IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2024, No. 3, pp. 452–471. DOI:10.46586/tches.v2024.i3.452-471

On the (Im)possibility of Preventing Differential Computation Analysis with Internal Encodings

Laurent Castelnovi[1] and Agathe Houzelot[1,2]

[1] IDEMIA, Cryptography & Security Labs, Pessac, France
[2] LaBRI, CNRS, Université de Bordeaux, Bordeaux, France
{firstname.lastname}@idemia.com

Abstract. White-box cryptography aims at protecting implementations of cryptographic algorithms against a very powerful attacker who controls the execution environment. The first defensive brick traditionally embedded in such implementations consists of encodings, which are bijections supposed to conceal sensitive data manipulated by the white-box. Several previous works have sought to evaluate the relevance of encodings to protect white-box implementations against grey-box attacks such as Differential Computation Analysis (DCA). However, these works have been either probabilistic or partial in nature. In particular, while they showed that DCA succeeds with high probability against AES white-box implementations protected by random encodings, they did not refute the existence of a particular class of encodings that could prevent the attack. One could thus wonder if carefully crafting specific encodings instead of drawing random bijections could be a solution.
This article bridges the gap between preceding research efforts and investigates this question. We first focus on the protection of the S-box output and we show that no 4-bit encoding can actually protect this sensitive value against side-channel attacks. We then argue that the use of random 8-bit encodings is both necessary and sufficient, but that this assertion holds exclusively for the S-box output. Indeed, while we define a class of 8-bit encodings that actually prevents a classical DCA targeting the MixColumns output, we also explain how to adapt this attack and exploit the correlation traces in order to defeat even these specific encodings. Our work thus rules out the existence of a set of practical encodings that could be used to protect an AES white-box implementation against DCA-like attacks.

Keywords: White-Box Cryptography · Encodings · Side-Channel Attacks · Differential Computation Analysis · AES

1 Introduction

The goal of white-box cryptography is to protect the secret keys embedded in software implementations of cryptographic algorithms that are executed in untrusted environments. In their seminal works [CEJvO03a, CEJVO03b], Chow et al. presented the two first white-box implementations of DES and AES respectively. For both these implementations, the main idea is to split the algorithm into a network of small look-up tables – small in the sense that they take a restricted number of bits as input. To hide the potentially key-dependant data contained in these tables, Chow et al. introduced the notion of *encoding*, which is a random bijection applied to the output of a table, later cancelled by the inverse bijection applied to the input of the next table in the network. The input encoding of the first table and the output encoding of the last one are called *external encodings*, as opposed to all other ones refered to as *internal encodings*. This sementical distinction comes from the fact that, contrary to internal encodings designed to cancel each other,

Received: 2024-01-15 Accepted: 2024-03-15 Published: 2024-07-18

external encodings cannot be removed without revealing them. Therefore, the white-box designer is constrained either to implement a non-standard algorithm, or to set the external encodings to the identity, which is the mandatory choice in most situations.

Unfortunately, encodings have been proven to be insufficient to protect white-box implementations. Indeed, Chow et al.'s proposals were broken only a few years after their publication by efficient algebraic attacks [BGEC05, WMGP07]. After that, several authors tried to find secure AES white-box designs [XL09, Kar11, BCD06], most of them keeping the idea of encodings but changing their size or adding other layers of security. But again, these designs were all broken by algebraic attacks [DMWP10, LRDM+14, DMRP13].

Despite the efficiency of these attacks, encodings remain of common use in white-box designs. Indeed, in such a context where no theoretically secure implementation of any standard block cipher is known, public white-box designs like those submitted to WhibOx contests[1] essentially rely on encoding techniques and a layer of obfuscation to defend their embedded key. Obfuscation makes algebraic attacks practically difficult as they require the adversary to recover some intermediate variables that obfuscation makes hard to identify without a painful reverse-engineering work. To free the attacker from such an arduous task, Bos et al. [BHMT16] proposed to adapt the well-known grey-box Differential Power Analysis of Kocher et al. [KJJ99] to the white-box model. Their so-called Differential Computation Analysis (DCA) consists, as its grey-box counterpart, in exploiting execution traces and recovering secret information through the use of statistical tests. The only difference between the two attacks lies in the nature of the traces. In the grey-box model, the traces render the intermediate variables of the algorithm through noisy side-channels such as execution time or power consumption. In the white-box model, instead of relying on noisy information indirectly linked to the computations, the adversary may record and analyze the values that are read or written by the computing device. This way, the traces are completely noise-free and the attack is even more devastating than in the grey-box model. Furthermore, the usual countermeasure against side-channel attacks that consists in masking the sensitive variables can be costly and presents challenges in the white-box model. In particular, it implies to draw some random values at run-time that must be kept secret and unaltered, which is not an easy task for the white-box designer.

In this context, encodings appear to be an attractive alternative solution. If, on the one hand, external encodings defeat DCA by impeding the attacker to predict intermediate key-dependant values, they are impossible to use in many use-cases because they imply a modification of the algorithm and thus make the scheme not standard anymore. On the other hand, intuitively, internal encodings may counter DCA because they lower the correlation between the predicted variables and the trace samples. However, this intuition is mitigated by many successful DCA led notably in [BHMT16] against white-box instances using internal encodings. Sasdrich et al. were the first to try to explain this phenomenon in [SMG16]. They made several experiments and argued that the success of DCA against encoded implementations is directly linked to the presence of high values in the Walsh spectrum of many encodings. A few years later, Alpirez Bock et al. found some necessary conditions for the encodings to prevent a DCA targeting an AES S-box output [ABBMT18]. Rivain and Wang then generalised their results in [RW19] and managed to compute the probability of success of DCA depending on some parameters. They proved that picking encodings at random cannot be a good strategy to protect white-box implementations against side-channel attacks, but did not discuss the idea of carefully crafting them. Since the success probabilities that they derived are never equal to 1, Rivain and Wang did not rule out the possibility of the existence of a particular class of encodings that effectively prevents DCA.

[1] https://whibox.io/contests/

Our Contributions. In this paper, we look for such DCA-resistant encodings for white-box implementations of AES. We investigate encodings which length is a power of 2 in order to avoid practical implementation difficulties. We exclude 2-bit bijections as there are only twenty-four of them, which is not enough to prevent exhaustive search on sensitive values. Moreover, the memory cost of encoding is prohibitive from 16 bits onwards. We thus focus on 4-bit and 8-bit encodings.

We show, based on Sasdrich et al.'s work, that the S-box output can be protected against DCA by random 8-bit encodings only. In particular, we argue that no 4-bit encoding can preserve the S-box output from DCA-like attacks. Regarding MixColumns, although it has been demonstrated that random 8-bit encodings are ineffective, we define a class of 8-bit encodings that actually prevent DCA. However, we also explain how to adapt the traditional attack in order to defeat even these specific encodings. We exhibit a difference of behaviour of the correlation coefficient depending on the key hypothesis, allowing the attacker to identify the correct key guess, regardless of the 8-bit encodings applied. Therefore, we show that MixColumns, and thus AES, cannot be protected from DCA by encodings of practical size.

Organisation. The rest of the paper is organised as follows: in Sect. 2, we introduce some notations and we recall the concepts that we will need in the subsequent sections. In Sect. 3, we expose our results about the protection of the S-box output. The case of MixColumns is discussed in Sect. 4. We conclude our work in Sect. 5.

2 Preliminaries

Throughout the paper, we use the notations described in Table 1.

Table 1: Notations.

$\mathrm{HW}(x)$	The Hamming weight of x
$\langle x, y \rangle$	The scalar product between x and y
$\mathrm{S}(x)$	The result of the application of the AES S-box on x
$\#A$	The cardinality of the set A
$\mathcal{U}(A)$	The uniform distribution over the set A
F_i	A coordinate function of F

2.1 Chow et al.'s White-Box Implementation of AES

AES. The Advanced Encryption Standard (AES) [DR99], was selected by the NIST in 2000 as the new American standard for symmetric encryption. This block cipher takes 128-bit inputs and can be configured for several key length. In this paper, we will focus on AES-128, which is the version with a 128-bit key. The encryption algorithm then consists of 10 rounds during which four operations are successively applied on a 16-byte state.

