, as announced in property P2.

In general, the terms $\sum_x(-1)^{\oplus_{i\in I}x_i}(-1)^{x\cdot z}$ are nonzero if and only if I is the support of z. In this case, the terms are strictly positive (equal to 2^n). In the development of $\left(\sum_{i=0}^{n-1}(-1)^{z_i}\right)^d$, the cardinality of the I have all the same parity:

- d,
- $d-2$, when two terms cancel,
- $d-4$, when four terms or two pairs of terms cancel,
- *etc.*

Thus, when d and $w_H(z)$ have different parities, $\widehat{\tilde{w_H}^d}(z)$ is null (property P3).

Eventually, in the case $\widehat{\tilde{w_H}^d}(z)$ is nonzero ($w_H(z) \leq d$ and of identical parity), we have seen that the development involves a sum of positive values multiplied by $\left(-\frac{1}{2}\right)^d$. Thus $\widehat{\tilde{w_H}^d}(z)$ has the same sign as $(-1)^d$ (property P4). □

A.2 Properties of $\widehat{w_H^d}(z)$

Similar results as those from Lemma 10 can be obtained for $\widehat{w_H^d}(z)$ instead of $\widehat{\tilde{w_H}^d}(z)$ (see some numerical values in Tab. 5):

Lemma 11. *(Invariance in bits reordering).* $\widehat{w_H^d}(a)$ *depends only on* $w_H(a)$.

Lemma 12. $\widehat{w_H^d}(z)$ *has the following properties:*

P5: $\widehat{w_H^d}(z) = 0$ *if and only if* $w_H(z) > d$*;*

P6: $\widehat{w_H^d}(z) = (-1)^d 2^{n-d} d! \neq 0$ *when* $w_H(z) = d$*.*

Proof. (of Lemma 12) We have:

$$\widehat{w_H^d}(z) = \widehat{(\tilde{w_H} + n/2)^d}(z) = \sum_{i=0}^{d} \binom{d}{i} \cdot \widehat{\tilde{w_H}^i}(z) \left(\frac{n}{2}\right)^{d-i}, \tag{7}$$

by linearity of the Fourier transform. Let z and d satisfy $w_H(z) > d$. Thus, $\forall i \leq d, w_H(z) > i$. So, according to Lemma 10, $\widehat{\tilde{w_H}^i}(z) = 0$, hence all terms are null in Eqn. (7) (property 5, backward implication). If $w_H(z) = d$, then the only nonzero term is $\binom{d}{d} \cdot \widehat{\tilde{w_H}^d}(z) \left(\frac{n}{2}\right)^0 = \widehat{\tilde{w_H}^d}(z)$ (property P6). If now $w_H(z) \leq d$, then some terms in Eqn. (7) are nonzero, but they are all of the same sign:

- $\binom{d}{i}$ and $\left(\frac{n}{2}\right)^{d-i}$ are positive, and
- $\widehat{\tilde{w_H}^i}(z)$ is either zero if i and $w_H(z)$ have different parities (property P3 of Lemma 10) or of sign $(-1)^{w_H(z)}$ (property P4 of Lemma 10).

So the sum in Eqn. (7) is nonzero, which proves the direct implication of property P5. □

B Comparison of SMT Results with Other Known Methods

B.1 Comparison of SMT Results with Indicators of Linear Codes

Remark 9. *We checked that the results from Tab. 3 obtained by the SMT can all be written as an indicator of a linear code, but when supp*(f) *is not a power of two (highlighted in gray).*

As accounted for in Lemma 7, those linear codes have parameters

$$[n, n - \log_2(\mathsf{Card}[\mathrm{supp}(f)]), d+1] \ ,$$

i.e. have length n, size $2^n/\mathsf{Card}[\mathrm{supp}(f)]$ and minimal distance $d+1$.

Linear codes of small length (< 36) have been completely characterized (*e.g.* in Magma [28]). For a given dimension n and distance $d+1$, the best size $k_{\max}$ is tabulated (*e.g.* by Markus Grassl [13]). In the list that follows, we characterize the d-CI functions of $n = 8$ variables:

Table 4: Values of $\widehat{\tilde{w_H}^d}(z)$ for $n = 8$.

