STRUCTURAL WEAKNESSES OF PERMUTATIONS WITH A LOW DIFFERENTIAL UNIFORMITY AND GENERALIZED CROOKED FUNCTIONS

Anne Canteaut and María Naya-Plasencia

Abstract. Any permutation with a low differential uniformity is shown to be such that its inverse has a derivative with a large image set. An attack exploiting this structural property is then presented against a recent hash function proposal, named Maraca, submitted to the SHA-3 competition. Moreover, the attack can be made much more efficient when the image sets of the derivatives of the inverse permutation are affine subspaces. This cryptanalytic approach leads to some generalizations of the notion of crooked functions, and to the study of their properties.

1. Introduction

Statistical attacks like differential and linear attacks are major cryptanalytic tools which apply to most cryptographic primitives. Around twenty years after the seminal paper by Biham and Shamir [**BS91**], all designers must provide with evidence that their primitives resist these attacks. Therefore, the search for functions which guarantee a high resistance to these attacks has been a major research area. Most notably, optimal functions regarding the corresponding security criteria, *e.g.* APN functions and AB functions, have been extensively studied. However, optimality is usually due to some particular algebraic or combinatorial structure. Thus, it can be wondered whether the related structure causes a weakness within the primitive. The most famous example of such a situation is the use of the inverse function over the field $\mathbf{F}_{2^8}$ as the nonlinear part of the block cipher standard AES, which provides with quadratic relations between the input and output bits of each round [**CP02**]. More generally, the following question arises: can the use of an APN function or of a function with a low differential uniformity be exploited for mounting an attack?

2000 *Mathematics Subject Classification.* 11T71; 06E30.

Key words and phrases. Boolean functions, differential uniformity, APN functions, crooked functions, hash functions, differential cryptanalysis.

This work was supported in part by the French Agence Nationale de la Recherche under Contract ANR-06-SETI-013-RAPIDE..

Here, we introduce another property which is highly related to the differential uniformity of a permutation: we focus on the highest number ∇_F of input differences which can lead to the same nonzero output difference. There is a trade-off between this quantity and the differential uniformity, implying that all permutations which guarantee a good resistance to differential cryptanalysis have a high ∇_F. But, we show that a high ∇_F may introduce an unexpected weakness within the underlying primitive: we present an attack based on this property against a new hash function named Maraca, which has been submitted to the SHA-3 competition. We also point out that, besides their cardinalities, the algebraic structures of the image sets of the derivatives of the inverse permutation are of great importance, in particular the case where these sets are affine subspaces is the most favourable one for the attacker. In other words, we show that the use of APN permutations satisfying the crooked property [**vDdF00, BdF98**] makes the primitive very weak in the context of Maraca. This also leads us to introduce a natural generalization of the crooked property in the light of our attack, which captures the functions with a higher differential uniformity and a higher nonlinearity.

The rest of the paper is organized as follows. In Section 2 the main concepts required for quantifying the resistance to differential attacks are recalled and the new quantity ∇_F is introduced; the link between both notions is also established. Section 3 shows how a high ∇_F can exploited for mounting an attack against Maraca. Moreover, we point out that the attack is even more efficient when the original inner permutation in Maraca is replaced by a function with a higher nonlinearity or with a lower differential uniformity, like the inverse function. Since our attack emphasizes the role played by the algebraic structures of the image sets of the derivatives, Section 4 finally focuses on the functions whose derivatives take their values in some affine subspaces. This leads to a generalization of the crooked property. We then prove several properties related to these new notions and provide with some open problems.

2. A structural property of permutations with a low differential uniformity

2.1. Resistance to differential cryptanalysis. The resistance of a cryptographic primitive to statistical attacks such as linear cryptanalysis or differential cryptanalysis mainly depends on the resistance provided by its nonlinear building blocks. These building blocks, which are named S(ubstitution)-boxes in the context of block ciphers, are mappings from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$, $m > 1$. These mappings are usually chosen to be permutations for many reasons: in the case of a block cipher, the whole cipher must obviously be a permutation for any fixed key, otherwise some ciphertexts will correspond to several plaintexts; for other types of primitives, the use of a permutation enables the designer to guarantee that there is no entropy loss during the computation (see e.g. [**Röc08**]).