1. **AddRoundKey:** A 128-bit round key derived from the master key is added to the state using an exclusive-or operation.
2. **SubBytes:** A non-linear bijection called S-box is applied on each byte of the state.
3. **ShiftRows:** This operation permutes the indexes of the state bytes.
4. **MixColumns:** The state is divided into four 4-byte vectors that are each multiplied by a 4×4 matrix MC.

In the last round, the MixColumns operation is replaced by an exclusive-or with a post-whitening key.

Chow et al.'s White-box. The first white-box implementation of AES was published by Chow et al. in 2002 [CEJVO03b]. Their main idea was to implement all the operations in the form of a network of look-up tables, the content of which is hidden by random permutations called encodings.

Definition 1 (Encoding). Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Let $E^{(0)}$ and $E^{(1)}$ be bijections over $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ respectively. The function $\overline{F} = E^{(1)} \circ F \circ E^{(0)}$ is called an encoded function of F, and $E^{(0)}$ and $E^{(1)}$ are called the input and output encodings respectively.

Chow et al's white-box implementation of AES is inspired from the T-tables suggested by Daemen and Rijmen in their AES proposal [DR99]. A very good simplified but complete description of this white-box implementation is given by Muir in [Mui13]. In a nutshell, the white-box proposed by Chow et al. implements AES in the following way. Let $(k_{r,0}, k_{r,1}, \ldots, k_{r,15})$ be the r^{th} round key. The r^{th} round of AES, $1 \leq r \leq 9$, is decomposed into:

1. Sixteen tables $T_{r,i}$ that take as input a byte x of the state and return an encoded S-box output.
2. Sixteen tables $M_{r,i}$ that take as input an encoded byte from the previous step and return an encoded version of a column vector of MC multiplied by the decoded input byte,
3. A number of XOR-tables that take two encoded inputs and return the encoded bitwise exclusive-or of the decoded inputs.

These XOR-tables are used to complete the multiplication of the input state of MixColumns by MC. Thus, they receive as input the 32-bit output of the tables $M_{r,i}$. Since it is not possible to have 64-bit inputs for XOR-tables, they actually receive two 4-bit or 8-bit inputs. This implies that the output encoding of $M_{r,i}$ has a special form: it is either eight concatenated 4-bit encodings, or four concatenated 8-bit encodings.

2.2 Differential Computation Analysis

In 2016, Bos et al. [BHMT16] showed that the side-channel attacks, although invented for the grey-box context, are even more devastating in the white-box model. The main idea of their *Differential Computation Analysis* (DCA) remains unchanged: secret values are extracted from leakage traces with the help of statistical tools. The only difference consists of the nature of the traces: while in the grey-box context the adversary would measure side-channel information that is only indirectly linked to secret values, as power consumption for example, a white-box attacker can obtain noise-free software traces by recording the accessed memory addresses or even the values of intermediate variables.

In more details, DCA consists of the following steps:

1. Select a sensitive variable V_k that depends on a few key bits only. For example for AES, one could select $V_k = \mathrm{S}(x \oplus k)$ for any byte x of the plaintext.
2. Acquire a set T of computation traces of length t.
3. For all key guess $\hat{k}$ and all point index $0 \leq i < t$, compute $\rho_{\hat{k},i} = \Delta(V_{\hat{k}}, T_i)$ with Δ an arbitrary score function. Note that in the following, we will often drop the subscript and denote by $\rho^{\times}$ (resp. ρ^{*}) the score for a bad (resp. correct) key hypothesis.
4. Validate the key hypothesis that maximises $\rho_{\hat{k},i}$.

In our work, like in many others [BU18, RW19, SMG16, HBG23], we choose for Δ Pearson's correlation coefficient in order to measure the correlation between trace points and sensitive values.

Definition 2 (**Correlation coefficient**)**.** Given two random variables X and Y, Pearson's correlation coefficient is defined as

$$\mathrm{Cor}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \ ,$$

with $\mathrm{Cov}(X,Y)$ denoting the covariance between X and Y and σ_X (resp. σ_Y) being the standard deviation of X (resp. Y).

While not specifically designed to protect an implementation against side-channel attacks, the encodings intuitively hinder the attacker since they can lower the correlation between the trace samples and the hypothesised sensitive variable. Nevertheless, many encoded implementations of AES have been broken by DCA [BHMT16]. The reason behind the success of these attacks was studied by Sasdrich et al. [SMG16] and Rivain and Wang [RW19] with the help of tools from the theory of Boolean functions.

2.3 Boolean Functions

First, let us recall a few definitions. Let $n, m \geq 1$ be two integers. A function f from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is called a *Boolean function* while a function F from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called a *vectorial function*. If $F(x) = (F_0(x), F_1(x), \ldots, F_{m-1}(x))$, the Boolean functions $F_0, F_1, \ldots, F_{m-1}$ are often referred to as the *coordinate functions* of F. The function F is said to be *balanced* if, for all $y \in \mathbb{F}_2^m$, $\#\{x \in \mathbb{F}_2^n \mid F(x) = y\} = 2^{n-m}$.

In the rest of this paper, we will use the notions of Walsh transform and Walsh spectrum of Boolean (resp. vectorial) functions. We thus recall the following definitions.

Definition 3 (**Walsh transform, Walsh spectrum**)**.**

- The Walsh transform of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as:

$$\begin{aligned} W_f : \mathbb{F}_2^n &\to \mathbb{Z} \\ u &\mapsto \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\langle u,x \rangle} \ . \end{aligned}$$

- The Walsh transform of a vectorial function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as:

$$\begin{aligned} W_F : \mathbb{F}_2^n \times \mathbb{F}_2^m &\to \mathbb{Z} \\ (u,v) &\mapsto \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle v,F(x) \rangle + \langle u,x \rangle} \ . \end{aligned}$$

- The Walsh spectrum of a vectorial function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is the set of all the values that $W_F(u,v)$ can take:

$$\mathcal{W}(F) = \{W_F(u,v) \mid (u,v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m\} \ .$$

In this paper, we are only interested in the case where the inputs u and v are of Hamming weight 1. In this case, the value $W_F(u,v)$ can be seen as a measure of the correlation between one input bit and one output bit of F. We will denote by $\mathcal{W}_1(F)$ the subset of the Walsh spectrum restricted to the values u and v of Hamming weight 1:

$$\mathcal{W}_1(F) = \{W_F(u,v) \mid (u,v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m, \mathrm{HW}(u) = \mathrm{HW}(v) = 1\} \ .$$

In the following, we will focus on the Walsh spectrum of encodings. Since the latter are bijections, their coordinate functions are balanced Boolean functions and their Walsh transforms have the following property.

Proposition 1. *Let $n \geq 2$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a balanced Boolean function. For all $u \in \mathbb{F}_2^n$ such that* $\mathrm{HW}(u) = 1$*, $W_f(u)$ is a multiple of* 4.

Proof. Let $n \geq 2$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a balanced Boolean function. Let $u \in \mathbb{F}_2^n$ with $\mathrm{HW}(u) = 1$. Let us denote by A the biggest set $A \subseteq \mathbb{F}_2^n$ such that $\sum_{x \in A}(-1)^{f(x)+\langle x,u \rangle} = 0$ and by B the set $B = \mathbb{F}_2^n \setminus A$. We then have:

$$W_f(u) = \sum_{x \in B}(-1)^{f(x)+\langle x,u \rangle} \quad .$$

The value $(-1)^{f(x)+\langle x,u \rangle}$ is constant over B since otherwise we could add at least two values in A. Therefore, proving that the cardinality of B is a multiple of 4 would conclude this proof. Since $\#B = 2^n - \#A$ and $n \geq 2$, it is also sufficient to prove that the cardinality of A is a multiple of 4.

