IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2021, No. 4, pp. 57–87. DOI:10.46586/tches.v2021.i4.57-87

Structural Attack (and Repair) of Diffused-Input-Blocked-Output White-Box Cryptography

Claude Carlet[1*], Sylvain Guilley[2] and Sihem Mesnager[3]

[1] LAGA, Department of Mathematics, University of Paris VIII, Paris, France
University of Bergen, Norway
claude.carlet@gmail.com

[2] Secure-IC S.A.S. (7th floor), 104 Boulevard du Montparnasse, 75014 Paris, France
LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France
sylvain.guilley@secure-ic.com

[3] Department of Mathematics, University of Paris VIII, F-93526 Saint-Denis, University Sorbonne Paris Cité, LAGA, UMR 7539, CNRS, 93430 Villetaneuse
LTCI, Télécom Paris, Polytechnic Institute of Paris, 91120 Palaiseau, France
smesnager@univ-paris8.fr

Abstract. In some practical enciphering frameworks, operational constraints may require that a secret key be embedded into the cryptographic algorithm. Such implementations are referred to as White-Box Cryptography (WBC). One technique consists of the algorithm's tabulation specialized for its key, followed by obfuscating the resulting tables. The obfuscation consists of the application of invertible diffusion and confusion layers at the interface between tables so that the analysis of input/output does not provide exploitable information about the concealed key material.

Several such protections have been proposed in the past and already cryptanalyzed thanks to a complete WBC scheme analysis. In this article, we study a particular pattern for local protection (which can be leveraged for robust WBC); we formalize it as DIBO (for Diffused-Input-Blocked-Output). This notion has been explored (albeit without having been nicknamed DIBO) in previous works. However, we notice that guidelines to adequately select the invertible diffusion ϕ and the blocked bijections B were missing. Therefore, all choices for ϕ and B were assumed as suitable. Actually, we show that most configurations can be attacked, and we even give mathematical proof for the attack. The cryptanalysis tool is the number of zeros in a Walsh-Hadamard spectrum. This "spectral distinguisher" improves on top of the previously known one (Sasdrich, Moradi, Güneysu, at FSE 2016). However, we show that such an attack does not work always (even if it works most of the time).

Therefore, on the defense side, we give a straightforward rationale for the WBC implementations to be secure against such spectral attacks: the random diffusion part ϕ shall be selected such that the rank of each restriction to bytes is full. In AES's case, this seldom happens if ϕ is selected at random as a linear bijection of $\mathbb{F}_2^{32}$. Thus, specific care shall be taken. Notice that the entropy of the resulting ϕ (suitable for WBC against spectral attacks) is still sufficient to design acceptable WBC schemes.

Keywords: White-Box Cryptography · obfuscation · Diffused-Input-Blocked-Output (DIBO) · spectral characteristics · number of zeros in Walsh spectrum · mathematical proof of attack · repaired DIBO.

*The research of the first author is partly supported by the Trond Mohn Foundation.

Received: 2021-04-15 Accepted: 2021-06-15 Published: 2021-08-11

1 Introduction

1.1 Historical Background on White-Box Cryptography

White-Box Cryptography is an implementation strategy for cryptographic algorithms that need to conceal a secret key even though their design is public. The requirement is very strong, as WBC shall resist even if the source code of the algorithm entangled with the key is completely disclosed. The first WBC implementations have been pioneered by Chow, Eisen, Johnson, and van Oorschot. They were examples of implementations for block ciphers DES [CEJvO02a] and AES [CEJvO02b]. Initially, the principle of WBC was to protect software implementations, which are very amenable to code disclosure (attack termed *code lifting*). However, WBC has also been repurposed to protect embedded firmware or even hardware implementations [CFD+10, SMG16]. Still, the idea is that the attacker will either manage to read intermediate values within the implementation or correlate on them directly or through some side-channel analysis.

Being aware of both those attacks, a WBC methodology often consists of two steps:

1. First, the representation of the block cipher is specialized for a given key as a succession of table lookups;

2. Second, each table is composed with input and/or output random bijections.

Those bijections are statically drawn (by the so-called white-boxing software), and aim at decorrelating the table contents before and after the composition. The reason for the randomness to be static is that the WBC instance can be produced in secure facilities. While it is deployed, refreshing the randomness is hopeless because it is assumed that the attacker is capable of hooking the random number source (i.e., to disable it).

The threat model assumes that the attacker knows precisely the abovementioned tables, e.g., he can decompile them or identify them through some side-channel information. Therefore, in the sequel, we take for granted that the attacker knows the WBC design, including its rationale precisely: he knows the tables and how they are built (but not the static randomness they embed nor the secret key they conceal).

Such a blending of lookup tables is based on usual primitives in block cipher design, namely linear operations for the diffusion and non-linear operations for the confusion. There are two kinds of bijections.

1. Internal encodings: they are randomizations within the algorithm, and therefore cancel pairwise (i.e., the second operation is the inverse of the first one, such that their composition is the identity) in the dataflow of table lookups, since the total application must not have its functionality altered.

2. External encodings: they modify the algorithm by applying a first transformation on the plaintext and a final one of the ciphertext, hence change the algorithm.

Clearly, in both cases, encodings must be invertible.

1.2 State-of-the-art approaches

The field of WBC is characterized by cryptanalyses appearing fast after white-box schemes have been proposed. For example, the early work of Chow et al. [CEJvO02a, CEJvO02b] has soon been shown vulnerable to differential cryptanalysis [GMQ07, WMGP07] as well as algebraic cryptanalytic attacks [BGEC04, MGH08, LRM+13]. This led to some new proposals for white-box implementations of AES. In 2009, Xiao et al. in [XL09] proposed a variant of the design of Chow et al. using larger linear encodings, for which again algebraic cryptanalytic attacks were identified in [MRP12]. Other approaches suggest building white-box AES implementations using perturbations [BCD06] (which were broken in [MWP10]).

Lately, some protections leveraging masking and shuffling techniques (repurposed from side-channel analysis) have been proposed by Lee et al. [LKK18]. But these countermeasures failed against new attack methods, equally inspired from the field of side-channel analysis, such as that of Rivain et al. [RW19]. Consequently, new WBC schemes inspired from high-order masking protection, such as [SEL21], aim to resist all the same. Today, it is too early to know whether this scheme provides enough security.

It is possible to classify the attacks on WBC into three categories:

1. *statistical attacks* (similar to cryptanalysis techniques), such as [GMQ07, WMGP07];

2. those which leverage techniques from *grey-box analysis* [BBB+19] (i.e., side-channel or fault injection analyses), such as differential fault analysis (DFA [TH16]), differential computation analysis (DCA [BHMT16, BBMT18]), collision or mutual information [RW19], or high-order computational attacks [BRVW19, MA20, GRW20];

3. those which rely on *Fourier transforms*, such as [SMG16, LK20, LJK20].

To be exhaustive in our presentation of the state-of-the-art, let us also mention survey papers such as [BT20, GPRW20], which typically report on attack methods observed at public contests, such as `WhibOx` [whi16, che17]. Eventually, a technical report is currently under drafting within the International Standardization Organization [ISO21].

In this paper, we develop the third category of attacks, in that we show that they are very well suited for attacking a WBC scheme based on tables dissimulation. We briefly introduce the theoretical notions required in this respect.

1.3 Spectral analysis mathematical notions

Let n and m be two positive integers. Given an n-variable Boolean function f, the support of f is the set $supp(f) = \{x \in \mathbb{F}_2^n;\ f(x) = 1\}$ and the Walsh transform of f maps every element $u \in \mathbb{F}_2^n$ to:

$$W_f(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+u\cdot x},$$

where an inner product in $\mathbb{F}_2^n$ has been chosen and is denoted by "$\cdot$". The Walsh transform can be seen as the correlation between $(-1)^f$ and a basis of functions $(x \mapsto (-1)^{u\cdot x})_{u\in\mathbb{F}_2^n}$. In this respect it is a powerful tool to analyse the properties of the target function f. The Walsh transform is closely related to the Fourier-Hadamard transform $\widehat{f}(u) = \sum_{x\in supp(f)}(-1)^{u\cdot x}$ by the relations $W_f(0) = 2^n - 2\widehat{f}(0)$ and $W_f(u) = -2\widehat{f}(u)$ for $u \neq 0$. A butterfly algorithm allows for the computation of W_f with complexity $n2^n$ additions [Car21, §2.3.1, at page 54].

Given any vectorial (n, m)-function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, the Walsh transform of F maps every pair $(u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ to the value at u of the Walsh transform of the Boolean function $v \cdot F$, that is:

$$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v\cdot F(x)+u\cdot x},$$

where two inner products in $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ have been chosen and are both denoted by "$\cdot$". Any function $v \cdot F$, $v \neq 0$, is called a component function of F. The multi-set of those values $W_F(u, v)$ where $u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^m, v \neq 0$, is called the *Walsh spectrum* of F. The *extended Walsh spectrum* of a function is the multi-set of the values taken by the absolute values of its Walsh transform. Affine equivalence (that is, composition on the left and on the right by affine automorphisms) preserves the extended Walsh spectrum. The computational complexity of the evaluation of the Walsh spectrum of (n, m)-functions is $2^m n 2^n$.

Incidentally, any vectorial Boolean function can be uniquely represented under the Algebraic Normal Form (ANF, [Car21, (2.6) at page 39]), as:

$$F(x) = \sum_{I \subseteq \{1,\ldots,n\}} \left(\prod_{i \in I} x_i \right) a_I = \sum_{I \subseteq \{1,\ldots,n\}} x^I a_I,$$

where a_I belongs to $\mathbb{F}_2^m$. The algebraic degree of F is defined as:

Definition 1 (Algebraic degree of F, [Car21, §2.2.1, at page 40])**.**

$$d^\circ_{\text{alg}} F = \max \big\{ |I| \mid I \subseteq \{1, \ldots, n\}, a_I \neq 0 \big\}.$$

Intuitively, the algebraic degree relates to the randomness of the (n, m)-function. We have that $d^\circ_{\text{alg}} F = 1$ for non-constant affine functions (i.e., sums of constants and nonzero linear functions), whereas $d^\circ_{\text{alg}} F$ is close to its maximal value n in general.

The following notion will not play a role in the present paper but it played a role in [SMG16], leading the authors to the choice of a particular distinguisher for their attack; this is why we recall its definition:

Definition 2 (Correlation-immunity [Car21, Def. 37, in §3.3.1 at page 129])**.** Let F be an (n, m)-function and t such that $0 \leq t \leq n$. The vectorial Boolean function $F(x)$ is termed correlation-immune of order t if its output distribution does not change when at most t coordinates x_i of x are kept constant.

Let us denote by w_H the Hamming weight function. Correlation-immune functions can be characterized in terms of Walsh spectrum:

Lemma 1 ([Car21, Proposition 41, in §3.3.1 at page 130])**.** *Let F be an (n, m)-function. Then F is correlation-immune at order t if and only if $W_F(u, v) = 0$ for every $u \in (\mathbb{F}_2^n)^*$ such that $w_H(u) \leq t$ and for every $v \in (\mathbb{F}_2^m)^*$, that is for every $v \in \mathbb{F}_2^m$.*

We shall also need to use the univariate representation of (n, n)-functions. Through an identification between the vector space $\mathbb{F}_2^n$ and the finite field $\mathbb{F}_{2^n}$ (the latter being an n-dimensional vector space over $\mathbb{F}_2$, it can indeed be identified with the former), there is a unique representation of any (n, n)-function F in the form

$$F(x) = \sum_{i=0}^{2^n - 1} a_i x^i \quad \in \quad \mathbb{F}_{2^n}[x]/(x^{2^n} + x)$$

with $a_i \in \mathbb{F}_{2^n}$. The algebraic degree of F equals then the maximum 2-weight $w_2(i)$ (i.e. binary expansion's Hamming weight) of the exponent i such that $a_i \neq 0$.

1.4 Contributions

In this paper, we study the security of WBC leveraging internal encoding, in particular through the random obfuscation of tabulated algorithm parts, like the T-box, based on DIBO concept. In particular, we expand the knowledge related to "spectral attacks". Namely, Our contributions are as follows:

- We prove that in most of the cases, there is indeed a bias in the analyzed tables related to the Walsh transform, which had been only observed previously, without that the reason be understood. This bias, which provides then what we call a spectral distinguisher, is related to a property of DIBO functions: the number of zeros in the spectrum (that is, the number of zero values taken by the Walsh transform) of their component functions is large. We show how this justifies the distinguisher that has been used in the state-of-the-art, which is the arithmetic mean of the absolute value of the Walsh transform of coordinate functions;

- We show that this state-of-the-art distinguisher is not the best possible, being only indirectly related to the property of DIBO functions. Our distinguisher by the number of zero values taken by the Walsh transform is stronger;
- We show that for most DIBO random obfuscations, our attack (leveraging our improved distinguisher) succeeds;
- We mathematically prove our attack success under a simple condition on the linear diffusion layer of the obfuscation scheme; our proof also justifies *a posteriori* why the state-of-the-art distinguisher works as well in similar conditions;
- We exhibit a subset of DIBO obfuscating functions for which the attack fails, hence the WBC is secure against spectral attacks.

1.5 Outline

The rest of the paper is organized as follows. The precise definition of the targeted algorithms to be protected and the DIBO white-boxing protection is provided in Sec. 2. The concept of the spectral distinguishers is presented in Sec. 3. This section also provides a comparison between one prominent state-of-the-art distinguisher (Sasdrich, Moradi, Güneysu, [SMG16]) and ours. The demonstration that the attack works unconditionally when weak random linear permutations (termed ϕ) are selected is provided in Sec. 4. Finally, we explicit in Sec. 5 the conditions upon which a DIBO can be secure w.r.t. our attack, which constitutes guidelines for robust WBC (it implies a reduction of choices for ϕ). Conclusions and perspectives are in Sec. 6. In appendix A, some examples of attacks are illustrated.

2 Studied WBC rationale: DIBO

2.1 Use-case on AES

In this section, we are concerned with a simple protection pattern, namely the white-boxing of the AES T-box.