Differential cryptanalysis has been introduced by Biham and Shamir [**BS91**] against block ciphers but it also applies to many other primitives like stream ciphers or hash functions. The underlying idea is to consider several pairs of inputs (x, x') in $\mathbf{F}_2^n$ whose difference is a given constant: $x + x' = \alpha$. Then, a differential attack may be mounted if, at some point of the considered primitive (typically at the output of the primitive, or before the last iteration), the difference between the images of x and x' takes some given value $\beta \in \mathbf{F}_2^n$ more often than the other ones.

These attacks then exploit the existence of a nonzero input difference α and of an output difference β such that

$$F(x+\alpha)+F(x)=\beta$$

for many elements $x \in \mathbf{F}_2^n$. For the most commonly used types of block ciphers, it is known that the existence of such a pair (α, β) depends on the existence of a similar property for the constituent Sbox [**NK95, HLL$^+$00**].

Clearly, the resistance to differential cryptanalysis is then related to the properties of the derivatives of the involved function.

Definition 2.1. Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. For any $a \in \mathbf{F}_2^n$, the *derivative of F in direction a* is the function D_aF from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ defined by

$$D_aF(x)=F(x+a)+F(x), \ \ \forall x \in \mathbf{F}_2^n \ .$$

It is well-known that the resistance of a cipher to differential cryptanalysis can be quantified by its differential uniformity.

Definition 2.2. [**Nyb93**] Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. For any a and b in $\mathbf{F}_2^n$, we define

$$\Delta_F(a,b)=\#\{x \in \mathbf{F}_2^n, \ D_aF(x)=b\}.$$

The multiset

$$\{\Delta_F(a,b), \ a,b \in \mathbf{F}_2^n, a \neq 0\}$$

is called the *differential spectrum of F*. The *differential uniformity* of F is defined by

$$\Delta_F = \max_{a\neq 0, \ b \in \mathbf{F}_2^n} \Delta_F(a,b).$$

Those functions for which $\Delta_F=2$ are said to be *almost perfect nonlinear (APN).*

For implementation reasons, most applications handle functions depending on an even number of variables, n. Since no APN permutation was known in that case until very recently [**Dil09**], most applications use permutations F with $\Delta_F=4$. It is worth noticing that, for applications dedicated to hardware environments, the implementation cost of the function is also a major constraint. Therefore, the most commonly used permutation of this type is probably the inverse function $x \mapsto x^{2^n-2}$ over the field $\mathbf{F}_{2^n}$.

2.2. Practical interpretation of the image sets of the derivatives of a permutation. We now introduce a new property which is highly related to the resistance of a permutation F to differential cryptanalysis.

Definition 2.3. Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. For any $\beta \in \mathbf{F}_2^n$, the *set of differences leading to β* is defined by

$$\mathcal{D}_F(\beta)=\{\alpha \in \mathbf{F}_2^n, \ \exists x \in \mathbf{F}_2^n, D_\alpha F(x)=\beta\}.$$

Then, we define

$$\nabla_F = \max_{\beta \in \mathbf{F}_2^n} \#\mathcal{D}_F(\beta).$$

Then, ∇_F is the highest number of input differences which can lead to the same output difference. When F is a permutation, then the sets $\mathcal{D}_F(\beta)$ correspond to the image sets of the derivatives of the inverse function F^{-1}, as shown in the next proposition.

Proposition 2.4. *Let F be a permutation over $\mathbf{F}_2^n$. For any $\beta \in \mathbf{F}_2^n$ we have:*

$$\begin{aligned}\mathcal{D}_F(\beta) &= \{\alpha \in \mathbf{F}_2^n, \exists x \in \mathbf{F}_2^n, F(x+\alpha)+F(x)=\beta\} \\ &= \{F^{-1}(x+\beta)+F^{-1}(x),\ \ x \in \mathbf{F}_2^n\} = \mathcal{I}m\left(D_\beta F^{-1}\right).\end{aligned}$$

Proof. Let $x \in \mathbf{F}_2^n$ be a solution of

$$F(x+\alpha)+F(x)=\beta.$$

With $y=F(x)$, this equation can equivalently be written as

$$y+\beta = F(x+\alpha)$$

that means

$$F^{-1}(y+\beta)=F^{-1}(y)+\alpha.$$

We then deduce that the set $\mathcal{D}_F(\beta)$ consists of all values $(F^{-1}(y+\beta)+F^{-1}(y))$ when y varies in $\mathbf{F}_2^n$. □

A particular family of permutations of $\mathbf{F}_2^n$ is the class of all monomial permutations $x \mapsto x^s$ where $\mathbf{F}_2^n$ is identified with the finite field with 2^n elements. Since the particular family of monomials permutations has been extensively studied and also since it corresponds to functions with a reasonable implementation cost in hardware, it plays a particular role both in practice and in theoretical works. In the following, the *degree* of a monomial function refers to its multivariate degree, *i.e.*, to the degree of the corresponding function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$, even if the function is described by a univariate polynomial in $\mathbf{F}_{2^n}[X]$. Here, it is important to point out that, for monomial permutations, all sets $\mathcal{D}_F(\beta), \beta \neq 0$ have the same size and the same structure.