If $f = x \mapsto \langle x, u \rangle$ or $f = x \mapsto \neg\langle x, u \rangle$, then we have $\#A = 0$, which concludes the proof. Otherwise, since f is balanced, there exist x_1, x_2, x_3 and x_4 in $\mathbb{F}_2^n$ such that:

$$\begin{array}{llll} \langle x_1, u \rangle = 0 & \text{and} & f(x_1) = 1, \\ \langle x_2, u \rangle = 0 & \text{and} & f(x_2) = 0, \\ \langle x_3, u \rangle = 1 & \text{and} & f(x_3) = 1, \\ \langle x_4, u \rangle = 1 & \text{and} & f(x_4) = 0. \end{array}$$

These four values x_1, x_2, x_3 and x_4 are different, so by definition of A, they belong to A. Now let $\mathcal{E} = \mathbb{F}_2^n \setminus \{x_1, x_2, x_3, x_4\}$. If $f(x) = \langle x, u \rangle$ or $f(x) = \neg\langle x, u \rangle$ for all x in $\mathcal{E}$, then $\#A = 4$. Otherwise, the argument above can be applied on $\mathcal{E}$ and the cardinality of A is increased by 4. Recursively, we get that $\#A$ is always a multiple of 4, then so are $\#B$ and $W_f(u)$. □

2.4 Previous Works

Sasdrich et al. were the first authors to try to understand the reason behind the success of side-channel attacks on encoded implementations [SMG16]. They made some experiments on an implementation using randomly generated 4-bit encodings as proposed by Chow et al. and successfully retrieved the secret key by targeting the outputs of the first-round S-boxes. They experimentally observed that this is linked to the high values in the Walsh spectrum of the encodings that were used but left the construction of secure ones as future work.

Alpirez Bock et al. also studied the link between internal encodings and the success of DCA [ABBMT18] even if they did not use the Walsh transform to do so. They showed that with a linear 8-bit encoding, a DCA targeting the output of a first-round S-box succeeds if and only if at least one of the rows of the matrix generating the encoding has a Hamming weight equal to 1. Alpirez Bock et al. also showed that with non-linear 4-bit encodings, the value of the correlation coefficient obtained for the right key guess is always a multiple of 1/4, which leads to a successful DCA in most cases.

One year later, Rivain and Wang took over the studies of Alpirez Bock et al. and generalised their results [RW19]. Placing themselves in an idealised model, they managed to compute the formula for the probability of success of a DCA depending on different parameters:

- The number of plaintext bits involved in the computation of the sensitive variable
- The bit-size of the encodings,
- The size $\#K$ of the set of key hypotheses.

In the particular case of 4-bit encodings, their analysis is consistent with the one of Alpirez Bock et al. Contrary to what was believed at the time, Rivain and Wang also showed that it is actually possible to break an implementation using random non-linear 8-bit encodings by targeting the output of the MixColumns instead of the one of the S-box. If we focus on AES white-boxes protected by 4-bit or 8-bit encodings, the conclusion of their work is roughly the following:

- An attack targeting an S-box output will succeed with very high probability if the encodings are only applied on nibbles while it will succeed with very low probability in the presence of 8-bit encodings.
- An attack targeting the output of the MixColumns will succeed with very high probability even with 8-bit encodings.

Rivain and Wang did not explain in which case the attacks were successful or not and only gave success probabilities in the presence of random encodings. In other words, they did not point out a potential common characteristic of the encodings that make DCA fail. The question that arises from their work is thus the one of the existence of a particular class of encodings that always prevent the attack. In the rest of the paper, we study the possibility of crafting encodings with a specific property instead of randomly drawing them.

3 Protecting the S-box Output

In [RW19], Rivain and Wang showed that in an idealised model, a DCA targeting one bit of a first round's S-box output protected by 4-bit encodings succeeds with probability close to 0.926, while an implementation using 8-bit encodings is broken with probability 0.0025 only. In this section, we will show that it is actually impossible to find 4-bit encodings that are efficient against side-channel attacks. We will also demonstrate that selecting 8-bit encodings with specific properties in order to further reduce the DCA success probability of 0.0025 is counterproductive. We will indeed argue that randomly drawing 8-bit encodings is both necessary and sufficient to prevent DCA.

3.1 Why Not to Use 4-bit Encodings

Since they are less expensive than their 8-bit counterpart in terms of memory, 4-bit encodings are often used in the literature. In the implementation described in Sect. 2.1, if the outputs of $M_{r,i}$ are encoded by 4-bit encodings, the total memory space per round of XOR-tables amounts to $8 \cdot 2^8 \cdot 4 = 2^{13}$ bits, while with 8-bit encodings it reaches $4 \cdot 2^{16} \cdot 8 = 2^{21}$ bits. Unfortunately, we will show in this section that there exists no family of 4-bit encodings that could be used to prevent side-channel attacks targeting the first round's S-box outputs.

We know from [SMG16] that the success of DCA is directly linked to the Walsh spectrum of the encoding that hides the targeted value. Our study consists in computing all the possible values for the Walsh transforms of 4-bit encodings and showing that the implementation can be broken by side-channel attacks for all of them. Let us start with a proposition.

Proposition 2. *Let $E^{(1)}, E^{(2)} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be two bijections with $n \geq 2$. Let $f = E^{(1)} \| E^{(2)}$ and $\{f_i\}_{0 \leq i < 2n}$ be the coordinate functions of f. For all $0 \leq i < 2n$ and for all $\omega \in \mathbb{F}_2^{2n}$ with $\mathrm{HW}(\omega) = 1$, $W_{f_i}(\omega)$ is a multiple of 2^{n+2}.*

Proof. For the sake of clarity, we will assimilate $\mathbb{F}_2^n$ to the set of integers $[\![0, 2^n - 1]\!]$ using the bijection $(x_0, x_1, \ldots, x_{m-1}) \mapsto x_0 + 2x_1 + \cdots + 2^{m-1}x_{m-1}$.

We will only prove the proposition for $0 \leq i < n$, but the demonstration is similar for $n \leq i < 2n$. Let $\mathcal{S}$ be the set:

$$\mathcal{S} = \{\{u + 2^n v \mid v \in [\![0, 2^n - 1]\!]\} \mid u \in [\![0, 2^n - 1]\!]\}.$$

By construction, $\mathcal{S}$ forms a partition of $[\![0, 2^{2n} - 1]\!]$, so for all $\omega \in \mathbb{F}_2^{2n}$, we have

$$\begin{aligned} W_{f_i}(\omega) &= \sum_{x \in \mathbb{F}_2^{2n}} (-1)^{f_i(x) + \langle x, \omega \rangle} \\ &= \sum_{S \in \mathcal{S}} \sum_{x \in S} (-1)^{f_i(x) + \langle x, \omega \rangle} \ . \end{aligned}$$

When $0 \leq i < n$, f_i is constant over S for all $S \in \mathcal{S}$ so for any $x_S \in S$, we get:

$$W_{f_i}(\omega) = \sum_{S \in \mathcal{S}} (-1)^{f_i(x_S)} \sum_{x \in S} (-1)^{\langle x, \omega \rangle} \ . \tag{1}$$

If $\omega < 2^n$ and $\mathrm{HW}(\omega) = 1$, then the function $g : S \to \mathbb{F}_2$ such that $g(x) = \langle x, \omega \rangle$ is balanced for all $S \in \mathcal{S}$, so $\sum_{x \in S} (-1)^{\langle x, \omega \rangle} = 0$ and $W_{f_i}(\omega) = 0$, which is obviously a multiple of 2^{n+2}.

If $\omega \geq 2^n$ and $\mathrm{HW}(\omega) = 1$, the function $g : S \to \mathbb{F}_2$ such that $g(x) = \langle x, \omega \rangle$ is constant for all $S \in \mathcal{S}$, so:

$$\sum_{x \in S} (-1)^{\langle x, \omega \rangle} = \#S \cdot (-1)^{\langle x_S, \omega \rangle} \ .$$

By construction, all the sets in $\mathcal{S}$ share the same cardinality, hence (1) becomes:

$$W_{f_i}(\omega) = 2^n \sum_{S \in \mathcal{S}} (-1)^{f_i(x_S) + \langle x_S, \omega \rangle} \ . \tag{2}$$

Furthermore, the integers $0, 1, \ldots, 2^{n-1}$ are all in different sets. Then, by choosing them as representatives of their respective set, (2) can be re-written as:

$$W_{f_i}(\omega) = 2^n \sum_{0 \leq x_s < 2^n} (-1)^{f_i(x_s) + \langle x_S, \omega \rangle} \ .$$

The above sum corresponds to the Walsh transform of a coordinate function of $E^{(2)}$. Note that this function is balanced on $\mathbb{F}_2^n$ because $E^{(2)}$ is a bijection, so Proposition 1 applies and its Walsh transform is a multiple of 4. Hence, $W_{f_i}(\omega)$ is a multiple of 2^{n+2}. □

Proposition 2 states that the Walsh spectrum of a function corresponding to the concatenation of two 4-bit encodings only contains multiples of 64. This gives information on the possible correlation scores under the correct key guess for a DCA targeting one bit of an S-box output protected by 4-bit encodings through the following proposition.