$w_H(z)$	$d = 0$	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 7$	$d = 8$	$d = 9$
0	256	0	512	0	2816	0	23552	0	250496	0
1	0	−128	0	−704	0	−5888	0	−62624	0	−774848
2	0	0	128	0	1280	0	14528	0	185600	0
3	0	0	0	−192	0	−2880	0	−40992	0	−600960
4	0	0	0	0	384	0	7680	0	129024	0
5	0	0	0	0	0	−960	0	−23520	0	−443520
6	0	0	0	0	0	0	2880	0	80640	0
7	0	0	0	0	0	0	0	−10080	0	−302400
8	0	0	0	0	0	0	0	0	40320	0

Table 5: Values of $\widehat{w_H^d}(z)$ for $n = 8$.

$w_H(z)$	$d=0$	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$	$d=9$
0	256	1024	4608	22528	117504	646144	3714048	22171648	136761984	868063744
1	0	−128	−102	−6848	−44032	−282368	−1828864	−12018848	−80253952	−544488128
2	0	0	128	36	13568	107520	813248	6026496	44311808	325432320
3	0	0	0	−192	−3072	−33600	−314880	−2728992	−22643712	−183169920
4	0	0	0	0	384	7680	99840	1075200	10450944	95477760
5	0	0	0	0	0	−960	−23040	−346080	−4193280	−44956800
6	0	0	0	0	0	0	2880	80640	1370880	18385920
7	0	0	0	0	0	0	0	−10080	−322560	−6108480
8	0	0	0	0	0	0	0	0	40320	1451520

- For $d = 1$, the lowest-weight 1-CI function is the indicator of $[8, 7, 2]$, the dual of the $[8, 1, 8]$ repetition code (also refer to Example 3).
- For $d = 3$, the lowest-weight 3-CI function is the indicator of $[8, 4, 4]$, a quasi-cyclic of degree 2 linear code.
- For $d = 4$, the lowest-weight 4-CI function is the indicator of $[8, 2, 5]$, the Cordaro-Wagner code of length 8.
- For $d = 5$, the lowest-weight 5-CI function is identical to the lowest-weight 7-CI.
- For $d = 6$, the lowest-weight 6-CI function is identical to the lowest-weight 7-CI.
- For $d = 7$, the lowest-weight 7-CI function is the indicator of the $[8, 1, 8]$ repetition code (also refer to Example 3).
- For $d = 8$, the lowest-weight 8-CI function is the constant 1 (see Example 1, from which we exclude the constant 0).

The case excluded from this list is that of the indicator of a code of length $n = 8$, size 12, and dual distance $d + 1 = 3$. The best binary linear code matching these characteristics is $[8, 4, 3]$, whose dual has size $2^4 = 16 > 12$. The best $\mathbb{Z}_4$-linear code is also of a size power of two, hence non-optimal.

A possible solution has already been discussed in Appendix D.2. of [21]. It has algebraic degree 6.

B.2 Comparison of SMT Results with Maiorana-McFarland Construction

We can wonder if the 2-CI function of weight 12 over $\mathbb{F}_2^8$ could have been found by the Maiorana-McFarland construction, presented in Sec. 3.3.3. Now, the weight of f, equal to $\widehat{f}(0)$, satisfies $W_f(0) = 2^n - 2\widehat{f}(0) = 2^r \mathsf{Card}[\phi^{-1}(0)]$, by considering Eqn. (4) in $(a, b) = (0, 0)$. Thus $\mathsf{Card}[\phi^{-1}(0)] = (2^n - 2\mathsf{Card}[\mathrm{supp}(f)])/2^r$. This cardinality is incompatible with the general construction ($g = 0$ and $\phi^{-1}(0)$ an affine space), since an affine space has cardinality power of two, whereas $(2^n - 2\mathsf{Card}[\mathrm{supp}(f)])/2^r$ can only take the following values: 116, 58 or 29.