Lemma 2.5. *Let $F : x \mapsto x^s$ be a monomial permutation of $\mathbf{F}_{2^n}$. Let d be the exponent of the inverse function of F,* i.e., *$ds \equiv 1 \bmod 2^n-1$. Then, for any nonzero $\beta \in \mathbf{F}_{2^n}$,*

$$\mathcal{D}_F(\beta) = \beta^d \mathcal{D}_F(1).$$

Proof. This is an immediate consequence of the fact that, for any $\beta \neq 0$ and for any $x \in \mathbf{F}_{2^n}$,

$$\begin{aligned}D_\beta F^{-1}(x) &= (x+\beta)^d + x^d \\ &= \beta^d \left[\left(\frac{x}{\beta}+1\right)^d + \frac{x}{\beta}\right] \\ &= \beta^d D_1 F^{-1}\left(\frac{x}{\beta}\right).\end{aligned}$$

□

Now, since $\mathcal{D}_F(\beta)$ corresponds to the image set of a derivative of F^{-1}, we deduce that any permutation F with a small Δ_F has a high ∇_F.

Theorem 2.6. *Let F be a permutation over $\mathbf{F}_2^n$ and let Δ_F denote its differential uniformity. Then, for any nonzero $\beta \in \mathbf{F}_2^n$, we have*

$$\#\mathcal{D}_F(\beta) \geq \frac{2^n}{\Delta_F}$$

and equality holds if and only if, for all $\alpha \in \mathbf{F}_2^n$, the equations

$$F(x+\alpha)+F(x)=\beta,$$

have either 0 *or* Δ_F *solutions.*

Proof. Let $x \in \mathbf{F}_2^n$ be a solution of

$$F^{-1}(x+\beta) + F^{-1}(x) = \alpha.$$

Since F is a permutation, this equivalently means that $y = F^{-1}(x)$ is a solution of

$$F(y+\alpha) + F(y) = \beta,$$

implying that both equations have the same number of solutions, *i.e.*,

$$\Delta_{F^{-1}}(\beta,\alpha) = \Delta_F(\alpha,\beta).$$

In particular, $\Delta_F = \Delta_{F^{-1}}$. Then, we have

$$\begin{aligned} 2^n = \sum_{\alpha \in \mathbf{F}_2^n} \Delta_{F^{-1}}(\beta,\alpha) &\leq \#\mathcal{D}_F(\beta) \max_\alpha \Delta_{F^{-1}}(\beta,\alpha) \\ &\leq \#\mathcal{D}_F(\beta)\Delta_{F^{-1}}, \end{aligned}$$

with equality if and only if

$$\forall \alpha \in \mathbf{F}_2^n, \alpha \neq 0, \ \Delta_F(\alpha,\beta) \in \{0, \Delta_F\}.$$

Then, we deduce that, for any $\beta \neq 0$,

$$\#\mathcal{D}_F(\beta)\Delta_F \geq 2^n.$$

□

Note that, for any permutation F, we obviously have $\mathcal{D}_F(0) = \{0\}$.

In particular, the permutations whose differential spectrum consists of two different values only (*i.e.* with a *two-valued differential spectrum*) seem to play a particular role. It is worth noticing that this situation holds for quadratic power permutations and their inverses, and also for all APN permutations.