The Advanced Encryption Standard (AES) block cipher relies on several rounds of encryption with a different round key at each round. The security of the AES relies on the growing complexity of the encryption with the number of rounds of encryption assuming that an attacker has only access to the input of the first round and the output of the final round. Nevertheless, as we explained, assuming in white-box cryptography that an attacker can have access to each encryption round input and output makes necessary to obfuscate the encryption rounds (even internally of their structure) to complexify the level of the required attack.

Each round of the AES can be implemented by calling public standard T boxes (4 in total for encryption, which means there are another 4 for decryption), which evaluate 8-bit to 32-bit functions. Notice that AES datapath (128 bit) consists of four T box calls per round column, hence sixteen per round. The encryption starts by an `AddRoundKey` step (sometimes abridged ARK), a mere bitwise `XOR`. Namely, each 8-bit message x is added a private key $k^\star$ before applying the composition of the Substitution box S (also known as `SubBytes`) and the diffusion (by `MixColumns`).

The operation we therefore consider for being white-boxed is thus this algorithmic step:

$$x \mapsto T(x + k^\star). \tag{1}$$

Notice that x and $k^\star$ are seen as elements of the finite field $\mathbb{F}_{2^8} = \mathbb{F}_{256}$, hence addition is the `XOR` operation. The subtraction is the same operation as the addition, and we use "+"

for both operations. For instance, we consider the first T-box, which is:

$$T(x) = \begin{pmatrix} \texttt{02} \\ \texttt{01} \\ \texttt{01} \\ \texttt{03} \end{pmatrix} S(x) = \begin{pmatrix} \texttt{02}S(x) \\ S(x) \\ S(x) \\ \texttt{03}S(x) \end{pmatrix}$$

where 01 (resp. 02, 03) is the element 1 (resp. α, $\alpha+1$) in $\mathbb{F}_{256}$ seen as $\mathbb{F}_2[\alpha]/\langle\alpha^8+\alpha^4+\alpha^3+\alpha+1\rangle$. In the sequel, the elements of $\mathbb{F}_{2^n}$, depending on the context, are considered as elements of a finite field or vectors of n bits. We will refer to either of the cases using the notation $\mathbb{F}_{2^n}$ vs $\mathbb{F}_2^n$.

2.2 WBC lookup tables protection with DIBO

It is easy to extract $k^\star$ from the table of 256 words of 32-bit defined as $\{T(x+k^\star), x \in \mathbb{F}_{256}\}$ (as per (1)). Indeed, there are only 256 tables, each one corresponding to one key.

In the DIBO white-boxing concept, the secret $k^\star$ is concealed by applying an *internal encoding*, which consists in the application of a random secret permutation ϕ to the output of the T-box and then of a second secret function B designed by blocks of small random permutations applied in parallel to the 32-bit words.
The obfuscated function is therefore defined as $O_{k^\star}$:

$$x \mapsto O_{k^\star}(x) = B \circ \phi \circ T(x + k^\star). \tag{2}$$

An example of WBC-obfuscated T-box is provided in equation (8) of Appendix 3.5. In this scenario, we recall that we assume that there is no *external encoding*. We call the chained function $B \circ \phi$ "DIBO", referring to "Diffused-Input-Blocked-Output".

Definition 3. Given two positive integers n and n_0, such that n is a multiple of n_0, we call *Diffused-Input-Blocked-Output* functions those $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ such that, up to a permutation of the output coordinates, $F = B \circ \phi$, where ϕ is a linear permutation[1] of $\mathbb{F}_2^n$ and B is such that there exist bijective (n_0, n_0)-functions $B_1, \ldots, B_{n/n_0}$ such that:

$$\begin{aligned} &B(x_1, \ldots, x_n) = \\ &(B_1(x_1, \ldots, x_{n_0}), B_2(x_{n_0+1}, \ldots, x_{2n_0}), \ldots, B_{\frac{n}{n_0}}(x_{n-n_0+1}, \ldots, x_n)), \end{aligned}$$

that is, $B(x_1, \ldots, x_n)$ equals the concatenation of the vectors

$$B_1(x_1, \ldots, x_{n_0}), B_2(x_{n_0+1}, \ldots, x_{2n_0}), \ldots, B_{\frac{n}{n_0}}(x_{n-n_0+1}, \ldots, x_n).$$

Remark 2.1. *We could wish to also write "up to a permutation of the input coordinates of B", but such permutation can be without loss of generality taken equal to the identity since applying such permutation is equivalent to applying a permutation to the output coordinates of ϕ and permuting the output coordinates of a linear permutation changes it in another linear permutation.*

The structure of the DIBO obfuscation scheme is motivated hereafter:

- The linear function ϕ shall have the full `MixColumns` bitwidth to protect the complete datapath of one AES column; therefore, we consider $n = 32$.

[1] Beware that a "permutation" has different meanings in different contexts. A *permutation of an ordered list* is a rearrangement of the elements into a one-to-one correspondence with the original list. In the context of functions, a *permutation* is synonymous for a bijective function (linear or non-linear); a *linear permutation* is a linear bijection.

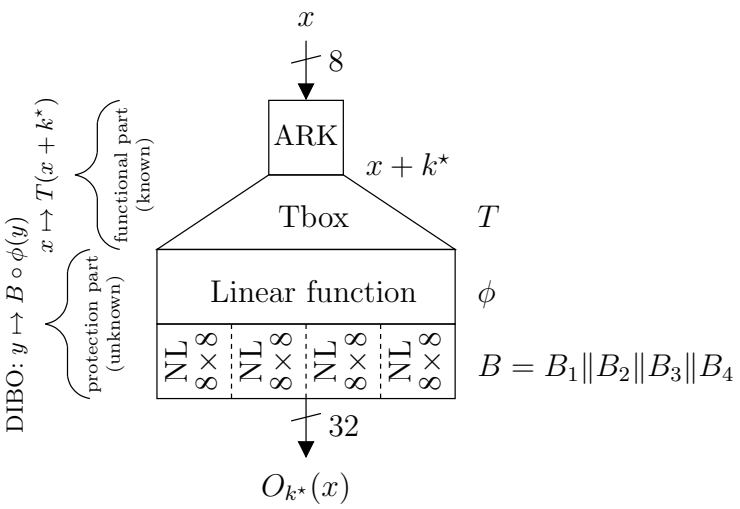

Figure 1: White-box protection $O_{k^\star}$ (equation (2)) of $x \in \mathbb{F}_2^8 \mapsto T(x + k^\star) \in \mathbb{F}_2^{32}$ (where T is known but $k^\star$ is one byte of the secret key), with DIBO function $B \circ \phi$ (i.e., the internal encoding). Notice that "NL" stands for the non-linear B_i, for $1 \leq i \leq 4$

- The blocked non-linear operations are so because the output of the T-box (i.e., `MixColumns`) is followed downstream by a `XOR` operation. Hence, the `XOR` concatenated two-inputs need to be a table of $2 \times n_0$ (where n_0 is the B_i output bitwidth); for this reason, s shall be of moderate size ($2, 3, 4, ..., 8$ max — as $2^{2\times 8}$-input tables are the maximum which is tolerable from an implementation standpoint[2]). Thus, we consider $n_0 = 8$.

Similar modelization is considered in state-of-the-art papers, such as [SMG16] (see TMC in Fig. 1 or Fig. 4) and [LK20] (see bottom part of type-II in Fig. 1(a) and type-IIM in Fig. 2).

The white-boxed version of $x \mapsto T(x + k^\star)$ is still a $\mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ function, depicted in Fig. 1. The function $O_{k^\star}$ protects $k^\star$ using secret bijections ϕ and B. Indeed, the full truth table of $O_{k^\star}$ is public, but it is assumed that the secret functions ϕ and B are sufficiently entropic to hide $k^\star$. (We shall prove in this paper that this assumption does not hold unless provisions are taken concerning properties of ϕ.)

Interestingly, the DIBO internal encoding (namely, $B \circ \phi$) has a mini-cipher structure with a linear function for diffusion and a non-linear function for confusion.

2.3 Intuitive rationale regarding the security of DIBO

Leveraging DIBO as an internal encoding is not new: it has been employed in [CEJvO02b, SMG16, LK20]. Most probably, submissions to competitions (`WhibOx`, Capture-The-Flag contests, etc.) also resort to DIBO without saying so (the white-boxing application being generally not disclosed). Sadly, most of the aforementioned competitions have resulted in successful WBC scheme attacks.

Now, the design of the white-box protection displayed in Fig. 1 seems strong, owing to the great deal of entropy a designer injects in the random DIBO internal encoding. Thus, it is tempting to believe that DIBO internal encoding provably hides the key $k^\star$.

[2]Obfuscated binary `XOR` function, which takes as input not one WBC-ed input, but two of them. It is the equivalent of the type-IV encoding in [CEJvO02b, Fig. 1, page 257], where *nibbles* are traded for *bytes* for increased security [RW19, §4].

One way to get an intuition in favor of such a hasty claim would consist in resorting to one representative example, showing that the DIBO construction implements the *WBC ambiguity* concept introduced in [CEJvO02b, §4.2].

Nota bene. *Beware that this Subsection 2.3 is just an argumentation to insist that a WBC shall be scrutinized carefully. The actual formal analysis of DIBO is the topic of Section 4.*

Assume $\phi : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is the identity, and that the blocked bijection B is chosen such that, for any $x \in \mathbb{F}_2^8$:

- $B_1(x) = (\texttt{02}S)^{-1}(x + c_1)$,
- $B_2(x) = S^{-1}(x + c_2)$,
- $B_3(x) = S^{-1}(x + c_3)$,
- $B_4(x) = (\texttt{03}S)^{-1}(x + c_4)$,

where we recall that $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ is the byte-level `SubBytes` bijection. Then $B \circ \phi \circ T(x + k^\star) = (x + (k^\star + c_1), x + (k^\star + c_2), x + (k^\star + c_3), x + (k^\star + c_4))$, where c_i $(1 \leq i \leq 4)$ are unknown. Hence the secret key $k^\star$ is unconditionally protected. The addition of the random bijection ϕ adds more entropy, thereby *apparently* increasing the secret's protection. Let us already notice that this strategy is incorrect, as we will elaborate on later in Sec. 4.

This argumentation is incomplete since absolute security holds only when there is no linear map ($\phi = Id$). Indeed, white-box diversity and ambiguity are required, as recalled in Sec. 5.2.

3 Distinguishers

In this section we detail the attack methodology against a white-boxed T-box protected by DIBO applied as an internal encoding.

3.1 Methodology

The white-boxing with DIBO generates tables $O_{k^\star}$ which admittedly are difficult to relate to $k^\star$ in the first place. However, owing to the absence of external encoding, it is possible, for all 256 key hypotheses, to attempt to revert the T-box part.

For each hypothesis on $k \in \mathbb{F}_2^8$, the attacker removes the functional part by defining a guess function:

$$\mathcal{A}_k : y \mapsto O_{k^\star}\left(T^{-1}(y) + k\right) = B \circ \phi \circ T\left(T^{-1}(y) + (k + k^\star)\right).$$

This table (represented as a tabulated function, where input y lives in $\mathsf{Im}(T)$, a subset of 256 values from $\mathbb{F}_2^{32}$) has the following properties:

1. It is clear that when the guessed key k matches the key actually concealed in $O_{k^\star}$, then $\mathcal{A}_k = B \circ \phi$, i.e., the T-box part has been successfully peeled off. The guess function $A_{k^\star}$ is thus only the DIBO part (i.e., the secret internal encoding).

2. On the contrary, when the guess key k is different from the actual key $k^\star$, then $\mathcal{A}_k$ is less structured (multiple composition of functions with different structures).

The attack strategy therefore boils down to distinguishing a DIBO from a more random-looking function.

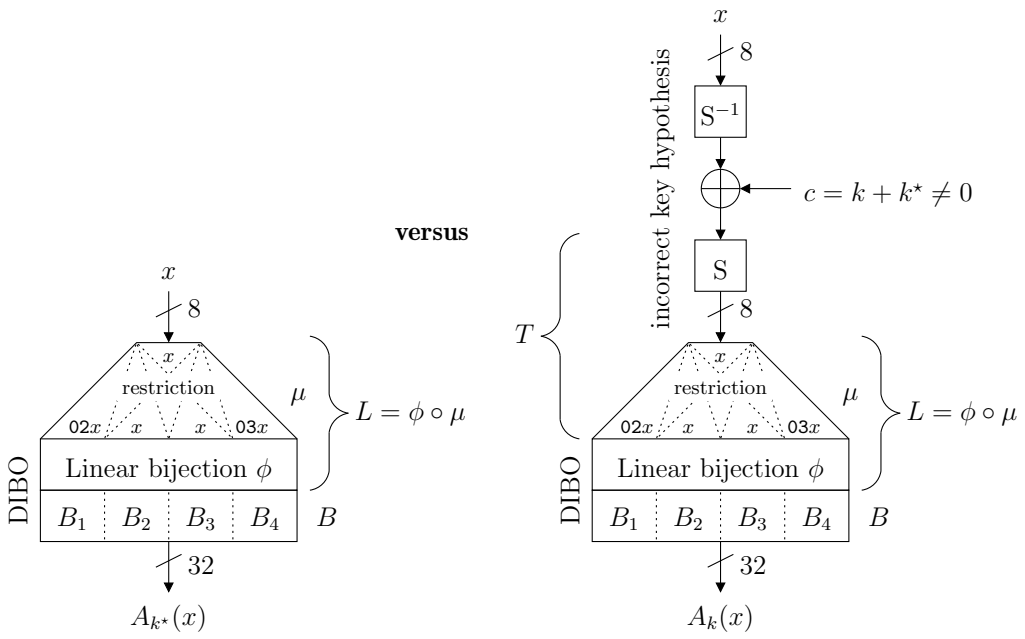

Figure 2: Two WBC situations to be distinguished, cases $A_{k^\star}$ and A_k, for $k \neq k^\star$.

Notice that the guessed function $\mathcal{A}_k$ is a restriction of a function $\mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$. The input values of $\mathcal{A}_k$ live within a unique set (i.e., its support), that is the image of $T : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$, since $\{T(x+k), \forall x \in \mathbb{F}_2^8\} = \{T(x), \forall x \in \mathbb{F}_2^8\} = \mathsf{Im}(T)$.