Thus, specific constructions with $g \neq 0$ and unstructured $\phi^{-1}(0)$ (*i.e.* $\phi^{-1}(0)$ is an arbitrary subset of $\mathbb{F}_2^s$) must be considered. Let us start with the case $a \neq 0$; the construction demands that the reciprocal image of a be empty for $1 \leq w_H(a) \leq d$. Therefore r must be strictly greater than d. But in addition, Eqn. (4) applied to $(a, b) = (0, 0)$ shows that $W_f(0, 0)$ must be a multiple of 2^r. As $W_f(0) = 256 - 2 \times 12 = 232$, the possible values for r are 1, 2 and 3, since $2^3 \mid 232$ but $2^4 \nmid 232$. This means that r must be chosen equal to 3. Now, always in $(a, b) = (0, 0)$, Eqn. (4) writes:

$$\sum_{v \in \phi^{-1}(0)} (-1)^{g(v)} = \mathsf{Card}[\phi^{-1}(0)] - 2\mathsf{Card}[\mathrm{supp}(g|_{\phi^{-1}(0)})] = 29 \ , \qquad (8)$$

where $g|_{\phi^{-1}(0)}$ represents the restriction of g to the set $\phi^{-1}(0)$. As $\phi : \mathbb{F}_2^s \to \mathbb{F}_2^r$, $0 \leq \mathsf{Card}[\phi^{-1}(0)] \leq 2^s = 32$. Thus, there are two possibilities for the cardinals in Eqn. (8), namely either

1. $\mathsf{Card}[\phi^{-1}(0)] = 29$ and $\mathsf{Card}[\text{supp}(\,g|_{\phi^{-1}(0)})] = 0$, or
2. $\mathsf{Card}[\phi^{-1}(0)] = 31$ and $\mathsf{Card}[\text{supp}(\,g|_{\phi^{-1}(0)})] = 1$.

However, as we will show, those two cases are incompatible with the conditions when $a = 0$. Indeed, in this case, it must be checked that for all b, $1 \leq w_H(b) \leq d$, $\sum_{v \in \phi^{-1}(0)} (-1)^{g(v) \oplus b \cdot v} = 0$. But when $\mathsf{Card}[\phi^{-1}(0)]$ is odd, so is $\sum_{v \in \phi^{-1}(0)} (-1)^{g(v) \oplus b \cdot v}$. Therefore, as $\mathsf{Card}[\phi^{-1}(0)] \in \{29, 31\}$, this quantity cannot be equal to zero.

So, in conclusion, the Maiorana-McFarland construction cannot disclose the 2-CI function of weight 12 over $\mathbb{F}_2^8$ found by the SMT.

Let us note $f_{n,d,w}$ d-CI functions of n variables and of weight w. The same reasoning can be applied to show that the cases $f_{9,2,12}$, $f_{10,2,12}$ and $f_{11,2,12}$ cannot be found by the Maiorana-McFarland construction. Indeed, we have $r > d$ because the nonzero images of ϕ must be of Hamming weight strictly greater than d. At the same time, Eqn. (4) demands that $W_f(0)$ be a multiple of 2^r. Concretely, we have those factorizations:

- For $f_{9,2,12}$: $W_{f_{9,2,12}}(0) = 2^9 - 2 \times 12 = 488 = 2^3 \times 61$;
- For $f_{10,2,12}$: $W_{f_{10,2,12}}(0) = 2^{10} - 2 \times 12 = 1000 = 2^3 \times 125$;
- For $f_{11,2,12}$: $W_{f_{11,2,12}}(0) = 2^{11} - 2 \times 12 = 2024 = 2^3 \times 253$.

In all cases, 2^3 is a multiple of $W_f(0)$ but not 2^4, thus $r = 3$. As $\sum_{v \in \phi^{-1}(0)} (-1)^{g(v)}$ is odd in all cases (it is equal to 61, 125 or 253), then the condition of Eqn. (4) when $a = 0$ cannot be fulfilled.