Corollary 2.7. *Let F be a permutation of $\mathbf{F}_2^n$ and let Δ_F denote its differential uniformity. Then,*

$$\nabla_F = \frac{2^n}{\Delta_F}$$

if and only if F has a two-valued differential spectrum. In particular, if Δ_F is not a power of 2*, then*

$$\nabla_F > \frac{2^n}{\Delta_F}.$$

Proof. The first statement is a direct consequence of the previous theorem. Moreover, if Δ_F is not a power of 2, it is clear that

$$\nabla_F \Delta_F = 2^n$$

cannot be satisfied. The fact that Δ_F must be a power of 2 when F has a two-valued differential spectrum was first observed in [**BCC09**]. □

Example 2.8. It follows from the previous corollary that some permutations may have the same differential uniformity and different values of ∇_F. For instance, let us consider the following monomial permutations of $\mathbf{F}_2^n$ with $n = 2t$, t odd:

$$\begin{aligned} F_1 : x &\mapsto x^{2^{2k}-2^k+1} \text{ with } 2 \leq k < n \text{ and } \gcd(k,n) = 2, \\ F_2 : x &\mapsto x^{2^n-2}. \end{aligned}$$

It is known that both permutations are differentially 4-uniform. Actually, the first one is a monomial permutation corresponding to a Kasami exponent [**Kas71**] and it satisfies $\Delta_{F_1} = 2^{\gcd(k,n)}$ [**BCC09, HP08**]. Moreover, F_1 is known to have a two-valued differential spectrum. Therefore,

$$\nabla_{F_1} = 2^{n-2}.$$

The second function F_2 is the inverse function over $\mathbf{F}_{2^n}$. It is well-known that $\Delta_{F_2}(\alpha, \beta) = 4$ if and only if $\beta = \alpha^{-1}$ [**Nyb93**]. Thus, when x varies in $\mathbf{F}_{2^n}$ and differs from these 4 solutions, $((x+\beta)^{-1} + x^{-1})$ takes exactly $(2^{n-1} - 2)$ distinct values since each value is obtained for exactly 2 elements x. It follows that

$$\nabla_{F_2} = 2^{n-1} - 1.$$

We now investigate the extremal possible values for ∇_F.

PROPOSITION 2.9. *Let F be a permutation of $\mathbf{F}_2^n$. Then,*

$$1 \leq \nabla_F \leq 2^{n-1}.$$

Moreover,

- $\nabla_F = 1$ *if and only if F has degree* 1.
- $\nabla_F = 2^{n-1}$ *if and only if at least one of the derivatives of F^{-1} is 2-to-1. This occurs in particular when F is APN.*

PROOF.

- Obviously, the minimal value $\nabla_F = 1$ corresponds to the highest possible Δ_F, *i.e.*, $\Delta_F = 2^n$, which is achieved for functions of degree 1 only.
- The upper bound $\nabla_F \leq 2^{n-1}$ comes from the fact that, for any nonzero β, $D_\beta F^{-1}(x) = D_\beta F^{-1}(x+\beta)$ for all $x \in \mathbf{F}_2^n$, implying that $\#\mathcal{D}_F(\beta) \leq 2^{n-1}$. Moreover, equality holds if and only if there exists a nonzero $\beta \in \mathbf{F}_2^n$ such that $\#\mathcal{I}m(D_\beta F^{-1}) = 2^{n-1}$. Therefore, each value in $\#\mathcal{I}m(D_\beta F^{-1})$ is obtained for exactly two inputs.

□

It is worth noticing that some permutations with $\Delta_F \geq 4$ might satisfy $\nabla_F = 2^{n-1}$. But, if we only consider the subclass of monomial permutations, then $\nabla_F = 2^{n-1}$ if and only if F is an APN permutation (since we know from Lemma 2.5 that all $\mathcal{D}_F(\beta)$ have the same size for $\beta \neq 0$).

3. Cryptanalysis of the hash function Maraca exploiting a high ∇_F

In the previous section, it has been pointed out that, if F is a permutation with a low differential uniformity (which is suitable in most cryptographic applications), then there is an output difference β which can be obtained from many input differences. Thus, we can wonder whether this property, which is inherent to the permutations which provide with a good resistance to differential cryptanalysis, may introduce some unexpected weakness in the primitive involving such permutations. This question is now answered positively: an attack against a recently proposed hash function is presented which exploits the previously mentioned property.

3.1. Brief description of Maraca. A cryptographic hash function is a function which associates to a binary word of any length a digest with a fixed size (typically, between 256 and 512 bits). Cryptographic hash functions are used for checking data integrity (e.g., when the hash value is signed with a digital signature scheme). Therefore, an important security issue is that it must be impossible for an attacker to find a collision, *i.e.*, two messages with the same hash value. More precisely, a hash function is considered as broken if there exists an algorithm for finding a collision more efficiently than the so-called *generic algorithm*, which consists in computing the hash value of randomly chosen inputs until two inputs with the same hash value are found.