Proposition 3. *Let $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an encoding. We have:*

$$\mathrm{Cor}(E_i(x), x_j) = \frac{1}{2^n} \cdot W_{E_i}(2^j)$$

for each bit x_j of the random variable $x \sim \mathcal{U}(\mathbb{F}_2^n)$ and each coordinate function E_i of E, with $0 \leq i, j < n$.

Proof. Let $f_1, f_2 : \mathbb{F}_2^n \to \mathbb{F}_2$ be two balanced Boolean functions and $x \sim \mathcal{U}(\mathbb{F}_2^n)$. According to [RW19, Sect. 2.4], we have:

$$\mathrm{Cor}(f_1(x), f_2(x)) = \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} (-1)^{f_1(u) \oplus f_2(u)} \ . \tag{3}$$

Since E is a bijection, all its coordinate functions are balanced. Furthermore, the bit x_j of x can be written as the output of the function $g_j(x) = \langle x, 2^j \rangle$ which is also balanced. Hence, we get:

$$\begin{aligned}\mathrm{Cor}(E_i(x), x_j) &= \mathrm{Cor}(E_i(x), g_j(x)) \\ &= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} (-1)^{E_i(u) + \langle u, 2^j \rangle} \\ &= \frac{1}{2^n} \cdot W_{E_i}(2^j) \ .\end{aligned}$$

□

Therefore, if one applies 4-bit encodings on the S-Box outputs, the correlation scores for a DCA will always be multiples of $64/256 = 1/4$. Note that this was also shown in a different manner by Alpirez Bock et al. in [ABBMT18].

Proposition 3 implies that, in order to resist DCA, one should select encodings with only small (absolute) values in their Walsh spectrum. One could then be tempted to craft encodings E such that $\mathcal{W}_1(E) = \{0\}$. This means that, for any coordinate function E_i of E and any bit x_j of the targeted intermediate value $x = \mathrm{S}(m \oplus k)$, $\mathrm{Cor}(E_i(x), x_j) = 0$. Therefore, the classical DCA would fail, but a variant that consists in searching the key hypothesis with the lowest maximum correlation score would succeed with very high probability, as we explain now.

Let $n \geq 2$ and $u \in \mathbb{F}_2^n$ with $\mathrm{HW}(u) = 1$. Let f be a random variable following the uniform distribution on the set B_n of balanced n-variable Boolean functions. A consequence of [RW19, Lemma 1] is that $W_f(u)$ can be modeled as an affine transformation of a random variable X following a hypergeometric distribution with parameters $(2^n, 2^{n-1}, 2^{n-1})$:

$$W_f(u) \sim 4X - 2^n \ . \tag{4}$$

Proposition 3 trivially translates (4) into:

$$\mathrm{Cor}(f(x), x_j) \sim \frac{4}{2^n} X - 1 \ . \tag{5}$$

Let $x^{(\hat{k})}$ be an S-box output computed under the key hypothesis $\hat{k}$. Then, setting $y = E(x)$, we get $\mathrm{Cor}(E_i(x), x_j^{(\hat{k})}) = \mathrm{Cor}(E_i(x), g^{(\hat{k})}(E(x))) = \mathrm{Cor}(g^{(\hat{k})}(y), y_i)$ with:

$$\begin{aligned} g^{(\hat{k})} : \mathbb{F}_2^8 &\to \mathbb{F}_2 \\ z &\mapsto \mathrm{S}_j(\mathrm{S}^{-1}(E^{-1}(z)) \oplus \hat{k} \oplus k) \ . \end{aligned}$$

Thanks to the good cryptographic properties of S, we can assimilate the functions $g^{(\hat{k})}$, $\hat{k} \in K \setminus \{k\}$, to outcomes of a random variable following the uniform distribution on B_8. Then, (5) holds, which means that, for all $k^\times \in K \setminus \{k\}$, $\mathrm{Cor}(E_i(x), x_j^{(k^\times)})$ is an outcome of the random variable $\mathrm{Cor}(f(x), x_j)$. Therefore:

$$\mathbb{P}(\mathrm{Cor}(E_i(x), x_j^{(k^\times)}) = 0) = \mathbb{P}\left(X = \frac{2^8}{4}\right) \ .$$

The probability for $\rho^\times$ to be equal to 0 for a wrong key guess $k^\times$ is the probability that $\mathrm{Cor}(E_i(x), x_j^{(k^\times)}) = 0$ for all $0 \leq i < 8$. Let us assume, as in [RW19], that $E_i(x)$ and $E_{i'}(x)$ are two independent and identically distributed random variables when $i \neq i'$. Then:

$$\mathbb{P}(\rho^\times = 0) = \mathbb{P}\left(X = \frac{2^8}{4}\right)^8 .$$

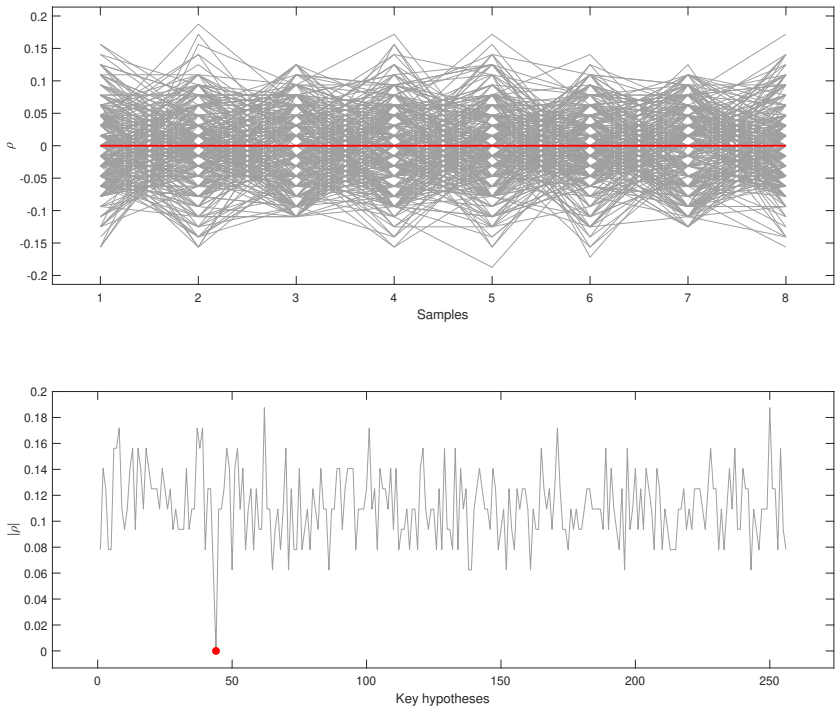

Figure 1: DCA targeting the most significant bit of the S-box output protected by 4-bit encodings having only zero in their Walsh spectrum. Top: Correlation traces for wrong (resp. good) key guesses in grey (resp. red). Bottom: The highest absolute value of the correlation scores for each key guess (good hypothesis highlighted in red).

Consequently, the correct key guess is the only one for which $\rho = 0$ with probability $(1-\mathbb{P}(X=2^6)^8)^{\#K-1} \approx 1-2^{-18.6}$. Therefore, the correct key guess can be identified with very high probability as the only one that never gives any correlation. We verified this experimentally (see Fig. 1), performing a DCA targeting the output of S_7 on 2^8 simulated traces: one trace per possible value of the input byte m, each trace being computed as $E_0 \circ S(m \oplus k) || \ldots || E_7 \circ S(m \oplus k)$.

Since selecting encodings $E = E^{(1)} \| E^{(2)}$ such that $\mathcal{W}_1(E) = \{0\}$ does not work, one could think of crafting encodings $E^{(1)}$ and $E^{(2)}$ with only the lowest possible non-zero (absolute) value in their Walsh spectrum. The latter being equal to 64 in the case of 4-bit encodings applied to the S-box output, we would have $\mathcal{W}_1(E^{(i)}) \subseteq \{-64, 64\}$ for $i \in \{1, 2\}$. Then, if one targets the bit $x_j^{(k)}$ with $0 \leq j < 4$, the correlation peaks should equal ± 0.25 for all point $E_i(x)$ with $0 \leq i < 4$. Similarly, if one targets the bit $x_j^{(k)}$ with $4 \leq j < 8$, the correlation peaks should equal ± 0.25 for all point $E_i(x)$ with $4 \leq i < 8$. This phenomenon can be observed in Fig. 2.