The case of functions $f_{9,3,24}$, $f_{10,3,24}$, $f_{11,3,24}$ and $f_{12,3,24}$ is similar, except that the only possible value for r is 4.

C Orthogonal Arrays *vs* d-CI Functions

In this appendix, we explain the difference between the minimal number of lines of orthogonal arrays and the weight of d-CI functions. We first give a graphical illustration of an orthogonal array $OA(w, n, s = 2, d)$ in Fig. 3. As explained in Theorem 2, the lines of an orthogonal array form a (n, w) code with dual distance greater than or equal to $d + 1$.

C.1 Classical Inequalities

If we note $w_{n,d}$ the minimal number of lines in an orthogonal array with n columns and of strength d, then the relationships depicted in Fig. 4 are verified. They are given in [10, §12.2, p. 318], and recalled here:

1. $w_{n,d} \geq w_{n,d-1}$;
2. $w_{n,d} \geq \frac{1}{2} \times w_{n+1,d}$;
3. $w_{n,d} \geq w_{n-1,d}$;
4. $w_{n,d} \geq 2 \times w_{n-1,d-1}$.

Indeed, from an $OA(w_{n,d}, n, s = 2, d)$, one can derive:

1. an $OA(w_{n,d}, n, s = 2, d - 1)$, since a strength d implies any strength d' such that $0 \leq d' \leq d$;
2. an $OA(2 \times w_{n,d}, n + 1, s = 2, d)$, by the construction $A \mapsto \begin{array}{c|c} 0 & A \\ \hline 1 & A \end{array}$;
3. an $OA(w_{n,d}, n - 1, s = 2, d)$, by dropping one factor;
4. an $OA(\frac{1}{2} \times w_{n,d}, n - 1, s = 2, d - 1)$, extracting A_0 from $\begin{array}{c|c} 0 & A_0 \\ \hline 1 & A_1 \end{array}$.

The first, second, and fourth properties also apply to d-CI functions. They have been demonstrated in Sec. 3.2 respectively as Lemma 3, 4 and 5. However, the third relationship does not apply to d-CI functions, since it is possible that the same codeword appears multiple times. This point is not covered in [10], because orthogonal arrays are considered non-simple (*i.e.* that can have multiple identical lines). To our best knowledge, it is, as of today, still an open problem. We tackle some aspects of it in the next Sec. C.2.

Incidentally, our results from Tab. 3 can complement the Tab. 12.1 of [10] (binary OAs):

- The OA with 9 columns and of strength 4 with no duplicate lines has a minimal number of lines equal to 128;
- The OA with 10 columns and of strength 4 with no duplicate lines has a minimal number of lines equal to 128;
- The OA with 10 columns and of strength 5 with no duplicate lines has a minimal number of lines equal to 256;
- The OA with 10 columns and of strength 6 with no duplicate lines has a minimal number of lines equal to 512.

C.2 On the Comparison Between $w_{n,d}$ and $w_{n-1,d}$

Remark 10. *It appears that all columns in Tab. 3 have the property that the values which are computed are increasing (in a non-strict way) with n. We leave open the question of knowing whether this is true for all values (computed or not). This result is easily proved for orthogonal arrays but not for d-CI*

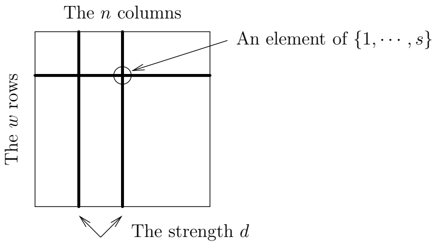

Figure 3: Notations for an orthogonal array $OA(w, n, s = 2, d)$.

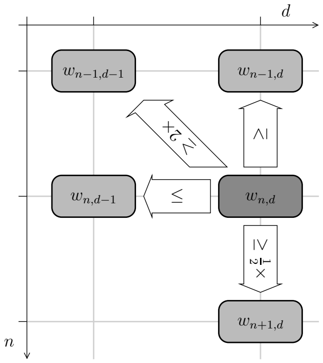

Figure 4: Relationships between the minimal number of columns of orthogonal arrays of strength d with n columns.