In practice, it is more convenient to consider the guess function, $A_k : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$, defined as:

$$A_k : x \mapsto \mathcal{A}_k(\mathtt{02}x, x, x, \mathtt{03}x) = \\ O_{k^\star}\left(S^{-1}(x) + k\right) = B \circ \phi \circ T\left(S^{-1}(x) + (k + k^\star)\right).$$

The problem, put chiefly, is to distinguish between the two situations depicted in Fig. 2. In this figure, the attacker does not know the internals (the non-linear functions B_i and ϕ are secret), therefore finding the correct key $k = k^\star$ amounts to deciding whether or not function A_k is the composition of $\mu : x \mapsto (\mathtt{02}x, x, x, \mathtt{03}x)$ and a DIBO. The question therefore amounts to finding some distinguishing property that the composition of μ and an unknown DIBO would feature (case $k = k^\star$), and that a less structured function A_k would not have (case $k \neq k^\star$).

In the sequel, we will denote by $L : \mathbb{F}_{2^8} \to \mathbb{F}_{2^{32}}$ the onto linear function $L(x) = \phi \circ \mu(x) = \phi(\mathtt{02}x, x, x, \mathtt{03}x)$. Therefore, the guess function also rewrites as:

$$A_k : x \mapsto O_k^\star\left(S^{-1}(x) + k\right) = B \circ L \circ S\left(S^{-1}(x) + (k + k^\star)\right). \tag{3}$$

3.2 State-of-the-art attacks

Historically, the first WBC settings were considering that the blocked outputs where only 4×4. Therefore, they had some linearities remaining. Hence the trial (e.g., in DCA [BHMT16]) was to **correlate** the **output bits** of the obfuscated function $O_{k^\star}$ (Fig. 1) with merely the function $x \mapsto T(x + k)$ before white-boxing (for all guesses on the key k).

Remark 1. This method is a direct transposition from the situation of Boolean masked implementations leaking (through their power) the Hamming weight, where the best model for correlation (using so-called Correlation Power Analysis) happens to be the leakage with mask set to zero. This is explained for instance in Lemma 21 of [PRB09].

The second trial was also with a situation of 4×4 blocked outputs. The paper [SMG16] noticed that DCA could fail and hinted (by analogy from the correlation highlighted in Remark 1) that the *correlation-immunity* might characterize the difference between the two situations in Fig. 2. Indeed, it is thus expected that the vectorial Boolean function $A_{k^\star}$ (case for the correct key) is more correlation-immune than others (i.e., A_k, $k \neq k^\star$). This spectral distinguisher is detailed in the next Section 3.3, and then our improved spectral distinguisher is introduced in Sec. 3.4.

Remark 2. The problem of devising the best distinguisher has been solved in some contexts. For instance, when the available information to the attacker is tainted with additive white Gaussian noise (AWGN), the maximum likelihood distinguisher allows to optimize the probability of recovering the secret key under minimum number of observations. This is explained in [HRG14] for unprotected schemes and in [BGHR14] in the masked case. Optimal distinguishers in the context of WBC are different in that they shall maximize the probability to extract the key leveraging one sole observation (the WBC implementation). Such optimal distinguishers are probabilistic, as each and every WBC implementation is generated amongst different randomization parameters (the DIBO part in this paper).

3.3 Spectral distinguisher of Sasdrich et al.

Coming back to the distinguisher issue illustrated in Fig. 2, Sasdrich et al. in [SMG16] came up with an idea in two steps. First of all, in the context of 4×4 blocked outputs, which feature some linearity, a notion of (generalized) correlation-immunity is leveraged. Namely, the $A_{k^\star}$ function is recognized as more likely to be correlation-immune than A_k for $k \neq k^\star$. As highlighted by Lemma 1, the Walsh spectrum of correlation-immune functions has a smaller support. Indeed, there are many zero values; precisely, the number of such zero values is at least $\sum_{i=1}^{t} \binom{n}{i}$ for correlation-immune functions of order t. Hence, it is expected that for the correct key guess $k = k^\star$, the spectrum "weight" is light. Second, a distinguisher which is able to capture this fact is sought. Correlation-immunity is not invariant under composition (on the right) with a linear permutation, since the spectrum is shuffled by such transformation and the structure of the Walsh support is modified (correlation-immunity is only invariant under composition by a permutation of the input coordinates). Correlation-immunity is then not really the proper notion from which a distinguisher can be derived. This (probably) led Sasdrich et al. to opt for the handy Walsh spectrum sum of absolute values distinguisher (or extended Walsh spectrum mean). In this respect, the most likely[3] key $\hat{k}$ according to Sasdrich et al. is:

Definition 4 (Spectral distinguisher of Sasdrich et al. [SMG16, §4.4 at page 200])**.**

$$\hat{k} = \underset{k \in \mathbb{F}_2^8}{\operatorname{argmin}} \sum_{x \in \mathbb{F}_2^8} \sum_{i=1}^{32} \left| W_{(A_k)_i}(x) \right| . \tag{4}$$

This distinguisher has subsequently been reused in [Lee18, LJK20, Ras20]. This distinguisher considers absolute values (norm-1) and not squares (norm-2) of the Walsh spectrum, since according to the Parseval relation, the sum of squares is independent of the choice of

[3]The most likely key, as uncovered by a distinguisher, is denoted by $\hat{k}$ (estimated key). This notation shall not be confused with that of the Fourier transform (introduced earlier in Sec. 1.3).

$f = A_k$:

$$\sum_{x\in\mathbb{F}_2^8} W_{f_i}^2(x) = \sum_x \left(\sum_u (-1)^{f_i(u)+u\cdot x}\right)^2 = \sum_{u,v}(-1)^{f_i(u)+f_i(v)} \sum_x (-1)^{x\cdot(u+v)}$$
$$= \sum_{u,v}(-1)^{f_i(u)+f_i(v)} 2^8 \ \mathbb{1}_{u=v} = 2^8 \ \sum_u (-1)^{f_i(u)+f_i(u)} = 2^{16}.$$

Also, Eqn. (4) considers each coordinate instead of each component function, for two reasons:

1. the computation in Eqn. (4) would otherwise be hardly tractable,
2. the criterion is reminiscent to DCA, which analyses bits one by one.

3.4 Our novel spectral distinguisher

There are two reasons that motivate for an improvement of distinguisher (4):

1. There is no argument in Sasdrich et al.'s paper to consider absolute values of the Walsh spectrum, though their original idea would naturally have led them to consider the number of zeros and not the values themselves;
2. According to the Cauchy-Schwarz inequality and to the Parseval relation, the expression $\sum_{x\in\mathbb{F}_{2^8}} \sum_{i=1}^{32} \left|W_{(A_k)_i}(x)\right|$ in their distinguisher is bounded above by $\sqrt{32\times(2^8)^2\times N}$, where N equals the number of non-zeros among the values of the Walsh transform of the coordinate functions. This gives a clue that N could turn out to be adequate as a distinguisher, with a value all the smaller as the key hypothesis is likely. Clearly, having N small would provide an independent explanation why the distinguisher of Sasdrich et al. is efficient (by the virtue of the aforementioned Cauchy-Schwarz bound).

Therefore, we proceed differently, in that we also noticed the small number of non-zeros in the spectrum of A_k when $k = k^\star$, but did not attempt to attribute it to an alleged correlation-immunity.

Instead, we analyzed the two situations depicted in Fig. 2, and noticed that the linear part in the T-box (namely `MixColumns`) could be merged into the DIBO linear diffusion ϕ. Another way to look at this is that ϕ is evaluated only through the restriction to the image of the function $x \in \mathbb{F}_2^8 \mapsto \mu(x) = (\mathtt{02}x, x, x, \mathtt{03}x) \in \mathbb{F}_2^{32}$. Let us denote by $L = L_1\|L_2\|L_3\|L_4$ the linear function:

$$x \in \mathbb{F}_2^8 \mapsto \phi(\mathtt{02}x, x, x, \mathtt{03}x) = (L_1(x), L_2(x), L_3(x), L_4(x)) = L(x) \in \mathbb{F}_2^{32}, \tag{5}$$

where each L_i is a linear function from $\mathbb{F}_2^8$ to $\mathbb{F}_2^8$, for $1 \le i \le 4$. We recall that:

- each linear application $x \mapsto x$, $x \mapsto \mathtt{02}x$, $x \mapsto \mathtt{03}x$ (all being $\mathbb{F}_2^8 \to \mathbb{F}_2^8$), and
- each linear application $x \mapsto \phi(x)$ (permutation $\mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$)

are bijective. Though, we notice that the L_i from Eqn. (5) may <u>not</u> be bijective. The bijectivity of a linear function L_i from $\mathbb{F}_2^n$ to itself can be characterized by its rank, denoted as $\mathrm{rank}(L_i)$ (which is equal, by definition, to the dimension of its image). We have that $\mathrm{rank}(L_i) \in \{0, \cdot, n\}$, and L_i is bijective if and only if L_i is of full rank, i.e., $\mathrm{rank}(L_i) = n$.

From this central observation, we understood that the depletion in A_k spectrum (when $k = k^\star$) had to be accounted by the non-injectivity of at least one L_i, $1 \le i \le 4$. Remarkably, it happens that this property can also be accounted for by a spectral property:

Lemma 2. *Let $1 \leq i \leq 4$, and let $F = B_i \circ L_i$ a function from $\mathbb{F}_2^8$ to $\mathbb{F}_2^8$ (corresponding to the ith output of $A_{k^\star}$). Then the number of zeros in $W_F(u, v)$ is at least $2^8 - 2^{\mathrm{rank}(L_i)}$.*

Proof. The number of zeros in a spectrum is invariant when composing on the right of the function by any linear bijection. Thus L_i can be replaced by $L_i'(x) = (x_1, \ldots, x_k, 0, \ldots, 0)$, where $k = \mathrm{rank}(L_i)$. Then, the spectrum is equal to zero for all u such that $(u_{k+1}, \ldots, u_n)$ is not all-zero, i.e., $2^n - 2^k$ of them. □

Therefore, our attack consists in the exact enumeration of the number of zeros in the Walsh spectrum. Namely, we leverage the following distinguisher:

Definition 5 (Our spectral distinguisher for WBC based on DIBO)**.**

$$\hat{k} = \underset{k \in \mathbb{F}_2^8}{\mathrm{argmax}} \, \#\Big\{ W_{A_k}(u, v) = 0 \mid u \in \mathbb{F}_2^8, v \in E \Big\} \tag{6}$$

where:

$$E = \{(\mathbb{F}_2^8, 0, 0, 0), (0, \mathbb{F}_2^8, 0, 0), (0, 0, \mathbb{F}_2^8, 0), (0, 0, 0, \mathbb{F}_2^8)\} \subset \mathbb{F}_2^{32}, \tag{7}$$

considering that $(\mathbb{F}_{2^8}, 0, 0, 0)$ stands for $\mathbb{F}_{2^8} \times \{0\}^3$ where 0 is the zero in $\mathbb{F}_{2^8}$.

Notice that our distinguisher restricts the 4×256 values of v to E, because it aims at highlighting the impact of the rank of each L_i on the spectrum of $B_i \circ L_i$. As there is no clue for the attacker which L_i has a rank strictly less than 8 (if any), the sum over all the $1 \leq i \leq 4$ is considered. Another equivalent formulation for our distinguisher (6) is:

$$\begin{aligned}
\hat{k} &= \underset{k \in \mathbb{F}_2^8}{\mathrm{argmax}} \sum_{i=1}^{4} \#\Big\{ W_{O_{k^\star}[1+8i,\ldots,8(i+1)](S^{-1}(\cdot)+(k+k^\star))}(u, v) = 0 \mid u, v \in \mathbb{F}_2^8 \Big\} \\
&= \underset{k \in \mathbb{F}_2^8}{\mathrm{argmax}} \sum_{i=1}^{4} \#\Big\{ W_{B_i \circ L_i \circ S(S^{-1}(\cdot)+(k+k^\star))}(u, v) = 0 \mid u, v \in \mathbb{F}_2^8 \Big\}.
\end{aligned}$$

Remark 3. This expression highlights that our distinguisher depends only in $k+k^\star$, denoted by c in the sequel. The same remark applies to the distinguisher of Sasdrich et al.

A comparison between distinguishers (4) and (6) is provided in the next section 3.5. Our new attack features several advantages. Namely, our differentiators are:

- our attack is not based on an arbitrary distinguisher, but on a motivated one. As we already described, our attack relies on distinguishing a DIBO function from the other (n, m)-functions, leveraging the number of zeros in its Walsh spectrum, which is high when $k = k^\star$ owing to the property related to ranks of L_i linear functions (Lemma 2).

- the distinguisher computation is tractable. Indeed, we recall from Sec. 1.3 that the computation of a Walsh Hadamard transform of an n-input Boolean function requires $n2^n$ additions. Thus the complexity of our distinguisher is: $2^{29} = 2^8 \times 4 \times 2^8 \times (8 \times 2^8)$. In contrast, the complexity of Eqn. (4) is $2^{23} = 2^8 \times 32 \times (8 \times 2^8)$ but uses less information since it only considers the 32 coordinate functions while we consider 4×2^8 component functions; the complexity of the Sasdrich et al. distinguisher would be the same as ours if as many components were studied, instead of only coordinates moreover, since we only count the number of zeros, our complexity in memory would be lower; (still, the natural extension of Sasdrich et al. would not be to consider the output as four bytes, but as a 32-bit vector, hence a total complexity of $2^{51} = 2^8 \times 2^{32} \times (8 \times 2^8)$);

- we provide with a proof that our distinguisher always succeeds provided at least one L_i ($1 \leq i \leq 4$) is neither bijective nor null (see Sec. 4).

Owing to Lemma 2, it is clear at this stage that our attack (counting the zeros in the Walsh spectrum) works only if not all ranks of L_i are full.

Remark 4. In particular, it is therefore wrong that spectral attacks always work, and the DIBO protection pattern is **not** broken. We show in Section 5 how it can be repaired, by forcing to choose ϕ in a given class of values.

3.5 Comparison between our distinguisher and that of Sasdrich et al. [SMG16]

Let us first notice that both distinguishers ([SMG16] and ours) apply in different contexts:

- It is explained in §4.2 of [SMG16] that this distinguisher is geared towards noisy power-analysis leakage (a situation referred to as "grey-box"), whereby correlations can be estimated across numerous (power, electromagnetic, etc.) side-channel traces. As another specificity, the distinguisher of Sasdrich et al. is definitely flexible regarding the internal encoding scheme: it is likely that it applies for a larger class than "pure" DIBO functions.