The probability that a wrong key hypothesis exhibits a correlation score higher than 0.25 on some of the eight points equals:

$$\mathbb{P}\left(|\rho^{\times}| > \frac{1}{4}\right) = 1 - \mathbb{P}\left(\frac{2^8 - 2^6}{4} < X < \frac{2^8 + 2^6}{4}\right)^8 \approx 2^{-10.3} \ .$$

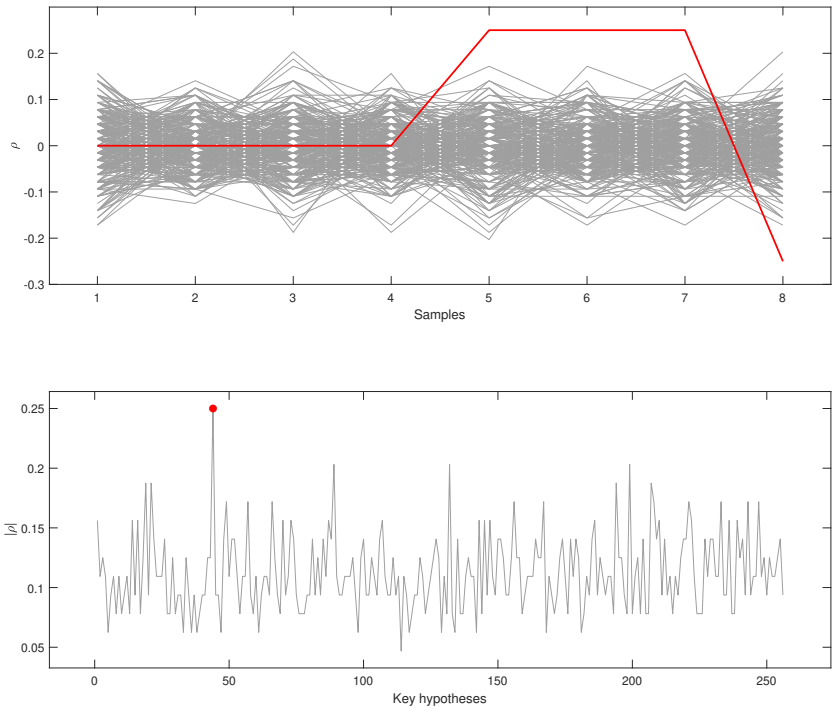

Figure 2: DCA targeting the most significant bit of the S-box output. Here, the Walsh spectrum of the 4-bit encodings do not contain any other value than 64 or -64. Top: Correlation traces for wrong (resp. good) key guesses in grey (resp. red). Bottom: The highest absolute value of the correlation scores for each key guess (good hypothesis highlighted in red).

It would thus be very unlikely that the DCA does not succeed. Furthermore, even if there are some higher peaks for some bad key guesses, the attacker could just focus on key hypotheses that give a correlation coefficient equal to ± 0.25 as noticed by Alpirez Bock et al. [ABBMT18].

Moreover, a DCA searching for the least correlation score could still succeed. Indeed, as a consequence of (1), if one targets the bit $x_j^{(k)}$ with $0 \leq j < 4$, then the correlation scores under the correct key guess for the points $E_i(x)$ for all $4 \leq i < 8$ would equal 0. Similarly, if one targets the bit $x_j^{(k)}$ with $4 \leq j < 8$, then the correlation scores under the correct key guess for the points $E_i(x)$ for all $0 \leq i < 4$ would equal 0 as well. The probability of this happening under a wrong key guess is $\mathbb{P}(\rho^{\times} = 0) = \mathbb{P}(X = 2^8/4)^4 \approx 2^{-13.3}$.

Therefore, we can conclude that a white-box implementation using 4-bit encodings to protect the S-box output can always be broken by DCA. The attacker could first perform a classical DCA and see if any key hypothesis gives very high correlations. If this is not the case, the attacker could verify if any key hypothesis gives correlation scores equal to 0 or ± 0.25.

3.2 On 8-bit Encodings

In the previous section, we have seen that one cannot successfully prevent a DCA targeting one bit of an S-box output by using 4-bit encodings. The case of 8-bit encodings is different.

Rivain and Wang [RW19] showed that applying random 8-bit encodings instead of two concatenated 4-bit ones decreases the success probability of such a DCA from 0.926 to approximately 0.0025. While this probability is notably low, it is not absolute zero. One might thus consider discarding encodings with Walsh spectrum values exceeding a certain threshold. This would prevent an excessively high correlation score for the correct key guess, but somewhat unexpectedly, it is actually counterproductive. Indeed, as we explain now, randomly selecting 8-bit encodings allows to decorrelate the sensitive value from the trace points, which is the best possible protection.

Let $E : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ be a random encoding. As argued in Sect. 3.1, for all $k^\times \in K \setminus \{k\}$, $\mathrm{Cor}(E_i(x), x_j^{(k^\times)})$ is an outcome of the random variable $\mathrm{Cor}(f(x), x_j)$. Moreover, so is $\mathrm{Cor}(E_i(x), x_j^{(k^*)})$, as the random selection of E from the set of bijections in $\mathbb{F}_2^8$ enables E_i to be considered an outcome of a random variable following $\mathcal{U}(B_8)$. This implies that the correlation scores of all key hypotheses follow the same distribution. Consequently, the probability of observing a given score is independent of the key byte's value, and all hypotheses share an equal probability of obtaining the highest correlation score.

Another way to get this intuition is to realise that for any wrong key guess $k^\times$, there exists an encoding E' such that $E'(\mathrm{S}(x \oplus k^\times)) = E(\mathrm{S}(x \oplus k))$, namely

$$E'(z) = E(\mathrm{S}(\mathrm{S}^{-1}(z) \oplus k^\times \oplus k)) \ .$$

This shows that without the knowledge of the encodings, the trace points do not give any information on the correct key byte. Note that a similar argument can be given for the protection of other sensitive variables, as long as the encoded value is in bijection with the part of the plaintext involved in its computation. In other words, randomly selecting encodings having the same size as the sensitive variable that they are supposed to protect is an efficient countermeasure against side-channel attacks.

In summary, random 8-bit encodings can thus serve as a flawless solution for the protection of the S-box output. Unfortunately, we will see in the next section that securing the MixColumns output is a far greater challenge.

Remark 1. At first glance, the success rate of 0.0025 computed by Rivain and Wang might seem contradictory to our conclusion which might mistakenly suggest that the success probability should equal $1/256 \approx 0.0039$. Nevertheless, it should be noted that the success probability given by Rivain and Wang corresponds to the event where the correct key guess obtains the maximum score *alone*. It is thus expected that this probability is lower than $1/256$. When considering the possibility of multiple candidates achieving the maximum score simultaneously, the success rate is actually higher than $1/256$ since the events "$k^{(i)}$ has the maximum score" and "$k^{(j)}$ has the maximum score" are not disjoint.

4 Protecting the MixColumns Output

4.1 Attack Scenario

Considering what has been discussed in the previous section to protect the S-box output, a straightforward way to protect the output of the first-round's MixColumns from a side-channel attacker would consist in evaluating MixColumns through a large look-up table that takes a 32-bit input and returns a 32-bit value encoded through a bijection of $\mathbb{F}_2^{32}$. The unaffordable drawback of this solution is the huge amount of memory that each such table would require.

For this reason, tables $M_{r,i}$ are used in [CEJVO03b], protected by either 4-bit or 8-bit encodings. Since $M_{r,i}$ contains the S-box output by definition of MC, it cannot be encoded with 4-bit encodings to resist DCA. Therefore, we only consider in the following the case of 8-bit encodings.