Boolean functions (refer to App. C for more details.) We observed that, for all the computed values, the functions f achieving these values have the property that the $(n-1)$-variable subfunctions $f_0(x) = f(x,0)$ and $f_1(x) = f(x,1)$ have disjoint supports. Note that, if such property could be proved, it would imply the increasing property of the entries of Tab. 3 with n; indeed, in every such case, the input at row n and column d in Tab. 3 is larger than or equal to the input at row $n-1$ and same column. Since, when an n-variable nonzero function f is d-CI and is such that f_0 and f_1 have disjoint supports, the nonzero $(n-1)$-variable function $f_0 \oplus f_1$ is also d-CI since, for every nonzero vector a of length $n-1$ and Hamming weight at most d, we have $\widehat{f_0 \oplus f_1}(a) = \widehat{f_0}(a) + \widehat{f_1}(a) = \widehat{f}(a,0) = 0$, and the Hamming weight of $f_0 \oplus f_1$ is equal to that of f (and of course the minimum Hamming weight of nonzero $(n-1)$-variable d-CI functions is smaller than or equal to the Hamming weight of $f_0 \oplus f_1$).

Actually, it would be sufficient to prove that $f_0 f_1$ is d-CI. Indeed, then $\widehat{f_0 f_1}(a) = 0$ for all $0 < w_H(a) \leq d$ and as $\widehat{f_0}(a) + \widehat{f_1}(a) = 0 = \widehat{f_0 \oplus f_1}(a) + 2\widehat{f_0 f_1}(a)$, we have two cases:

1. *$f_0 f_1 = 0$, then $f_0 \oplus f_1$ is nonzero d-CI of weight at most that of f,*
2. *$f_0 f_1 \neq 0$, then $f_0 f_1$ is nonzero d-CI of weight at most that of f (and $f_0 \oplus f_1$ can be zero).*

Proposition 1. *If there exists a d-CI function f of n variables and of minimal weight $w_{n,d}$ whose support is a linear code C, then we have $w_{n,d} \geq w_{n-1,d}$.*

Proof. Let us assume that there exists $f : \mathbb{F}_2^n \to \mathbb{F}$ d-CI, of minimal weight $w_{n,d}$, and whose support is a linear code C. If the code C contains all the n vectors e_i ($i \in [\![1, n]\!]$) of weight 1, then $C = \mathbb{F}_2^n$ because these vectors form a basis $(e_1, \cdots, e_n)$ of $\mathbb{F}_2^n$. This concerns only the case $d = n$. Otherwise, when $d < n$, the number of codewords in C is strictly smaller than 2^n. Hence there exists one coordinate $1 \leq i \leq n$ such that $e_i \notin C$. Let C' be the code C in which the coordinate i has been erased. C' is a set of $(n-1)$-bit words, that are all different. Indeed, C being linear, if two codewords of C differ by one position at coordinate i, then by addition, C contains the codeword e_i, which is a contradiction. As a consequence, C' is a code. The indicator f' of C' has the same weight as f, and is also d-CI. Therefore, the minimal weight of d-CI functions of $(n-1)$ inputs is smaller than that of f', *i.e.* $w_{n-1,d} \leq w_{n,d}$. □

Lemma 13. *In the general case (C is unrestricted), we have:*

- *the* inequality*:* $w_{n,d} \geq 2^n \left(1 - \frac{n}{2(d+1)}\right)$, *and*
- *the* property*: if* $w_{n,d} = 2^n \left(1 - \frac{n}{2(d+1)}\right)$, *then* $w_{n-1,d} \leq w_{n,d}$.