- Our context is that of genuine "white-box", where the white-boxed table $O_{k^\star}$ is exactly known by the attacker; there is thus no need to collect multiple traces. In particular, the Walsh spectrum can be computed, and zero values can be enumerated without ambiguity. Still, this scenario only works provided the white-boxing rationale is DIBO. As a corollary, it would not apply in the "grey-box" context since it does not allow for accurate zero-value counting (the spectrum is not known).

This being said, we wish nonetheless to compare empirically the distinguishing power of our distinguisher and that of Sasdrich et al., in the canonical white-box context (where the algorithm has been successfully lifted).

In a view to exacerbate the differences between both distinguishers, we allow switching the S-box from the AES ($x \mapsto x^{2^n-2}$) to a Gold function ($x \mapsto x^3$). Indeed, when sticking with the AES S-box, the two distinguishers (4) and (6) happen to perform the same empirically. Indeed, Lemma 2 is very strong: the number of zeros induced by a reduction of the rank of one L_i is the dominant factor. This is not the same when resorting to taking (8,32)-functions whose extended Walsh spectrum (i.e., taken as an absolute value) and stripped of its zero values has significant variance, for example, has two absolute values appearing about the same number of times and one very small and the other very large. Note that this is a property independent of the linear bijection we compose (i.e., which only depends on T in the standard case). The T function does not have this property. For example, the Gold power function ($x \mapsto x^{2^i+1}$) and the Kasami power function ($x \mapsto x^{2^{2t}-2^t+1}$) have either bent or semi-bent components under some conditions; which is not precisely what we want to obtain (absolute values too close), but they could be suitable if we make them undergo the same treatment as the inverse function to get T.

For the sake of clarity, we exhibit explicitly one example of a T-box which can be attacked with our distinguisher and not with the one of Sasdrich et al. It consists in the

choice of ϕ as the permutation of $\mathbb{F}_2^{32}$ of a matrix:

$$\begin{pmatrix}
0&1&0&0&0&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0\\
1&1&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&0&0&1&0&0&1&1&0&0&0&1&0&0&0&1\\
1&1&1&0&0&0&0&0&0&1&1&1&1&0&0&0&1&0&0&0&1&1&0&0&0&1&0&1&0&1&0&0\\
1&0&0&1&1&0&1&1&0&1&0&0&0&1&1&1&0&0&1&1&0&1&0&1&0&1&1&0&0&1&0&1\\
0&0&1&1&0&0&0&0&1&1&1&1&1&1&0&1&1&1&0&1&0&0&1&1&0&1&1&1&0&1&1&0\\
0&1&0&0&1&1&0&1&0&0&1&1&0&1&1&0&0&0&1&0&0&1&1&1&0&1&0&0&1&1&1&0&1\\
1&0&0&1&1&1&0&1&1&1&1&1&0&1&0&0&1&1&1&1&1&0&0&1&0&1&0&1&0&1&1&1\\
1&1&1&1&1&1&0&0&1&1&0&0&0&1&1&1&1&1&0&1&0&1&1&0&0&1&0&0&1&0&0&1\\
1&0&1&1&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&1&0&1&0&1&0&1&1&0&1&1&1&0\\
1&0&0&1&0&0&1&1&1&0&1&1&0&0&1&1&1&0&1&1&0&1&1&1&0&1&0&1&1&0&1&1\\
0&1&1&0&1&0&0&1&1&1&1&0&0&1&1&1&1&0&0&0&1&1&0&1&0&0&0&1&0&1&0&0\\
0&0&0&0&1&1&0&0&1&0&0&1&1&1&0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1&1&1\\
0&0&1&1&1&0&1&1&0&1&0&0&0&0&1&0&1&1&1&1&1&0&1&1&0&1&1&0&0&0&1&0&0\\
1&1&0&0&0&0&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&1&0&1&0&0&1&1&1&0&1&0\\
0&1&1&1&0&1&1&0&0&0&1&1&0&0&0&0&1&0&0&0&0&1&1&1&1&0&1&1&1&0&1&0\\
1&0&1&0&1&1&1&1&0&1&0&1&1&0&1&1&1&0&1&1&1&0&0&0&1&0&1&0&0&0&1&0\\
0&1&1&0&0&0&1&1&0&0&1&1&0&0&0&0&0&1&1&1&0&1&1&0&0&0&0&0&1&0&0&0\\
1&1&0&0&1&0&1&1&1&0&0&0&1&0&0&1&0&1&0&0&0&0&1&0&0&0&0&1&0&1&0&0\\
0&1&1&0&0&1&1&1&0&1&0&0&1&0&0&0&0&1&0&1&1&0&1&0&1&0&1&0&0&0&1&0\\
1&1&1&0&0&0&1&0&0&1&0&1&0&0&0&1&0&0&1&0&1&0&1&1&0&0&1&0&1&1&1&1\\
0&1&0&0&0&1&0&1&0&0&0&0&0&0&0&1&1&0&1&0&1&1&0&1&1&1&0&1&0&1&1&1\\
1&1&0&0&0&1&1&0&0&0&1&1&0&1&1&1&0&0&1&1&0&1&0&0&0&0&1&0&0&1&0&1\\
0&0&1&1&0&0&1&1&1&0&1&1&1&1&0&0&1&1&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0\\
0&0&1&1&1&0&0&1&1&0&1&1&1&0&1&1&0&0&1&0&1&1&0&1&1&0&1&0&0&1&0&1&1\\
0&0&0&1&0&0&1&0&1&0&0&0&1&0&1&1&1&0&1&0&0&1&1&1&0&0&0&0&1&0&0&1\\
0&0&1&1&1&0&0&0&0&1&0&1&1&1&0&1&0&0&1&0&1&0&0&0&1&1&1&1&1&0&1&0\\
1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&1&1\\
0&1&0&1&1&1&1&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&1&1&0&0&1&0&1&1&1\\
0&0&0&1&1&0&1&1&1&0&1&1&1&1&0&1&1&1&0&1&1&1&0&1&1&0&1&0&0&0&1&0\\
1&1&0&1&1&1&1&1&0&1&1&0&0&0&0&1&1&1&0&0&1&0&1&1&1&0&1&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0&1&1&0&1&1&1&1&1&1&0&0&0\\
1&1&0&0&1&0&0&0&0&0&0&1&1&0&0&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{pmatrix}$$

and of B chosen as B_1, B_2, B_3 and B_4, displayed consecutively below:

$$\begin{bmatrix}
158&83&170&101&150&200&118&236&63&135&149&235&109&189&153&73\\
207&171&157&97&188&45&245&138&110&255&8&55&222&37&196&126\\
111&198&168&145&187&5&22&191&125&12&186&179&90&129&223&44\\
64&182&71&162&159&76&59&215&232&18&224&15&152&74&7&89\\
184&249&33&108&156&229&246&214&92&16&96&75&107&176&4&116\\
61&124&52&66&26&234&154&227&40&13&3&106&24&30&228&60\\
56&139&122&19&190&136&54&204&80&51&2&220&104&134&86&10\\
144&181&238&41&14&27&50&231&20&180&46&173&251&123&212&178\\
62&210&166&130&155&137&43&199&146&161&112&206&98&160&93&201\\
244&113&0&94&95&69&49&48&85&253&141&23&225&143&78&100\\
131&205&254&6&68&84&121&239&219&217&247&194&91&218&233&11\\
119&102&35&57&169&65&1&120&203&42&105&132&221&17&38&133\\
53&164&250&128&34&28&183&114&163&151&202&31&209&127&185&226\\
237&32&167&142&213&147&29&177&241&99&82&252&175&79&197&208\\
115&148&248&72&77&25&165&81&240&174&243&39&230&193&58&140\\
88&216&70&87&36&242&21&211&9&103&195&67&192&117&47&172
\end{bmatrix}$$

$$\begin{bmatrix}
247&98&219&143&78&15&110&11&161&102&187&217&183&221&162&40\\
165&196&175&37&86&171&107&212&45&82&21&153&97&113&166&119\\
246&240&173&50&104&130&62&151&222&46&95&115&77&1&148&154\\
144&100&155&138&229&231&134&12&19&125&218&210&121&109&152&228\\
31&188&202&10&47&205&76&75&65&42&3&255&252&206&90&9\\
238&137&131&223&224&27&96&57&159&163&54&225&253&156&172&245\\
23&56&185&60&213&108&220&71&239&6&99&226&192&79&157&127\\
7&118&8&208&38&132&169&215&191&184&49&193&227&124&204&244\\
59&182&48&53&70&170&216&55&234&142&248&214&201&241&179&52\\
74&26&200&4&139&178&250&5&141&61&135&207&122&22&68&20\\
243&14&32&230&64&180&236&81&58&13&194&72&147&160&87&123\\
195&211&242&51&233&24&94&164&106&16&63&128&186&105&28&73\\
89&149&174&93&41&177&83&43&92&18&199&198&145&158&150&249\\
39&167&85&235&111&66&117&44&140&101&69&2&84&17&146&254\\
29&30&114&237&103&67&25&176&251&116&168&91&120&35&112&209\\
129&189&136&190&80&181&126&197&34&203&0&36&232&133&33&88
\end{bmatrix}$$

$$\begin{bmatrix}
140&31&1&27&187&189&157&222&86&219&56&14&16&212&111&11\\
192&130&147&0&13&19&84&68&197&60&126&143&210&43&149&8\\
153&30&151&79&99&144&195&50&54&213&39&232&150&240&208&122\\
108&246&146&103&85&41&44&201&167&9&253&2&181&245&23&209\\
154&124&20&162&137&152&249&156&203&75&37&48&106&95&165&239\\
228&76&64&52&218&139&160&242&191&170&29&142&216&202&186&7\\
116&18&33&107&55&49&113&36&40&110&117&24&193&63&121&17\\
217&58&69&184&131&175&252&229&236&238&10&67&42&237&199&5\\
45&72&62&177&251&250&133&129&127&221&21&25&136&35&81&77\\
176&211&214&248&101&102&235&168&220&241&28&134&225&70&223&3\\
34&88&182&120&155&112&196&125&180&80&12&161&254&230&174&159\\
100&78&92&114&66&200&65&135&82&171&53&166&169&15&215&224\\
6&94&71&141&138&128&255&47&188&61&46&51&90&98&204&22\\
118&244&183&73&227&247&109&158&205&4&190&105&164&119&207&231\\
234&148&194&163&145&91&93&185&132&233&173&178&57&59&172&97\\
198&115&32&206&26&83&96&87&123&226&243&74&89&38&179&104
\end{bmatrix}$$

$$\begin{bmatrix}
46&136&28&212&45&27&249&31&58&12&77&179&213&108&42&14\\
185&142&99&44&50&238&160&209&87&255&176&55&236&25&156&189\\
123&175&139&82&70&111&57&119&73&170&3&216&66&227&52&197\\
203&184&252&49&24&35&169&163&165&251&53&122&183&153&246&214\\
244&223&231&21&192&161&177&215&250&186&201&164&155&124&221&17\\
104&173&208&26&63&196&172&72&152&65&235&39&16&88&199&97\\
121&94&134&103&232&242&128&159&41&190&130&60&86&168&171&106\\
20&4&129&141&148&205&240&146&180&29&245&207&69&137&195&8\\
107&19&101&211&68&239&198&109&233&81&54&34&187&30&78&11\\
234&217&174&98&22&7&110&91&23&162&71&167&132&220&228&117\\
145&206&222&56&224&84&112&144&67&61&131&74&0&188&80&135\\
226&182&204&210&5&127&83&36&202&147&150&90&229&64&93&248\\
138&219&32&96&95&113&59&37&191&143&254&158&154&40&48&100\\
241&105&92&13&89&18&120&149&218&47&230&62&79&43&193&181\\
116&247&200&115&9&157&76&2&51&178&237&126&85&114&75&166\\
102&243&125&6&10&1&38&133&253&151&33&118&15&140&225&194
\end{bmatrix}$$