Rivain and Wang theoretically studied the side-channel resistance of this strategy in [RW19]. They considered an output byte of the first round's MixColumns as target for a DCA. Although it is computationally feasible, guessing four bytes of the key to predict the targeted sensitive byte can be quite long. Nevertheless, if the attacker is able to choose the messages to encrypt, he may reduce the number of hypotheses by setting two appropriate bytes of the plaintexts to a constant. If we denote the plaintext by (m_0, m_1, m_2, m_3), with m_2 and m_3 two constants, the targeted byte s can be written as:

$$\begin{aligned} s &= \mathrm{S}(m_0 \oplus k_0) \oplus \mathrm{S}(m_1 \oplus k_1) \oplus 2 \cdot \mathrm{S}(m_2 \oplus k_2) \oplus 3 \cdot \mathrm{S}(m_3 \oplus k_3) \\ &= \mathrm{S}(m_0 \oplus k_0) \oplus \mathrm{S}(m_1 \oplus k_1) \oplus c \end{aligned} \tag{6}$$

for some unknown constant c. Then, the value $\mathrm{S}(m_0 \oplus k_0) \oplus \mathrm{S}(m_1 \oplus k_1)$ can be predicted under an only 16-bit key hypothesis while c does not hinder the attacker since the addition of a constant only affects the sign of the correlation scores.

In such an attack scenario, Rivain and Wang showed that, if the encodings that hide the output of the MixColumns are random bijections over $\mathbb{F}_2^8$, the probability of success of a DCA targeting one bit of s is very close to 1 – approximately 0.99995. Nevertheless, this probability is not 1, so the problem that naturally arises from this result is to describe the set of encodings that prevent these attacks.

4.2 On Encodings with a Null Walsh Spectrum

We know from our analysis in Sect. 3.1 that, in order to reduce the correlation scores between a sensitive variable and its encoded version, one needs to select an encoding with a Walsh spectrum that contains only small (absolute) values. The question of describing the set of encodings that prevent DCA then boils down to finding an appropriate bound on the maximum tolerable (absolute) value in the Walsh spectrum of the encoding.

Let us first investigate the case $\mathcal{W}_1(E) = \{0\}$. We have seen in Sect. 3.1 that when targeting an S-box output protected by an encoding with a Walsh spectrum containing only 0, the attacker may distinguish the correct key guess as the one showing the least correlation. Consequently, one could wonder if the same phenomenon happens when targeting a MixColumns output. We thus made an experiment. Let $f : \mathbb{F}_2^8 \times \mathbb{F}_2^8 \to \mathbb{F}_2^8$ be the function $(x_0, x_1) \mapsto \mathrm{S}(x_0 \oplus k_0) \oplus \mathrm{S}(x_1 \oplus k_1)$. We performed a DCA targeting the output of f_0 on 2^{16} simulated traces: one trace per possible value of (x_0, x_1), each trace being computed as $E_0 \circ f(x_0, x_1) || \dots || E_7 \circ f(x_0, x_1)$. The result of this experiment is shown in Fig. 3.

Contrary to what we observed for the S-box output in Fig. 1, the correct key guess is far from being the only one with a correlation score at 0 when $\mathcal{W}_1(E) = \{0\}$. In fact, we count exactly 511 key hypotheses with a null score out of 2^{16}. Since the application of [RW19, Formula (2)] gives a probability $\mathbb{P}(\rho^{\times} = 0) \approx 2^{-58.6}$ that an incorrect key guess produces a null score, these hypotheses cannot be a random outcome. Figure 4 shows a structured behaviour that strengthens this intuition. To understand what happens, we have to study in depth the behaviour of the correlation coefficient when the key guess is incorrect.

Since the functions f and $E \circ f$ are surjective and balanced, so are their coordinate functions. Thus, for any $0 \leq i, j < 8$, the correlation score between $E_j \circ f$ and f_i can be calculated using (3), seeing f as a function from $\mathbb{F}_2^{16}$ to $\mathbb{F}_2^8$:

$$\begin{aligned} \mathrm{Cor}(E_j \circ f, f_i) &= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} \sum_{x_1 \in \mathbb{F}_2^8} (-1)^{E_j \circ f(x_0, x_1) \oplus f_i(x_0, x_1)} \\ &= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} \sum_{x_1 \in \mathbb{F}_2^8} (-1)^{E_j \circ f(x_0, x_1) \oplus \mathrm{S}_i(x_0 \oplus k_0) \oplus \mathrm{S}_i(x_1 \oplus k_1)} . \end{aligned}$$

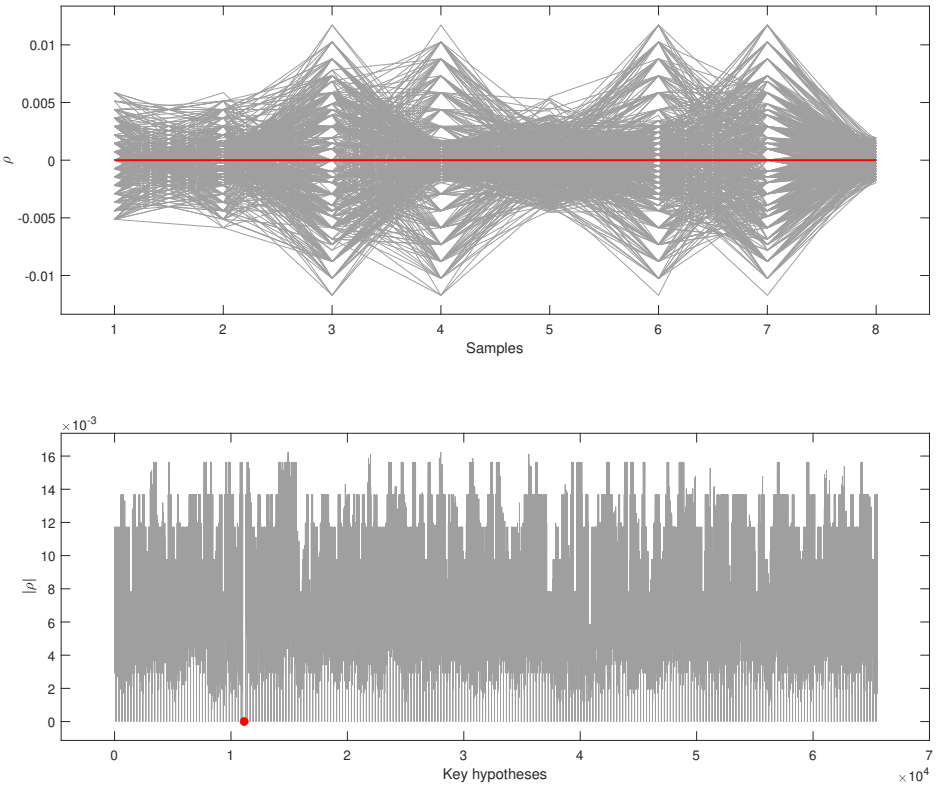

Figure 3: DCA targeting f_0. Here, the Walsh spectrum of the 8-bit encoding contains only 0. Top: Correlation traces for wrong (resp. good) key guesses in grey (resp. red). Bottom: The highest absolute value of the correlation scores for each key guess (good hypothesis highlighted in red).

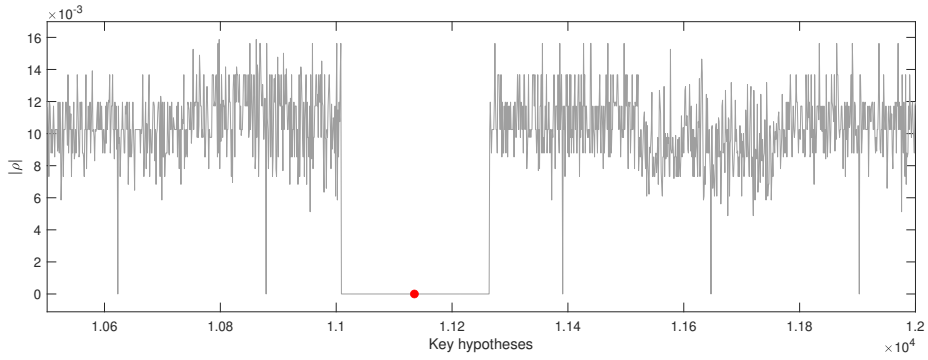

Figure 4: Zoom on the correct key guess of the bottom part of Fig. 3.