Proof. Let us define the set $\mathcal{E} = \{(x,y) \in C^2 / \exists i \in [\![1,n]\!] / x \oplus y = e_i\}$. We have:

$$\begin{aligned}
\mathsf{Card}[\mathcal{E}] &= \sum_{i=1}^{n} \mathsf{Card}[\{(x,y) \in C^2 / x \oplus y = e_i\}] \\
&= 2^{-n} \sum_{i=1}^{n} \sum_{x,y \in C} \sum_{a \in \mathbb{F}_2^n} (-1)^{(x \oplus y \oplus e_i)\cdot a} \\
&= 2^{-n} \sum_{i=1}^{n} \left(\sum_{a \in \mathbb{F}_2^n / a_i = 0} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 - \sum_{a \in \mathbb{F}_2^n / a_i = 1} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 \right) \\
&= 2^{-n} \sum_{i=1}^{n} \left(\sum_{a \in \mathbb{F}_2^n} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 - 2 \sum_{a \in \mathbb{F}_2^n / a_i = 1} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 \right) \\
&= 2^{-n} \left(n \sum_{a \in \mathbb{F}_2^n} \sum_{x,y \in C} (-1)^{(x \oplus y)\cdot a} - 2 \sum_{i=1}^{n} \sum_{a \in \mathbb{F}_2^n / a_i = 1} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 \right) \\
&= n\mathsf{Card}[C] - 2^{1-n} \sum_{a \in \mathbb{F}_2^n} w_H(a) \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 \quad .
\end{aligned}$$

As the indicator of C is d-CI, we have $\sum_{x \in C} (-1)^{a\cdot x} = 0$ if $0 < w_H(a) \leq d$. Thus:

$$\begin{aligned}
\mathsf{Card}[\mathcal{E}] &\leq n\mathsf{Card}[C] - 2^{1-n}(d+1) \sum_{a \in \mathbb{F}_2^n} \left(\sum_{x \in C} (-1)^{a\cdot x} \right)^2 + 2^{1-n}(d+1)\mathsf{Card}[C]^2 \\
&= n\mathsf{Card}[C] - 2^{1-n}(d+1)2^n\mathsf{Card}[C] + 2^{1-n}(d+1)\mathsf{Card}[C]^2 \\
&= \mathsf{Card}[C] \left(n - 2(d+1) + 2^{1-n}(d+1)\mathsf{Card}[C] \right) \quad ,
\end{aligned}$$

since $\sum_{a \in \mathbb{F}_2^n} (-1)^{u\cdot a} = 0$ when $u \neq 0$. Consequently:

- as the cardinal of $\mathcal{E}$ is non-negative, $n - 2(d+1) + 2^{1-n}(d+1)\mathsf{Card}[C] \geq 0$, *i.e.* $\mathsf{Card}[C] \geq 2^n \left(1 - \frac{n}{2(d+1)}\right)$, which is trivial if $d+1 \leq n/2$ but definitely of different nature than the well-known inequality $\mathsf{Card}[C] \geq 2^d$ (Lemma 2); We notice that the same result has been found by other means in [2].

- if $\mathsf{Card}[C]$ is minimal ($\mathsf{Card}[C] = 2^n \left(1 - \frac{n}{2(d+1)}\right)$), then $\mathsf{Card}[\mathcal{E}] = 0$, and thanks to an argument similar to that of Proposition 1, we can derive from C a code of length $(n-1)$ by puncturing any coordinate (the minimal distance of C is at least 2), and thus $w_{n-1,d} \leq w_{n,d}$.

□

Remark 11. *All the lower bounds on $w_{n,d}$ presented in Tab. 1 for $0 < d \leq n \leq 13$ are greater than or equal to the minoration given in Lemma 13,* i.e

$w_{n,d} \geq 2^n \left(1 - \frac{n}{2(d+1)}\right)$. *It is an open problem to know whether the minoration of Lemma 13 is always smaller than or equal to than the values obtained by Delsarte LP bound.*

Remark 12. *The Preparata code, that is the formal dual of the Kerdock code, has a length* 2^n *(n even,* $n \geq 4$*) and dual distance of* $2^{n-1} - 2^{n/2-1}$*. As the Kerdock code has an optimal cardinal (large) for this minimal distance and this length, the Preparata code probably has a minimal cardinal.*