which results in $O_{k^\star}$ laid out as follows:

```
cd6f8aa7 2efa21fa 2ebbb93a cd0d4b09 2e7e26ba 2ecbf30f 2e3cc2c0 2e76fa27
2e7e26ba 6cab5331 cd45d7c6 b4250117 b450ed3e 2e49414a b4bc7f8b 2ed54d52
b43437cc cd0d4b09 2e910891 6ca73669 b45bf0ea 2e43221f cd8ee143 6cba6313
b459d6d9 6cf8e58d 2ed54d52 cd77c18c cd3b8597 cd8ee143 b4f6423d b4d2cfbe
cdc082f5 b459d6d9 6c9ba29b 2e49414a 6ca73669 cdd42396 2efa21fa b4e80e2e
6cd119a6 b44a0c71 cde914bf 2e41ec47 b4642afb b4e6b656 cd1149a2 cde914bf
6c23ade4 b4e6b656 b4fba0a9 6cd119a6 2edebae8 2eb36e87 2edebae8 2e317049
2e3cc2c0 6c82a938 cd45d7c6 b4940a84 2e5567a3 b45bf0ea 2ea05eac b4d2cfbe
2ed54d52 cd4f7ca5 b41fdf32 6cb0a86b b4edd489 b4aef5cb cdd8b080 cd0d4b09
cd4d7b2f b4e6b656 6c803088 2e2a2f57 2e2a2f57 b481d07a 6cd119a6 cd077d2d
cde914bf b47a5d5f 2e2a2f57 6c1e81f6 b408cd23 2e8cf79e b4b1c462 2edebae8
cd1a394c 2e7e26ba 2e1b921d cdfcabe2 cd709c46 cdd647e6 cd8ee143 cd578e48
b4aef5cb cd8d4fb3 6ca73669 2ecc3b2c 6cfffd39 cd45d7c6 6caa756a cdfcabe2
2e679f58 cdfcabe2 cd2f281c 2e3cc2c0 b459d6d9 b474fe30 b41fdf32 b49dd104
b4aef5cb 6c0b1744 2efa21fa cde02ef1 b43769b4 b4d2cfbe cdf904ae cd709c46
cd7f25b7 b41fdf32 6c5e1dce 2e49414a b45bf0ea 2ec8bf4d cd709c46 6c7978e5
2ec8bf4d 2e43221f b4f6423d b43769b4 cd2f281c 2e76fa27 b4250117 6cfffd39
2e317049 b4b1c462 b4b1c462 2eb36e87 b4642afb cd4d7b2f cd077d2d b44a0c71
cd4d7b2f 6c23ade4 2e41ec47 b47a5d5f cd077d2d b4fba0a9 cd1149a2 b47a5d5f
2e910891 2ecc3b2c cde02ef1 cd6f8aa7 6cf8e58d b474fe30 6c5e1dce b450ed3e
6cba6313 cd578e48 b43769b4 2ea05eac cdc082f5 b474fe30 cd4f7ca5 b4bc7f8b
2ec8bf4d 2e5567a3 cd3b8597 cd578e48 cdd8b080 2ebbb93a cdd42396 2ecc3b2c
6c9ba29b 6c5e1dce cd4f7ca5 cd77c18c 2ecbf30f cd1a394c 6cfffd39 b4940a84
cd1a394c 6cab5331 6c82a938 cd2f281c cde02ef1 b4e80e2e b43437cc cdd8b080
2e5567a3 2e43221f cdd647e6 cdf904ae b49dd104 6cb0a86b b450ed3e 6c9ba29b
2ea05eac b4f6423d cdd647e6 6c7978e5 cd8d4fb3 b4edd489 cd6f8aa7 b4e80e2e
6cf8e58d cdc082f5 cd7f25b7 6cb0a86b 2e1b921d 6caa756a 6c82a938 2e76fa27
b4250117 b4940a84 2e679f58 2e1b921d b4edd489 6c0b1744 cdd42396 2e910891
cdf904ae 6c7978e5 6cba6313 cd3b8597 2ecbf30f 6cab5331 6caa756a 2e679f58
b408cd23 b408cd23 2e317049 2eb36e87 2e41ec47 cd1149a2 6c803088 b481d07a
6c1e81f6 6c803088 b4fba0a9 b44a0c71 6c23ade4 b4642afb 6c1e81f6 b481d07a
cd8d4fb3 6c0b1744 2ebbb93a b43437cc b4bc7f8b cd77c18c b49dd104 cd7f25b7
```

(8)

This table shows the 256 entries of $O_{k^\star}(x)$, for $x \in \{0, 1, \ldots, 255\}$; the values are encoded in hexadecimal. For instance, $O_{k^\star}(0) =$ `0xcd6f8aa7`, $O_{k^\star}(1) =$ `0x2efa21fa`, $\ldots$, $O_{k^\star}(255) =$ `0xcd7f25b7`.

Our distinguisher (6) finds the correct key $k^\star =$ `0x55`, whereas the distinguisher (4) of Sasdrich et al. fails. Nonetheless, the correct key is ranked at 6th position, which is close to being the best.

One example is insufficient to draw significant conclusions. Hence, we propose to compare the distinguisher's performance in average and for examples on several ranks, as shown in Fig. 3. The distinguisher of Sasdrich et al. is computed negatively, so that the figure of merit is: "the larger the distinguisher, the more likely the key".

Various metrics proposed in Fig. 3 show that our distinguisher is able (in most cases) to distinguish better than that of Sasdrich et al. Those metrics are respectively:

- Success rate: probability to extract the correct key;
- Guessing entropy: rank of the correct key amongst the guessed candidates;
- Value of the distinguisher: gives some hint about the distinguishing margin, later on defined as (15) in Sec. A.2.1.

Notice that the situation depicted in Fig. 3 is not the one analyzed mathematically in Sec. 4 because the S-box is different.

Now, let us underline the two compelling reasons why it is important to study our (motivated) distinguisher:

1. we managed to come up with the proof (Sec. 4) that our distinguisher *always* works if at least one L_i is not bijective, hence a so-called "structural cryptanalysis", and
2. this gave us the solution to repair the DIBO scheme (Sec. 5), by ensuring that all four L_i are invertible.

Distinguisher: A_k spectrum mean absolute value ((4) from [SMG16])

Distinguisher: number of zeros in the A_k Walsh spectrum ((6), this work)

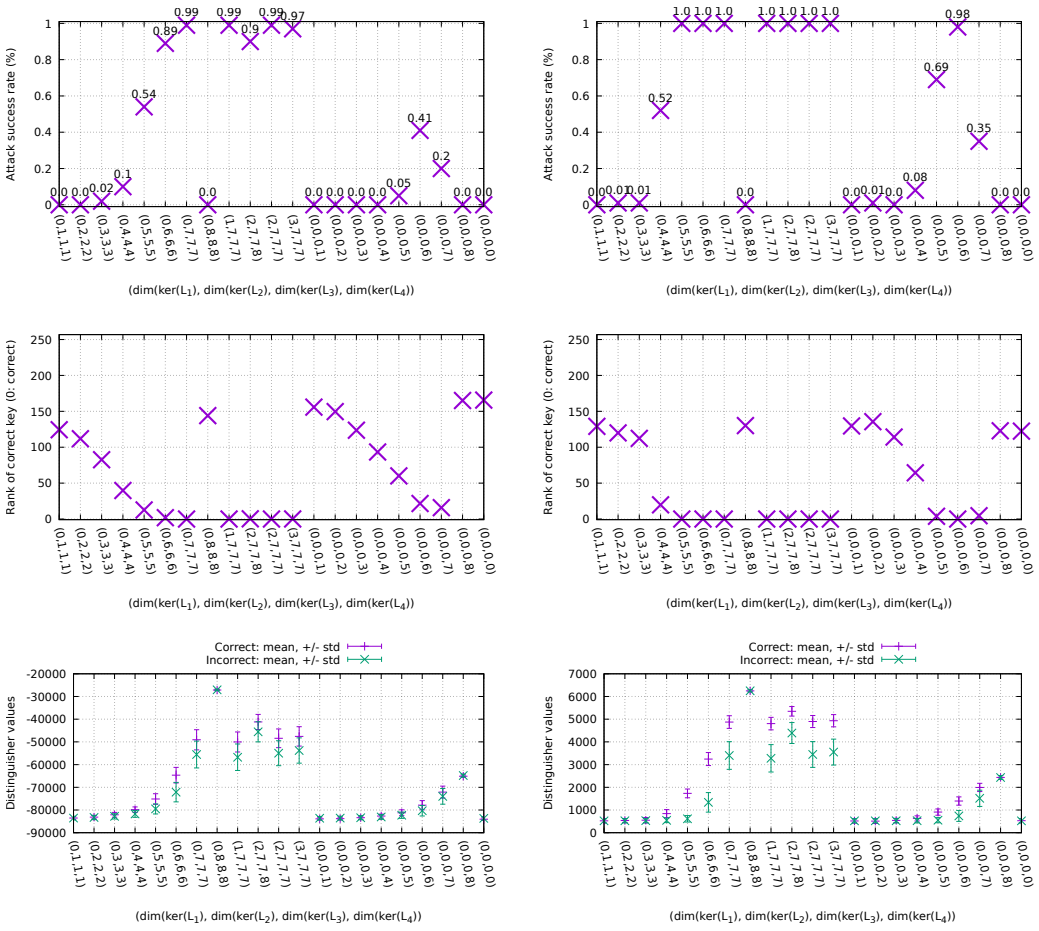

Figure 3: Distinguisher performance comparison between [SMG16] and ours

4 Mathematical proof of the cryptanalysis

Let n be an even integer, that will be equal to 8 in practice. We identify $\mathbb{F}_2^n$ with the finite field $\mathbb{F}_{2^n}$ and we recall that $tr(x)$ denotes the trace of any element x of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$: $tr(x) = \sum_{i=0}^{n-1} x^{2^i}$. Let $\mathcal{B}_n$ be the vector space over $\mathbb{F}_2$ of all Boolean functions over $\mathbb{F}_{2^n}$, equipped with the inner product $f \cdot g = \sum_{x \in \mathbb{F}_{2^n}} f(x)g(x)$. For a vector subspace $\mathcal{V}$ of $\mathcal{B}_n$, we denote by $\mathcal{V}^\perp$ its dual (i.e. orthogonal): $\mathcal{V}^\perp = \{f \in \mathcal{B}_n \mid \forall g \in \mathcal{V},\ f \cdot g = 0\}$. Let $RM(r,n)$ be the Reed-Muller code of length 2^n and order r, that is, the vector space of all n-variable Boolean functions of algebraic degree at most r. We recall that the dual code $RM(r,n)^\perp$ of $RM(r,n)$ equals $RM(n-r-1,n)$.

In this section, we fix $1 \leq i \leq 4$. We fix also $v = (v_1, v_2, v_3, v_4) \in \mathbb{F}_{2^n}^4$ where $v_i \neq 0$ and $v_j = 0$ for $j \neq i$.

4.1 Setting up the mathematical criterion

We are interested in counting the number of those $u \in \mathbb{F}_{2^n}$ such that $W_F(u,v) = 0$ where $F(x)$ equals the concatenation of the vectors equal to $B_i \circ L_i(S(S^{-1}(x)+c))$ for $i = 1, \ldots, 4$, where B_i is a $(8,8)$-function, $L = (L_1, L_2, L_3, L_4)$ is a linear injective $(8,32)$-function, S is the `SubBytes` function (at byte level, namely $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8$) and $c = k + k^\star$:

$$W_F(u,v) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{tr(ux)+tr(v_i B_i \circ L_i(S(S^{-1}(x)+c)))}.$$

Now, let us make two observations:

- Without loss of generality, we may take $S(x)$ equal to the multiplicative inverse function x^{-1}, since the composition of x^{-1} with a linear permutation only permutes the values of the Walsh transform. More precisely, the size of $\{u \in \mathbb{F}_{2^n} \mid W_F(u,v) = 0\}$ is equal to the number of u in $\mathbb{F}_{2^n}$ such that the Walsh transform of $x \in \mathbb{F}_{2^n} \mapsto tr(v_i B_i \circ L_i((x^{-1}+c)^{-1}))$ is equal to 0 at u.
- Let us denote by r the rank of L_i. Then, $L_i(x) = \sum_{l=1}^{r} tr(b_l x) a_l$ for some linearly independent elements $b_1, \ldots, b_r$ of $\mathbb{F}_2^n$ and some linearly independent elements $a_1, \ldots, a_r$ of $\mathbb{F}_2^n$. Therefore, $tr(v_i B_i \circ L_i(x))$ may be viewed as the composition $g(tr(b_1 x), \ldots, tr(b_r x))$ of some r-variable Boolean function g, depending only on v and the a_l's, with the linear map $x \in \mathbb{F}_{2^n} \mapsto (tr(b_1 x), \ldots, tr(b_r x)) \in \mathbb{F}_2^r$.

Following the above observations, the number of u in $\mathbb{F}_{2^n}$ such that $W_F(u,v) = 0$ is equal to the number of u in $\mathbb{F}_{2^n}$ such that the Walsh transform

$$W_{g_c}(u) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{tr(ux)+g(tr(b_1(x^{-1}+c)^{-1}),\ldots,tr(b_r(x^{-1}+c)^{-1})} \tag{9}$$

is equal to 0 where g_c is defined as $g_c(x) = g(tr(b_1(x^{-1}+c)^{-1}), \ldots, tr(b_r(x^{-1}+c)^{-1})) = tr(v_i B_i \circ L_i((x^{-1}+c)^{-1}))$.

We are now in position to state the key problem, on which relies the attack proposed in the paper, and that we shall address successfully.

Problem 1. *Fix $1 \leq r \leq 7$ and linearly independent elements $b_1, \ldots, b_r$ of $\mathbb{F}_{2^n}$. Let g be a non constant r-variable Boolean function. Prove that the size of $(W_{g_c})^{-1}(0) = \{u \in \mathbb{F}_{2^n} \mid W_{g_c}(u) = 0\}$ is less than the size of $(W_{g_0})^{-1}(0) = \{u \in \mathbb{F}_{2^n} \mid W_{g_0}(u) = 0\}$ for any $c \neq 0$.*

Remark 5. When $r = 0$, g_c is a constant, hence the number of zeros of W_{g_c} is $2^n - 1$ irrespective $c = 0$ or $c \neq 0$. This yields to a tie in our distinguisher, namely more than one key (namely, the set of all possible keys) is returned. Nonetheless, in this case, we can shift the focus to another output byte $1 \leq i \leq 4$, such that $\mathrm{rank}(L_i) \notin \{0, 8\}$.

We firstly study the case $r = 1$. In that case, there are only two possibilities: $g(tr(b_1x)) = tr(b_1x)$ or $g(tr(b_1x)) = tr(b_1x) + 1$. If $c = 0$, then $W_{g_c}(u)$ vanishes at all points except $u = b_1$, and if $c \neq 0$, then $W_{g_c}(u)$ vanishes at less that $2^n - 1$ points, since it is well known that the only functions that vanish at $2^n - 1$ points are affine functions, and g_c is not affine.

We investigate the case $2 \leq r \leq 7$ in the two next subsections considering firstly the case where $c = 0$ ($k = k^\star$) and secondly the case $c \neq 0$ ($k \neq k^\star$).

4.2 When $k = k^\star$

When $c = k + k^\star = 0$, we can state a simple lower bound on the number of 0 of W_{g_0} depending only on r. According to Lemma 2, let g be an r-variable Boolean function. The size of $(W_{g_0})^{-1}(0)$ is larger than or equal to $2^n - 2^r$.

Remark 6. This result actually works for any value $0 \leq r \leq 8$.

Therefore, we need now to prove that the number of zeros in W_{g_c} is strictly less than $2^n - 2^r$ when $c \neq 0$.

4.3 When $k \neq k^\star$

4.3.1 Upper bound on the number of zeros in W_{g_c}

Concerning $c = k + k^\star \neq 0$, we use in our study a result established in [BC99] and not so well known.