Among all the 2^{16} possible hypotheses for the couple (k_0, k_1), let us consider the elements from the set $\{(k'_0, k_1)\}_{k'_0 \in \mathbb{F}_2^8}$. Let $f' : (x_0, x_1) \mapsto S(x_0 \oplus k'_0) \oplus S(x_1 \oplus k_1)$. The correlation scores between $E_j \circ f$ and f'_i can be written as:

$$\begin{aligned}
\mathrm{Cor}(E_j \circ f, f'_i) &= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} \sum_{x_1 \in \mathbb{F}_2^8} (-1)^{E_j \circ f(x_0,x_1) \oplus \mathrm{S}_i(x_0 \oplus k'_0) \oplus \mathrm{S}_i(x_1 \oplus k_1)} \\
&= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} \sum_{x_1 \in \mathbb{F}_2^8} (-1)^{E_j \circ f(x_0,x_1) \oplus \mathrm{S}_i(x_0 \oplus k'_0) \oplus \mathrm{S}_i(x_0 \oplus k_0) \oplus \mathrm{S}_i(x_0 \oplus k_0) \oplus \mathrm{S}_i(x_1 \oplus k_1)} \\
&= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} (-1)^{\mathrm{S}_i(x_0 \oplus k'_0) \oplus \mathrm{S}_i(x_0 \oplus k_0)} \sum_{x_1 \in \mathbb{F}_2^8} (-1)^{E_j \circ f(x_0,x_1) \oplus \mathrm{S}_i(x_0 \oplus k_0) \oplus \mathrm{S}_i(x_1 \oplus k_1)} .
\end{aligned}$$

Notice that the sum over x_1 is equal to $W_{E_j}(\omega)$ for ω such that $f_i = \langle f, \omega \rangle$. Indeed, for all $x_0 \in \mathbb{F}_2^8$, $x_1 \mapsto f(x_0, x_1)$ is a bijection so we can make the change of variable $u = f(x_0, x_1)$ and write:

$$\begin{aligned}
\mathrm{Cor}(E_j \circ f, f'_i) &= \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} (-1)^{\mathrm{S}_i(x_0 \oplus k'_0) \oplus \mathrm{S}_i(x_0 \oplus k_0)} \sum_{u \in \mathbb{F}_2^8} (-1)^{E_j(u) \oplus \langle u, \omega \rangle} \\
&= W_{E_j}(\omega) \cdot \frac{1}{2^{16}} \sum_{x_0 \in \mathbb{F}_2^8} (-1)^{\mathrm{S}_i(x_0 \oplus k'_0) \oplus \mathrm{S}_i(x_0 \oplus k_0)} . \qquad (7)
\end{aligned}$$

Therefore, the correlation scores obtained under a wrong key guess (k'_0, k_1) are multiples of the Walsh transform of some coordinate function of the encoding. The same reasoning can be applied to key hypotheses of the form (k_0, k'_1) for all $k'_1 \in \mathbb{F}_2^8$. Consequently, when $\mathcal{W}_1(E) = \{0\}$, all the correlation scores for those 511 key guesses will be equal to zero.

No other key guess is expected to get a null correlation score since, as stated previously, it happens with probability $2^{-58.6}$ for a random guess. Then, when $\mathcal{W}_1(E) = \{0\}$, the correct key is revealed by the following procedure:

1. Gather in a set Z the hypotheses that get a null correlation score; there are at least 511 of them,
2. For all $z \in Z$, compute $z \bmod 256$; the value that appears the most frequently – at least 256 times – is the correct value of one key byte,
3. For all $z \in Z$, compute $\lfloor z/256 \rfloor$; the value that appears the most frequently – at least 256 times – is the correct value of the other key byte.

Since the implementation can be broken if the encodings have been selected such that $\mathcal{W}_1(E) = \{0\}$, one could be interested in what happens when encodings with small absolute values in their Walsh spectrum are preferred.

4.3 On Encodings with a Non-Null Walsh Spectrum

Let $E : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ be a random encoding with $\mathcal{W}_1(E) \neq \{0\}$. According to Proposition 1, any non-zero value in $\mathcal{W}_1(E)$ is a multiple of 4. This implies through Proposition 3 that $|\rho^*|$ is a multiple of $1/64$. In addition, [RW19, Formula (2)] implies that under an incorrect key guess, $\mathbb{P}(|\rho^\times| > 1/64) \approx 2^{-10.9}$. This means that even for the lowest non-zero possible absolute value of $\mathcal{W}_1(E)$, only about thirty key guesses in average show higher correlation scores than the correct one, as can be observed in Fig. 5. Hence, the attacker could successfully recover the entire AES key by brute-force (about $2^{(16-10.9)\cdot 8} \approx 2^{40}$ remaining keys to test). Thus, even if in such a case a DCA is considered by Rivain and Wang as having failed since the correct guess does not get the best correlation score, E should be regarded as insecure as soon as $\mathcal{W}_1(E)$ contains a non-zero value.

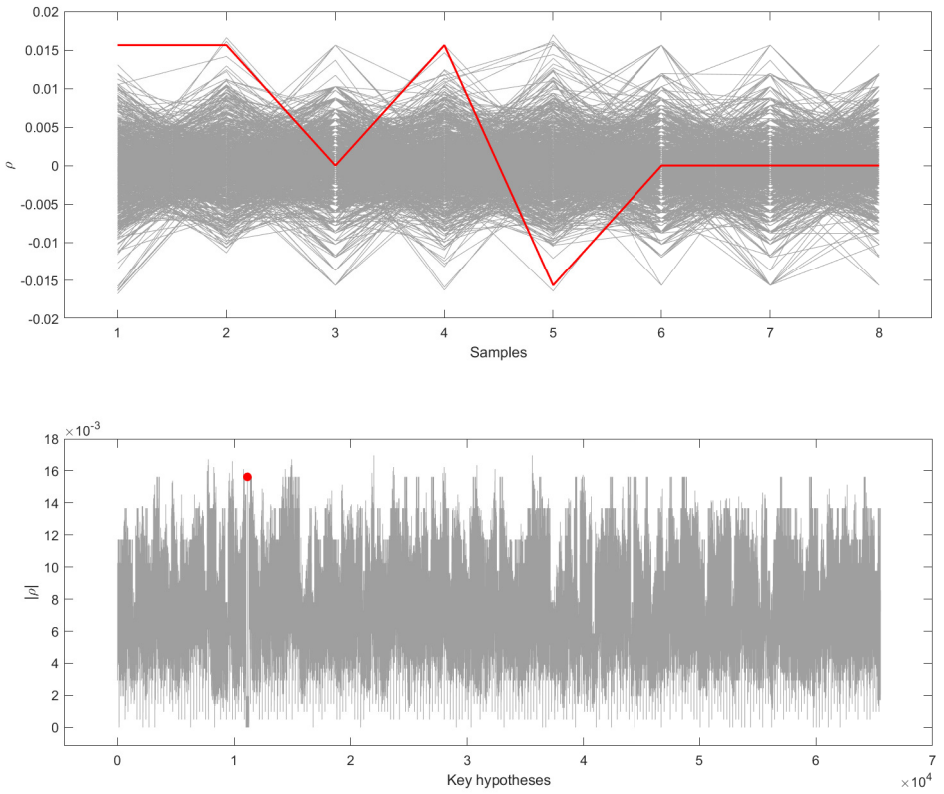

Figure 5: DCA targeting f_0. Here, the Walsh spectrum of the 8-bit encoding is included in $\{0, 4, -4\}$. Top: Correlation traces for wrong (resp. good) key guesses in grey (resp. red). Bottom: The highest absolute value of the correlation scores for each key guess. The good hypothesis is highlighted in red and is ranked 34^{th}.

Furthermore, note that (7) also has an effect that defeats any value of $\mathcal{W}_1(E)$ more efficiently. One can in fact exhaustively verify that the sum over x_0 can take only ten values: 0, 256, ± 8, ± 16, ± 24, ± 32, with 256 appearing only for $k'_0 = k_0$. Therefore, the distribution of the correlation scores when one of the guessed bytes is correct shows a specific behaviour that can be detected by an attacker and used to identify the correct byte value:

1. After the DCA has been performed, build 256 groups of 256 correlation traces, each group gathering the traces for which one byte of the key hypothesis is constant,
2. If, in a group, some samples from the correlation traces take exactly ten different values, then consider the corresponding constant key hypothesis as the correct one.

Figure 6 exhibits the particular distribution of the correlation scores when both key bytes are incorrect and when one of the two bytes is correct.

In the end, we have seen in Sect. 4.2 and Sect. 4.3 that no matter the 8-bit encodings that are selected to protect the MixColumns output, the resulting implementation can always be broken by DCA. Therefore, this work does rule out the existence of a particular class of encodings that could protect AES against side-channel attacks.

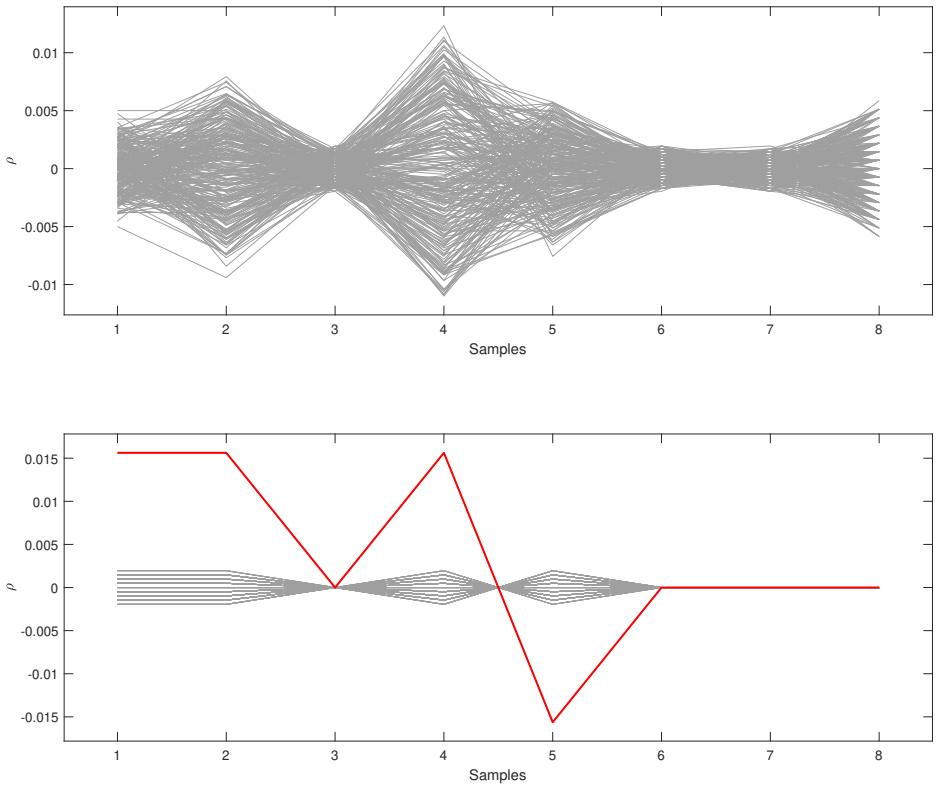

Figure 6: Top: Correlation traces for random key guesses. Bottom: Correlation traces when one of the two key guess bytes is correct. The curve highlighted in red corresponds to the two correct key bytes.

5 Conclusion

In this paper, we have elaborated on the research conducted by Rivain and Wang in [RW19]. Their work showed that the success probability of a DCA targeting an S-box output is quite high when employing random 4-bit encodings but significantly low with the use of their 8-bit counterpart. They also demonstrated that even random 8-bit encodings are insufficient to protect a MixColumns output with overwhelming probability. With these success rates being neither 0 nor 1, the question that arose from their work was the one of the existence of a particular class of encodings capable of thwarting DCA.

In this work, we studied the relevance of the strategy consisting in carefully crafting encodings with specific properties instead of drawing them at random. We initially delved into the protection brought by 4-bit encodings to the S-box output. We showed that as long as the Walsh spectrum of the encodings contains a non-zero value, the S-box output remains vulnerable to DCA. We then emphasised that opting for encodings with a Walsh spectrum exclusively consisting of 0 actually ensures that the correct key hypothesis will never attain the top ranking. However, we demonstrated that in such cases, the adversary could still differentiate the correct key hypothesis by identifying the one with the least correlation. Consequently, the use of 8-bit encodings becomes imperative for safeguarding

the S-box output. We argued that these encodings must be randomly selected, as this leads to complete decorrelation between the trace points and the sensitive value.

Protecting the MixColumns output is more difficult. Indeed, while using random 32-bit encodings would effectively protect this sensitive value, such an approach would be impractical due to the extensive memory space requirements. We thus got interested in 8-bit encodings. In contrast to the S-box output scenario, we demonstrated that selecting encodings with a null Walsh spectrum does not allow the attacker to distinguish the correct key hypothesis based on minimal correlation. However, we also illustrated how to adapt the attack and exploit the correlation traces in order to defeat even this specific class of encodings. We showed that this method can also be used to break any other encoding more efficiently than classical DCA. Consequently, we did rule out the existence of a particular class of 8-bit encodings capable of protecting the MixColumns output, and thereby an entire AES implementation, against side-channel attacks. Carefully crafting encodings is thus as pointless as drawing them at random.

Beyond this result, our work highlights the challenge of establishing a general result regarding the effectiveness of encodings against DCA. Indeed, while there exist encodings that prevent DCA under some conditions – met for example by the first round's S-box output, our attack on the MixColumns output shows that when these conditions are not satisfied, the effect of the encodings heavily depends on the operation they are supposed to protect. An interesting future work would thus be to analyse other cryptographic algorithms. Specifically, finding – or building – a cryptosystem exclusively employing operations that can be provably secured against side-channel attacks through the application of encodings would be a significant breakthrough in white-box cryptography.

Acknowledgments

We would like to thank Emmanuelle Dottax, Christophe Giraud and Laurent Grémy for their helpful comments on the preliminary version of this paper. We also thank Arnaud Casteigts for our interesting discussions and for helping us generate the encodings that we used in our experiments.

References

[ABBMT18] 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*, pages 103–120, Cham, 2018. Springer International Publishing.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. White box cryptography: Another attempt. Cryptology ePrint Archive, Paper 2006/468, 2006.

[BGEC05] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a white box aes implementation. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, pages 227–240, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[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*, pages 215–236, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[BU18] Alex Biryukov and Aleksei Udovenko. Attacks and countermeasures for white-box designs. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 373–402, Cham, 2018. Springer International Publishing.

[CEJvO03a] Stanley Chow, Phil Eisen, Harold Johnson, and Paul C. van Oorschot. A white-box des implementation for drm applications. In Joan Feigenbaum, editor, *Digital Rights Management*, pages 1–15, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[CEJVO03b] Stanley Chow, Philip Eisen, Harold Johnson, and Paul C. Van Oorschot. White-box cryptography and an aes implementation. In Kaisa Nyberg and Howard Heys, editors, *Selected Areas in Cryptography*, pages 250–270, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[DMRP13] 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*, pages 34–49, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[DMWP10] 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*, pages 292–310, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[DR99] J. Daemen and V. Rijmen. *AES Proposal: Rijndael*, September 1999.

[HBG23] Tobias Holl, Katharina Bogad, and Michael Gruber. Whiteboxgrind – automated analysis of whitebox cryptography. In Elif Bilge Kavun and Michael Pehl, editors, *Constructive Side-Channel Analysis and Secure Design*, pages 221–240, Cham, 2023. Springer Nature Switzerland.

[Kar11] Mohamed Karroumi. Protecting white-box aes with dual ciphers. In Kyung-Hyune Rhee and DaeHun Nyang, editors, *Information Security and Cryptology - ICISC 2010*, pages 278–291, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[LRDM+14] Tancrède Lepoint, Matthieu Rivain, Yoni De Mulder, Peter Roelse, and Bart Preneel. Two attacks on a white-box aes implementation. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography – SAC 2013*, pages 265–285, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[Mui13] James A. Muir. A Tutorial on White-Box AES. Cryptology ePrint Archive, Paper 2013/104, 2013.

[RW19] Matthieu Rivain and Junwei Wang. Analysis and improvement of differential computation attacks against internally-encoded white-box implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019, Issue 2:225–255, 2019.

[SMG16] Pascal Sasdrich, Amir Moradi, and Tim Güneysu. White-box cryptography in the gray box. In Thomas Peyrin, editor, *Fast Software Encryption*, pages 185–203, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[WMGP07] Brecht Wyseur, Wil Michiels, Paul Gorissen, and Bart Preneel. Cryptanalysis of white-box des implementations with arbitrary external encodings. In Carlisle Adams, Ali Miri, and Michael Wiener, editors, *Selected Areas in Cryptography*, pages 264–277, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[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.