Theorem 1 ([Car21, §2.3.5, page 70]). *Let f be a Boolean function over $\mathbb{F}_{2^n}$. The size of the Fourier-Hadamard support $\{u \in \mathbb{F}_{2^n}; \hat{f}(u) = \sum_{x \in \mathbb{F}_{2^n}} f(x)(-1)^{tr(ux)} \neq 0\}$ is larger than or equal to $2^{d^\circ_{alg} f}$.*

Let us say a few words on how this bound can be proven, since this will have an impact below. The size of the support $\{u \in \mathbb{F}_2^n;\ \widehat{f}(u) \neq 0\}$ is larger than or equal to the size of the support of the Fourier-Hadamard transform of any restriction g of f, obtained by keeping constant some of its input bits (we do not give an argument for this fact, which is either related to Cayley graphs or deduced from the so-called Poisson summation formula, and would lead us too far from our subject, but these arguments can be found in Subsection 2.3.5 of [Car21]), and choosing a multi-index I of size $d^\circ_{alg} f$ such that x^I is part of the algebraic normal form of f, the restriction obtained by fixing to 0 the coordinates of indices lying outside I has odd weight and its Fourier-Hadamard transform takes therefore nonzero values only. We observe that, in general, there will be many more elements in the support of the Fourier-Hadamard transform of f than in that of g, and the bound of Theorem 1 is often far from being an equality, but it will be enough to reach our goal. We deduce from the theorem:

Corollary 1. *Let g be an r-variable Boolean function. Let $b_1, \ldots, b_r$ be linearly independent elements of $\mathbb{F}_{2^n}$. Let $c \neq 0$. Then, the size of $(W_{g_c})^{-1}(0)$ is less than or equal to $2^n - 2^d + 1$ where d denotes the algebraic degree of $g_c : x \mapsto g(tr(b_1(x^{-1} + c)^{-1}), \ldots, tr(b_r(x^{-1} + c)^{-1}))$.*

Proof. According to Theorem 1, there are at most $2^n - 2^d$ elements u in $\mathbb{F}_{2^n}$ such that $\hat{f}(u) = 0$. And since for $u \neq 0$ we have $W_{g_c}(u) = -2\widehat{g_c}(u)$, there are at most $2^n - 2^d + 1$ such inputs for which $W_{g_c}(u) = 0$. □

Theorem 1 and Corollary 1 give together a positive answer to Problem 1 for any r-variable Boolean function g such that $2^n - 2^r > 2^n - 2^{d^\circ_{alg} g_c} + 1$, that is, such that $2^{d^\circ_{alg} g_c} > 2^r + 1$, or equivalently, $d^\circ_{alg} g_c \geq r + 1$. Note that in the case $d^\circ_{alg} g_c = r$, we have

that the size of $(W_{g_c})^{-1}(0)$ is less than or equal to $2^n - 2^r + 1$ and in most cases, it will be less than $2^n - 2^r - 1$ (because, as we explained, the bound of [BC99] is very seldom an equality or almost an equality, and also because there are other reasons that we give in the remark at the end of the present section why there must be a difference between the cases $c = 0$ and $c \neq 0$).

4.3.2 Study of $d^\circ_{alg} g_c$

According to the preceding arguments, we need to study the algebraic degree of

$$g_c : x \mapsto g(tr(b_1(x^{-1} + c)^{-1}), \ldots, tr(b_r(x^{-1} + c)^{-1}))$$

To make our expressions lighter, we set $h(x) = g(tr(b_1 x), \ldots, tr(b_r x))$. Note that h is an n-variable of algebraic degree at most r. We now prove that there is a high probability of getting $x \mapsto h((x^{-1} + c)^{-1})$ of algebraic degree $n - 1$.

We for that equip the set $\mathcal{B}_n$ of all n-variable Boolean functions with the uniform probability Pr (defined as $\Pr(A) = |A|/|\mathcal{B}_n|$). We denote $Pr[A \mid B]$ the conditional probability of A given B. With this notation, we prove

Theorem 2. *For any $r \leq n - 1$ and $c \neq 0$,*

$$\Pr[\{d^\circ_{alg} h((x^{-1} + c)^{-1}) = n - 1\} \mid \{d^\circ_{alg} h \leq r\}] \geq 1 - 2^{-n}$$

where $\{d^\circ_{alg} g_c = n - 1\}$ denotes the subset of $\mathcal{B}_r$ formed by all r-variable Boolean functions g such that the function g_c is of algebraic degree $n - 1$.

Proof. The probability distribution that we consider for h is uniform over the vector space of Boolean functions of algebraic degree at most r over $\mathbb{F}_{2^n}$. It is shown in [Car20] that given a permutation F and a function G we have:

$$d^\circ_{alg}(G \circ F) = \max_{\substack{i \in \{0,\ldots,2^n-2\} \\ d^\circ_{alg}((F^{-1}(y))^i G(y)) = n}} \big(n - w_2(i)\big), \tag{10}$$

where F^{-1} is the compositional inverse of F (beware that "-1" represents here the compositional inverse while in "$(x^{-1} + c)^{-1}$", it represents the multiplicative inverse). Taking here $F(x) = (x^{-1} + c)^{-1}$, which is its own compositional inverse, we deduce:

$$d^\circ_{alg} h((x^{-1} + c)^{-1}) = \max_{\substack{i \in \{0,\ldots,2^n-2\} \\ d^\circ_{alg}((x^{-1}+c)^{-i} h(x)) = n}} (n - w_2(i)).$$

Thus, since we know that $d^\circ_{alg} h \leq n - 1$ (which is equivalent to saying that the sum of the values taken by $h(x)$ when x ranges over $\mathbb{F}_{2^n}$ equals 0), and then $d^\circ_{alg} h((x^{-1}+c)^{-1}) \leq n-1$ since $(x^{-1} + c)^{-1}$ is a permutation, we have $d^\circ_{alg} h((x^{-1} + c)^{-1}) \leq n - 2$ if and only if

$$\forall i \in \{0, \ldots, n - 1\},\ d^\circ_{alg}((x^{-1} + c)^{-2^i} h(x)) < n.$$

Since h is a Boolean Function, this is equivalent to $d^\circ_{alg}((x^{-1} + c)^{-1} h(x)) < n$, that is,

$$\sum_{x \in \mathbb{F}_{2^n}} (x^{-1} + c)^{-1} h(x) = 0$$

or equivalently:

$$\forall \beta \in \mathbb{F}_{2^n}, \quad \sum_{x \in \mathbb{F}_{2^n}} tr(\beta(x^{-1} + c)^{-1}) h(x) = 0. \tag{11}$$

Let E be the following subspace of $\mathcal{B}_n$: $E = \{h_\beta := tr(\beta(x^{-1}+c)^{-1}),\ \beta \in \mathbb{F}_{2^n}\}$. Since $(x^{-1}+c)^{-1}$ is a permutation of $\mathbb{F}_{2^n}$, E is a subspace of dimension n. Condition (11) says that h lies in the vector space $E^\perp \cap RM(r,n)$. The dual of $E^\perp \cap RM(r,n)$ is $E + RM(n-r-1,n)$.

We shall show that any function h_β has algebraic degree $n-1$. This will imply that the sum $E + RM(n-r-1,n)$ is direct and allow us to deduce its dimension. Let us now compute the algebraic degree of h_β. Before all, observe that $tr(\beta(x^{-1}+c)^{-1}) = tr(\beta c^{-1}((cx)^{-1}+1)^{-1})$. Thus, since two affinely equivalent Boolean functions have the same algebraic degree, one can suppose without loss of generality, that $c=1$ when determining the algebraic degree of $tr(\beta(x^{-1}+c)^{-1})$. Now, observe that,

$$(x^{-1}+1)^{-1} = \left((x^{2^n-2}+1)^{2^{n-1}-1}\right)^2 = \left(\sum_{w=0}^{2^{n-1}-1} x^{(2^n-2)w}\right)^2 = 1 + \left(\sum_{w=1}^{2^{n-1}-1} x^{2^n-1-w}\right)^2.$$

For $\beta \neq 0$, we have then:

$$tr\left(\beta(x^{-1}+1)^{-1}\right) = tr(\beta) + \sum_{w=1}^{2^{n-1}-1} tr\left(\beta^{1/2} x^{2^n-1-w}\right).$$

There is no exponent of 2-weight n in this sum and the terms whose exponents have 2-weight $n-1$ are those corresponding to $w = 2^i$ for $i = 0,\ldots,n-2$. The part in the above expression made of the terms with exponents of 2-weight $n-1$ equals then:

$$\begin{aligned}\sum_{i=0}^{n-2} tr\left(\beta^{1/2} x^{2^n-1-2^i}\right) &= tr\left(\left(\sum_{i=0}^{n-2} \beta^{2^{n-i-2}}\right) x^{2^{n-1}-1}\right) \\ &\overset{j=n-i-2}{=} tr\left(\left(\sum_{j=0}^{n-2} \beta^{2^j}\right) x^{2^{n-1}-1}\right).\end{aligned}$$

This vanishes if and only if $\sum_{j=0}^{n-2} \beta^{2^j} = tr(\beta) + \beta^{2^{n-1}} = 0$, that is, $\beta = 0$ since $tr(\beta) + \beta^{2^{n-1}} = 0$ implies $\beta \in \mathbb{F}_2$ and n being even, $\sum_{j=0}^{n-2} \beta^{2^j}$ equals 1 for $\beta = 1$. Hence, we deduce that every nonzero component function $tr(\beta(x^{-1}+c)^{-1})$ is of algebraic degree $n-1$ for any $\beta \neq 0$.

That implies that the sum $E + RM(n-r-1,n)$ is a direct sum, any element of $RM(n-r-1,n)$ having algebraic degree at most $n-r-1 < n-1$ for $r \geq 1$. Thus, the dimension of $E^\perp \cap RM(r,n)$ is $2^n - n - \sum_{k=0}^{n-r-1} \binom{n}{k} = -n + \sum_{k=n-r}^{n} \binom{n}{k} = -n + \sum_{k=0}^{r} \binom{n}{k}$. Thus, the conditional probability that $h((x^{-1}+c)^{-1})$ is of algebraic degree at most $n-2$ given that h lies in $RM(r,n)$ is $2^{-n+\sum_{k=0}^{r}\binom{n}{k}} / 2^{\sum_{k=0}^{r}\binom{n}{k}} = 2^{-n}$. □

A numerical validation is computed empirically over 10,000 random draws. We get that, for $n=8$ and $r=n-1$, $\Pr[\{d^\circ_{alg} g_c = n-1\}] \approx 0.997$, which is indeed better than the bound of Theorem 2 ($1-2^{-n} \approx 0.996$). Besides, this confirms that our spectral attack (definition 5) works in an overwhelming number of situations.

4.4 A final remark

There is another possible approach, to complete our view of Problem 1. Consider a vectorial function of the form $F = B \circ L$ where L is an injective $(8,32)$-function and B is the concatenation of four $(8,8)$-functions $B_1,\ldots,B_4$. Let $E_i = \{x \in \mathbb{F}_2^8;\ L(x) \in \{(0,\ldots,0)\} \times \mathbb{F}_2^8 \times \{(0,\ldots,0)\}\}$ where the number of zeros on the left is $i-1$ and the number of zeros on the right is $n-i$. Then L being injective, the vector spaces E_i are

in a direct sum equal to $\mathbb{F}_2^8$ (and the sum of their dimensions equals then 8). We have $F(\sum_{i=1}^4 x_i) = \sum_{i=1}^4 B_i \circ L(x_i) = \sum_{i=1}^4 F(x_i)$ with $x_i \in E_i$. We can see that in the expression of F, there is no product of one coordinate of x_i with those of other x_j. On the other hand, there can be products between the coordinates of the same x_i. This is characterized by the Walsh transform of F: for any $u \in \mathbb{F}_2^8$ and $v \in \mathbb{F}_2^{32}$, we have $W_F(u,v) = \sum_{x_i \in E_i, \forall i=1,\ldots,4} (-1)^{\sum_{i=1}^4 (v \cdot F(x_i) + u \cdot x_i)} = \prod_{i=1}^4 \sum_{x_i \in E_i} (-1)^{v \cdot F(x_i) + u \cdot x_i}$. We can see that thanks to the fact that 0 is absorbent, W_F has more zeros than a random function. But it is more difficult to precisely compare the cases $c = 0$ and $c \neq 0$ with this approach than with the previous one.

5 Our countermeasure: conditions for DIBO to resist our spectral attack

5.1 Average insecurity of DIBO on AES

From the previous analysis, one can state the following

Countermeasure 1. *A DIBO obfuscation scheme is **immune** to our attack provided **all four** linear functions L_i, $1 \leq i \leq 4$, are invertible.*

In general, many linear $L_i : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ are permutations. Namely, the number of permutations is $\prod_{i=0}^7 (2^8 - 2^i)$, therefore the proportion of invertible linear mappings in $\mathbb{F}_2^8$ is $2^{-8^2} \prod_{i=0}^7 (2^8 - 2^i) \approx 0.290$.

But now, for a DIBO obfuscations scheme to be attackable by our distinguisher, it suffices that at least one L_i is non-invertible. Hence the proportion of vulnerable DIBO is:

$$1 - \left(\prod_{i=0}^{7} \left(1 - 2^{i-8} \right) \right)^4 \approx 0.993. \tag{12}$$

Thus, a WBC implementation of AES leveraging DIBO with random (unconstrained) L has an overwhelming probability ($> 99\%$) to be attackable.

5.2 The White-Box diversity and ambiguity

The condition to resist our attack is therefore to choose $L : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ such that all four L_i are permutations. The number of such permutations is cryptographically large: for a given i, the number of permutations L_i is

$$\prod_{i=0}^{7} \left(2^8 - 2^i \right) \approx 2^{62}.$$

The estimate of the diversity (defined in [CEJvO02b, §4.1]) of one byte of output of $O_{k^\star}$ is the number of bijections in $\mathbb{F}_2^8$, which is $256! \approx 2^{1684}$. Indeed, $B_i \circ \phi_i$ boils down to a bijection in $\mathbb{F}_{256}$. Therefore, by restricting the choice of ϕ to those such that L_i has maximum rank (or null rank) for all $1 \leq i \leq 4$, our spectral attack (and therefore also that of Sasdrich et al. [SMG16]) is efficiently avoided.

Ambiguity follows from a similar argument as in Sec. 2.3: Let $A_{k^\star}$ the guess function (3) for the correct key.

$$A_{k^\star}(x) = (B_i \circ L_i)(x)$$

where L_i is a permutation. (Indeed, we ignore the trivial case where $L_i = 0$.) Then, $A_{k^\star}$ has the expression of a licit A_k, for any $k \neq k^\star$. Indeed, let $c = c + k^\star$, one has

$$A_{k^\star}(x) = (B'_i \circ L'_i) \circ S(S^{-1}(x) + c),$$

where L'_i is any permutation, and $B'_i(x) = B_i \circ L_i \circ S(S^{-1}(L'^{-1}_i(x)) + c)$ is a bijection.

5.3 Resistance to other attacks

Restricting the values that L takes on (from $GL(32, \mathbb{F}_2)$ to this group modulo $\mathrm{rank}(L_i) = 8$, namely the countermeasure 1 considered at the beginning of Sub. 5.1) does not open specific vulnerabilities against correlation attacks.

For instance, in front of the DCA, which is a correlation attack (see fifth step in [BHMT16, §4, page 226]), it is well-known [PRB09] that the optimal attack is the expectation of the model over all randomization parameters. Owing to our WBC framework (2), the optimal DCA model is:

$$\mathcal{M}(x) = \mathbb{E}(B \circ \phi \circ T(X + k^\star) \mid X = x) = \mathbb{E}(B \circ L \circ S(X + k^\star) \mid X = x).$$

The expectation is taken on blocked bijections B and linear permutations L as mentioned above. The set $\mathscr{B}$ of blocked bijections can be partitioned as:

$$\mathscr{B} = \mathscr{B}_0 \cup \mathscr{B}_1 = \{b \in \mathscr{B} \mid \mathsf{LSB}(b(0)) = 0\} \cup \{b \in \mathscr{B} \mid \mathsf{LSB}(b(0)) = 1\},$$

where LSB is the Least Significant Bit. It is to be noticed that $\mathscr{B}_1 = \mathscr{B}_0 + \texttt{0xffffffff}$, where $\texttt{0xffffffff}$ is the "all one" 32-bit word. Therefore,

$$\begin{aligned}
\mathcal{M}(x) &= \frac{1}{2}\mathbb{E}(B \circ L \circ S(X + k^\star) \mid X = x, \mathsf{MSB}(B(0)) = 0) \\
&+ \frac{1}{2}\mathbb{E}(B \circ L \circ S(X + k^\star) \mid X = x, \mathsf{MSB}(B(0)) = 1) \\
&= \frac{1}{2}\mathbb{E}(B \circ L \circ S(X + k^\star) \mid X = x, \mathsf{MSB}(B(0)) = 0) \\
&+ \frac{1}{2}\mathbb{E}(\texttt{0xffffffff} + B \circ L \circ S(X + k^\star) \mid X = x, \mathsf{MSB}(B(0)) = 0) \\
&= \frac{1}{2}\texttt{0xffffffff} = (0.5, \ldots, 0.5).
\end{aligned}$$

As a consequence, the model $x \mapsto \mathcal{M}(x)$ is constant and therefore does not depend on the key $k^\star$. Therefore attacks of DCA type do fail.

6 Conclusions and perspectives

We introduced a novel distinguisher for AES T-box obfuscated by a random DIBO function. It consists in the counting of zero values in a Walsh-Hadamard spectrum of a guess function.

We started by justifying why such an enumeration makes sense in the context of WBC. Then, we proved that it works as long as at least one restriction (out of four) of the DIBO linear part to external blocked bijections is not invertible. In this respect, we have concluded on the mathematical investigations aiming at accounting for the reason of success of spectral attacks in the field of DIBO; First and foremost, our characterization of weak vs strong linear parts allows to solve the remaining open problem stated in the research of Lee, Jho and Kim in IEEE Access 2020 [LJK20].

In such situation (which happens $> 99\%$ of the time), we showed mathematically that our attack always works. On the opposite, we indicated that when all four linear restrictions are bijective, then our attack (and others, such as "correlation attacks") fails. This situation can happen even without jeopardizing the entropy in the DIBO linear (diffusion) part. Hence, this is a new recommendation to make sure WBC schemes based on DIBO obfuscation are safe. Thus, we have repaired the DIBO obfuscation scheme against spectral attacks.

Our attack efficiency is backed by a thorough analysis of the distinguishing feature of DIBO (when at least one L_i does not have full rank) compared to a random function. As

a perspective, we notice that further investigations could lead to either the identification of more powerful distinguishers, or to establish formally that our distinguisher achieves the best.

Acknowledgments

The authors wish to thank Lucille Tordella on the one hand and Matthieu Desjardins on the other hand for their precious help in the early phases of this work. This work has been partly financed by the BRAINE ("Big data pRocessing and Artificial Intelligence at the Network Edge") H2020 ECSEL European Project, under Grant Agreement N° 876967. The analysis methods presented in this paper have been integrated into Secure-IC's Catalyzr tool [SI21].

References

[BBB+19] Estuardo Alpirez Bock, Joppe W. Bos, Chris Brzuska, Charles Hubain, Wil Michiels, Cristofaro Mune, Eloi Sanfelix Gonzalez, Philippe Teuwen, and Alexander Treff. White-Box Cryptography: Don't Forget About Grey-Box Attacks. *J. Cryptol.*, 32(4):1095–1143, 2019.

[BBMT18] Estuardo Alpirez Bock, Chris Brzuska, Wil Michiels, and Alexander Treff. On the Ineffectiveness of Internal Encodings - Revisiting the DCA Attack on White-Box Cryptography. In Bart Preneel and Frederik Vercauteren, editors, *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*, volume 10892 of *Lecture Notes in Computer Science*, pages 103–120. Springer, 2018.

[BC99] Anna Bernasconi and Bruno Codenotti. Spectral Analysis of Boolean Functions as a Graph Eigenvalue Problem. *IEEE Trans. Computers*, 48(3):345–351, 1999.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. White box cryptography: Another attempt. *IACR Cryptol. ePrint Arch.*, 2006:468, 2006.

[BGEC04] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a White Box AES Implementation. In *Selected Areas in Cryptography*, pages 227–240, 2004.

[BGHR14] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks Will Fall Off – Higher-Order Optimal Distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 344–365. Springer, 2014.

[BHMT16] Joppe W. Bos, Charles Hubain, Wil Michiels, and Philippe Teuwen. Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 215–236. Springer, 2016.

[BRVW19] Andrey Bogdanov, Matthieu Rivain, Philip S. Vejre, and Junwei Wang. Higher-Order DCA against Standard Side-Channel Countermeasures. In Ilia Polian and Marc Stöttinger, editors, *Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3-5, 2019, Proceedings*, volume 11421 of *Lecture Notes in Computer Science*, pages 118–141. Springer, 2019.

[BT20] Estuardo Alpirez Bock and Alexander Treff. Security Assessment of White-Box Design Submissions of the CHES 2017 CTF Challenge. In Guido Marco Bertoni and Francesco Regazzoni, editors, *Constructive Side-Channel Analysis and Secure Design - 11th International Workshop, COSADE 2020, Lugano, Switzerland, April 1-3, 2020, Revised Selected Papers*, volume 12244 of *Lecture Notes in Computer Science*, pages 123–146. Springer, 2020.

[Car20] Claude Carlet. Graph Indicators of Vectorial Functions and Bounds on the Algebraic Degree of Composite Functions. *IEEE Trans. Inf. Theory*, 66(12):7702–7716, 2020.

[Car21] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Monograph in *Cambridge University Press*, January 7 2021. ISBN-10: 1108473806; ISBN-13: 978-1108473804.

[CEJvO02a] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. A White-Box DES Implementation for DRM Applications. In *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002*, volume 2696 of *LNCS*, pages 1–15. Springer, 2002.

[CEJvO02b] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. White-Box Cryptography and an AES Implementation. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *LNCS*, pages 250–270. Springer, 2002.

[CFD+10] Zouha Cherif, Florent Flament, Jean-Luc Danger, Shivam Bhasin, Sylvain Guilley, and Hervé Chabanne. Evaluation of White-Box and Grey-Box Noekeon Implementations in FPGA. In Viktor K. Prasanna, Jürgen Becker, and René Cumplido, editors, *ReConFig*, pages 310–315. IEEE Computer Society, 2010.

[che17] CHES 2017 Capture the Flag Challenge: The `WhibOx` Contest, An ECRYPT White-Box Cryptography Competition, May 15 to September 24 2017. https://whibox-contest.github.io/, accessed on August 3, 2018.

[GMQ07] Louis Goubin, Jean-Michel Masereel, and Michaël Quisquater. Cryptanalysis of White Box DES Implementations. In *Selected Areas in Cryptography, 14th International Workshop, SAC 2007*, volume 4876 of *LNCS*, pages 278–295. Springer, 2007.

[GPRW20] Louis Goubin, Pascal Paillier, Matthieu Rivain, and Junwei Wang. How to reveal the secrets of an obscure white-box implementation. *J. Cryptographic Engineering*, 10(1):49–66, 2020.

[GRW20] Louis Goubin, Matthieu Rivain, and Junwei Wang. Defeating State-of-the-Art White-Box Countermeasures with Advanced Gray-Box Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):454–482, 2020.

[HRG14] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 55–74. Springer, 2014.

[ISO21] ISO/IEC JTC1/SC 27/WG 3 & 2. ISO/IEC DTR 24485.3; Information technology – Security techniques – Security properties, test and evaluation guidance for white box cryptography, June 2021. https://www.iso.org/standard/78890.html.

[Lee18] Seungkwang Lee. A White-Box Cryptographic Implementation for Protecting against Power Analysis. *IEICE Trans. Inf. Syst.*, 101-D(1):249–252, 2018.

[LJK20] Seungkwang Lee, Nam-Su Jho, and Myungchul Kim. On the Linear Transformation in White-Box Cryptography. *IEEE Access*, 8:51684–51691, 2020.

[LK20] Seungkwang Lee and Myungchul Kim. Improvement on a Masked White-Box Cryptographic Implementation. *IEEE Access*, 8:90992–91004, 2020.

[LKK18] Seungkwang Lee, Taesung Kim, and Yousung Kang. A Masked White-Box Cryptographic Implementation for Protecting Against Differential Computation Analysis. *IEEE Trans. Information Forensics and Security*, 13(10):2602–2615, 2018.

[LRM+13] Tancrède Lepoint, Matthieu Rivain, Yoni De Mulder, Peter Roelse, and Bart Preneel. Two Attacks on a White-Box AES Implementation. In Tanja Lange, Kristin E. Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 265–285. Springer, 2013.

[MA20] Houssem Maghrebi and Davide Alessio. Revisiting higher-order computational attacks against white-box implementations. In Steven Furnell, Paolo Mori, Edgar R. Weippl, and Olivier Camp, editors, *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP 2020, Valletta, Malta, February 25-27, 2020*, pages 265–272. SCITEPRESS, 2020.

[MGH08] Wil Michiels, Paul Gorissen, and Henk D. L. Hollmann. Cryptanalysis of a Generic Class of White-Box Implementations. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 414–428. Springer, 2008.

[MRP12] Yoni De Mulder, Peter Roelse, and Bart Preneel. Cryptanalysis of the Xiao-Lai White-Box AES Implementation. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2012.

[MWP10] Yoni De Mulder, Brecht Wyseur, and Bart Preneel. Cryptanalysis of a Perturbated White-Box AES Implementation. In Guang Gong and Kishan Chand

Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 292–310. Springer, 2010.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[Ras20] Sandra Rasoamiaramanana. *Design of white-box encryption schemes for mobile applications security. (Conception de schémas de chiffrement boîte blanche pour la sécurité des applications mobiles)*. PhD thesis, University of Lorraine, Nancy, France, 2020.

[RW19] Matthieu Rivain and Junwei Wang. Analysis and Improvement of Differential Computation Attacks against Internally-Encoded White-Box Implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):225–255, 2019.

[SEL21] Okan Seker, Thomas Eisenbarth, and Maciej Liskiewicz. A White-Box Masking Scheme Resisting Computational and Algebraic Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):61–105, 2021.

[SI21] Secure-IC. Catalyzr tool, 2021. https://cadforassurance.org/tools/software-assurance/catalyzr/, https://www.secure-ic.com/solutions/catalyzr/, accessed on July 2nd 2021.

[SMG16] Pascal Sasdrich, Amir Moradi, and Tim Güneysu. White-Box Cryptography in the Gray Box – A Hardware Implementation and its Side Channels. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 185–203. Springer, 2016.

[TH16] Philippe Teuwen and Charles Hubain. Differential Fault Analysis on White-box AES Implementations, December 19 2016. https://blog.quarkslab.com/differential-fault-analysis-on-white-box-aes-implementations.html, accessed on April 13, 2021.

[whi16] `WhibOx` workshop: White-Box Cryptography and Obfuscation, 14 August 2016. Santa-Barbara, California. https://www.cryptoexperts.com/whibox2016/, accessed on April 13, 2021.

[WMGP07] Brecht Wyseur, Wil Michiels, Paul Gorissen, and Bart Preneel. Cryptanalysis of White-Box DES Implementations with Arbitrary External Encodings. In *Selected Areas in Cryptography, 14th International Workshop, SAC 2007*, volume 4876 of *LNCS*, pages 264–277. Springer, 2007.

[WO11] Carolyn Whitnall and Elisabeth Oswald. A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. *J. Cryptographic Engineering*, 1(2):145–160, 2011.

[XL09] Yaying Xiao and Xuejia Lai. A Secure Implementation of White-Box AES. In *2009 2nd International Conference on Computer Science and its Applications*, pages 1–6, 2009. Jeju, South Korea. DOI: 10.1109/CSA.2009.5404239.

A Some examples of attacks

In this appendix, we will experimental show the impact of the ranks of L_i on the efficiency of the distinguishers.

A.1 Preliminaries

We recall that $L = \phi \circ \mu$ (see (5)), where ϕ is a secret permutation of $\mathbb{F}_2^{32}$ and $\mu : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ is the diffusion function `MixColumns`. Notice that our attack would work as well if μ was any linear $\mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ onto.

Let us detail how we choose random permutations ϕ such that each L_i has a given rank. It is possible to draw random ϕ at random, and discarding it if the ranks of L_i do not correspond to the expected ones. However, in practice this approach is very inefficient. For this reason, we opted to choose L_i and deduce ϕ from them.

- Building a random L_i of rank r can be done as follows. Three random matrices M_L, M_C, M_R containing bits, of size respectively 8×8, $8 \times r$, and $r \times 8$, are drawn. While their ranks are not respectively 8, r and r, they are drawn again. (In practice, 3 or 4 iterations are sufficient.) Then, L_i is chosen as the linear function of matrix $M_L M_C M_R$.

- The deduction of ϕ from the four L_i ($1 \leq i \leq 4$) is obtained from Lemma 3.

Lemma 3. *Let $L : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ a linear injection. Then, there exists a linear bijection ϕ such that $L = \phi \circ \mu$, where μ is the `MixColumns` application $x : \mathbb{F}_{2^8} \mapsto (02x, x, x, 03x) \in \mathbb{F}_{2^8}^4$ (exploded on bits).*

Proof. Let $L : \mathbb{F}_2^8 \to \mathbb{F}_2^{32}$ a linear injection. The dimension of $\mathsf{Im}(L)$ is 8. As μ (defined in the body of the lemma) is also injective, $\mathsf{Im}(\mu)$ has also dimension 8. Let ϕ_0 a bijection from $\mathsf{Im}(\mu)$ to $\mathsf{Im}(L)$. The space vector $\mathsf{Im}(\mu)$ (resp. $\mathsf{Im}(L)$) can be completed in a space vector E (resp. F) of dimension 24. Let ψ a linear bijection from E to F. Then, ϕ can be chosen as $\phi(y) = \phi_0(y) + \psi(y)$. □

In the next subsection A.2, we illustrate how our distinguisher works (in terms of distinguishing margin).

A.2 Some examples of attacks

We contrast in this section the success rate of our distinguisher (recall Def. 5) and a related one which would consist in selecting coordinates instead of values of v in E (defined in (7)). The second variant is closer to that of Sasdrich et al., in that it has a lesser complexity.

A.2.1 Outcome of attacks on DIBO without constraints on ϕ

In order to illustrate the performance of the distinguishers, we repeat multiple experiments. They all consist in breaking the DIBO obfuscation of $x \mapsto T(x + k^\star)$, where the correct key is $k^\star = \texttt{0x80}$. The random parts involved are:

- ϕ, a random bijective linear function, drawn randomly by selecting 32×32 random i.i.d. bits forming a matrix, and repeating the process until the matrix is invertible.

- B, the concatenation of 4 bijections $\mathbb{F}_2^8 \to \mathbb{F}_2^8$, each of which being obtained by permuting randomly the set $\{\texttt{0x00}, \texttt{01}, \ldots, \texttt{0xff}\}$.

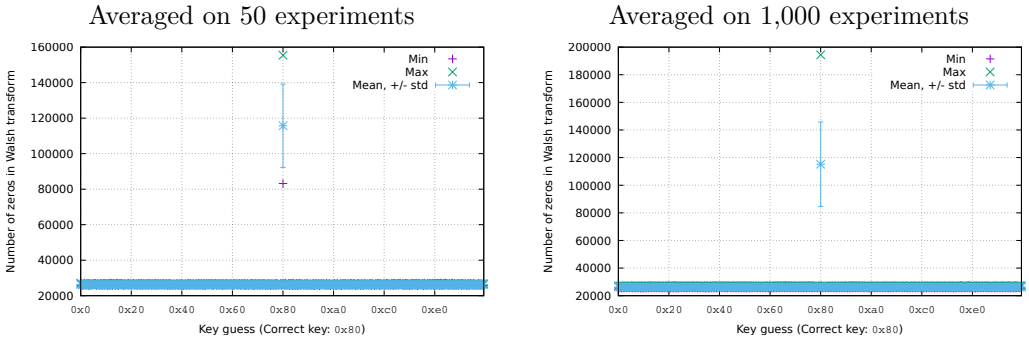

Figure 4: Number of zeros of Walsh spectrum of A_k when sampling Walsh transform by byte component (correct key $k^\star = \texttt{0x80}$ is in the middle).

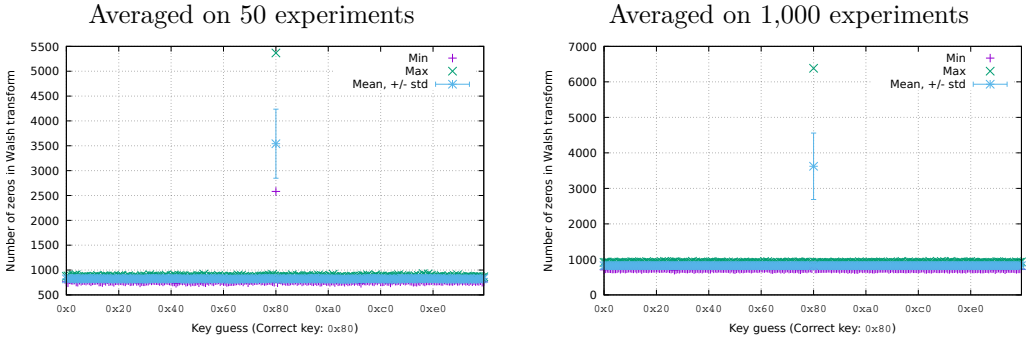

Figure 5: Number of zeros of Walsh spectrum of A_k when sampling Walsh transform by coordinates (correct key $k^\star = \texttt{0x80}$ is in the middle).

Namely, we represent in a first graph:

$$D_{\mathsf{component}}(c) = \sum_{j=1}^{4} \#\{W_{A_k}(u, v) = 0 \mid u \in \mathbb{F}_2^8, v \in E\}, \tag{13}$$

as a function of $c = k + k^\star$. It is represented in Fig. 4. For a low number of experiments (namely 50 different WBC T-boxes are generated), our attack always works. However, as per (12), after $1,000$ experiments, circa 993 WBC T-boxes are attackable whereas circa 7 are not. This can be seen in the right hand side graph of Fig. 4, where the minimum value of our distinguisher for the correct key guess $k = k^\star$ disappears amongst the values for incorrect key guesses $k \neq k^\star$.

The second graph represents in Fig. 5 the following distinguisher:

$$\begin{aligned} D_{\mathsf{coordinate}}(c) &= \#\{W_{A_k}(u, v) = 0 \mid u \in \mathbb{F}_2^8, w_H(v) = 1\} \\ &= \sum_{i=1}^{32} \#\{W_{(A_k)_i}(u, v) = 0 \mid u \in \mathbb{F}_2^8\}, \end{aligned} \tag{14}$$

as a function of $c = k + k^\star$. It samples the number of values in the Walsh spectrum only on the coordinates of A_k.

It can be seen that sometimes, the attack fails (not on the 50 first attacks case, but on the 1,000 ones, sometimes, the distinguisher for the good has a number of zero which cannot

Table 1: Relative distinguishing margins ,as per (15), extracted from distinguisher computations of Fig. 4 and 5

	50 experiments	1,000 experiments
$D_{\mathsf{component}}$ (13) (favorite, our Def. 5)	60.1	46.1
$D_{\mathsf{coordinate}}$ (14) (weaker variant)	53.5	43.1

be distinguished from that of the incorrect keys.) This is inline with the computation of the proportion of vulnerable DIBOs shown in (12).

Besides, one can notice that our distinguisher (recall Def. 5, or (13)) and the "per-coordinate" sibling perform qualitatively the same. However, the version of Def. 5 is more acute in that the contrast between the peak for the correct key $k^\star = \texttt{0x80}$ and the incorrect key guesses ($k \neq k^\star$) is larger. Such contrast can be quantified with a notion of *relative distinguishing margin* (RDM), defined in [WO11, §3]. For a given distinguisher D, it is equal to:

$$\mathrm{RDM}(D) = \frac{D(k^\star) - \max_{k \neq k^\star} D(k)}{\sqrt{\mathrm{Var}(\{D(k) \mid k \in \mathbb{F}_2^8\})}}. \tag{15}$$

This metric is computed in Tab. 1. It shows quantitatively that our spectral distinguisher $D_{\mathsf{component}}$ distinguishes more accurately than the version $D_{\mathsf{coordinate}}$ operating coordinate-wise.

A.2.2 Outcome of attacks on DIBO with constraints on ϕ

We now analyse our repair of DIBO (forcing that $\forall i,\ 1 \leq i \leq 4, \mathrm{rank}(L_i) = 8$), and the strength of our spectral attack when this is not the case.

We therefore classify the attack results according to the dimension of the kernel of the linear applications L_i. Namely, we illustrate 7 situations, using $D_{\mathsf{coordinate}}$ (since its discriminating power is similar to that of $D_{\mathsf{component}}$):

- Case '0,0,0,0', where $\forall i \in \{1,2,3,4\}$, $\ker(L_i) = \{0\}$ (attack failure)
- Case '0,0,0,1', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{0,0,0,1\}$ (attack success)
- Case '0,0,1,1', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{0,0,1,1\}$ (attack success, with larger margin)
- Case '0,1,1,1', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{0,1,1,1\}$ (attack success, with still larger margin)
- Case '1,1,1,1', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{1,1,1,1\}$ (attack success, with still even larger margin)
- Case '0,0,0,2', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{0,0,0,2\}$ (attack success)
- Case '0,0,0,3', where $\{\mathrm{rank}(L_i), 1 \leq i \leq 4\} = \{0,0,0,3\}$ (attack success)

The results are displayed in Tab. 2. We notice that when L_i all have full rank, then the attack always fails. This confirms the theoretical analysis we conducted in Sec. 4.

Besides, one can see that having at least one kernel (out of the four) of L_i which has a non-zero dimension allows one to distinguish the correct key from the incorrect keys. Now, one can also see that it is more important to have more trivial kernels than having kernels of larger dimensionality. Indeed, the RDM between correct key and rivals is getting linearly larger when number of non-trivial kernels increases (from 1 to 4), whereas the number of zeros in the Walsh transform only increases marginally when the dimensionality

Table 2: Attack statistics based on 1,000 attacks, for L matching the properties expressed above (correct key $k^\star = \texttt{0x80}$ is in the middle).

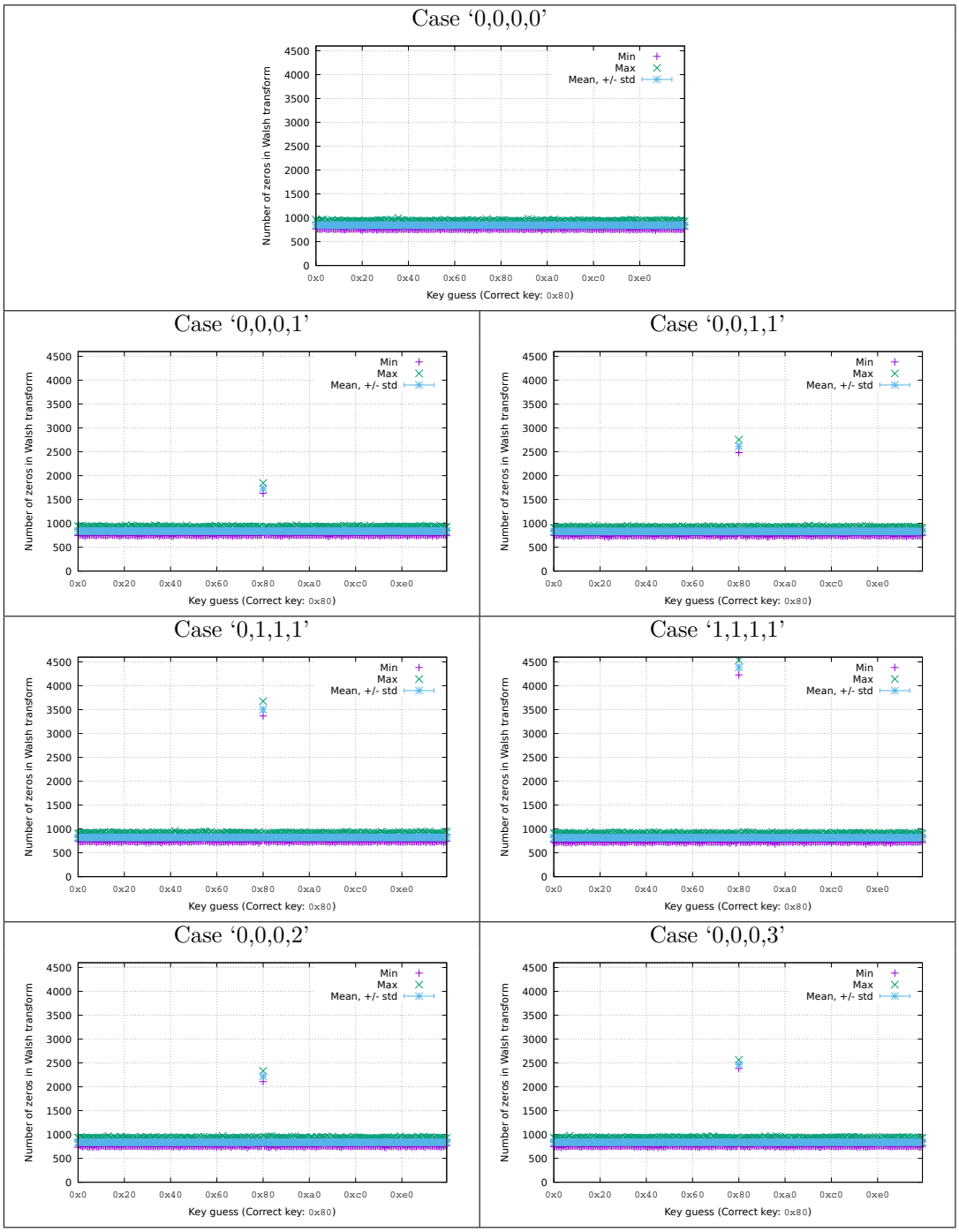

Table 3: Statistics on the number of zeros in the Walsh transform. We recall that "Case" is the list of the four kernel dimensions, sorted in increasing order.

Case	Mean	Median
Case '0,0,0,0'	844	843
Case '0,0,0,1'	1728	1727
Case '0,0,1,1'	2616	2615
Case '0,1,1,1'	3498	3498
Case '1,1,1,1'	4384	4385
Case '0,0,0,2'	2218	2218
Case '0,0,0,3'	2461	2460

of the kernel is growing from 1 to 2, and from 2 to 3. Precisely, some statistics about the count of zeros in the Walsh transform is given in Tab. 3.