Maraca is a new keyed hash function which has been submitted to the SHA-3 competition [**Jen08**]. It is an iterated hash function: the message is split into blocks. Then, the initial state of the function is initialized by a constant, and the internal state is transformed by iterating a function parametrized by the successive message blocks. The round permutation in Maraca applies to the n-bit internal state, where $n = 1024$, but one of the main features is that each message block is inserted four times, separated by 46 rounds. Then, a usual differential attack requires the study of the difference propagation on at least 46 rounds of the function.

As a keyed hash algorithm, Maraca takes as inputs a message of any length and a key, and it produces a hash value in $\mathbf{F}_2^h$ where typical values for h are 256, 384 and 512. The original message is padded in order to get a message whose length is a multiple of n bits: the n-bit key is first appended to the message as a prefix, and the resulting message is then padded with a value depending on the key and on the message length. Then, the padded message is split into blocks M_i where i varies from 0 to $(\ell - 1)$, *i.e.,* the first message block M_0 corresponds to the key. Note that our collision attack is considering messages of the same length and with the same key.

The internal state in Maraca and the message blocks which are inserted at each round are elements of $\mathbf{F}_2^n$. Each message block M_i is inserted four times, at Rounds i, $(i+21-6(i \bmod 4))$, $(i+41-6((i+2) \bmod 4))$ and $(i+46)$. More precisely, the original value of M_i is inserted at Round i, while rotated versions of M_i are inserted at the other three rounds, with rotations of 128 bits, 3×128 bits and 6×128 bits respectively. From now on, these rotated versions of M_i are denoted by M_i', M_i'' and M_i'''. It is worth noticing that the last round which uses the message block M_i is Round $i + 46$.

The round function at Round i can be decomposed as follows:

- the new message block M_i is inserted for the first time by adding it to the current internal state (where the addition is the addition in $\mathbf{F}_2$);
- an inner permutation Perm of $\mathbf{F}_2^n$ is applied to the internal state;
- $(M'_{i-3-6((i+2) \bmod 4)} + M''_{i-23-6(i \bmod 4)} + M'''_{i-46})$ is added to the internal state;
- two iterations of Perm are applied to the internal state.

Then, we are ready to start the next round and to introduce the message block M_{i+1}, if any. If no message block has to be inserted anymore, the all-zero block is used. The message insertion phase ends up when all message blocks have been used four times, implying that, for an ℓ-block message, the message insertion phase consists of $(\ell + 46)$ rounds. The hash value in $\mathbf{F}_2^h$ is finally extracted from the internal state after applying 30 additional iterations of Perm.

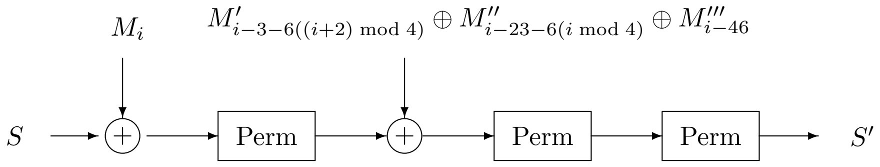
M_i
$M'_{i-3-6((i+2) \bmod 4)} \oplus M''_{i-23-6(i \bmod 4)} \oplus M'''_{i-46}$
S
Perm
Perm
Perm
S'

Figure 1. Round i in Maraca

The inner permutation Perm used in Maraca is formed by 128 parallel applications of a unique permutation P of $\mathbf{F}_2^8$ whose first three coordinates are linear:

$$\begin{array}{rcl} P_1(x_0,\ldots,x_7) & = & (x_0 \oplus x_4 \oplus x_5 \oplus x_7) \\ P_2(x_0,\ldots,x_7) & = & (x_1 \oplus x_2 \oplus x_3 \oplus x_5) \\ P_3(x_0,\ldots,x_7) & = & (x_1 \oplus x_3 \oplus x_4 \oplus x_5) \end{array}$$

and the other five coordinates are quadratic. A constant is then added to the result and this is finally followed by a bit permutation. Perm can then be seen as a function which takes as input an element $(b_1,\ldots,b_{128})$ in $(\mathbf{F}_2^8)^{128}$, and which outputs

$$\sigma(P(b_1),\ldots,P(b_{128}))$$

where σ is a permutation of the n bits composing a word of $\mathbf{F}_2^n$, *i.e.*,

$$\sigma(x_1,\ldots,x_n) = (x_{\pi(1)},\ldots,x_{\pi(n)})$$

with π a permutation of $\{1,\ldots,n\}$.

Since the internal state in Maraca has $n = 1024$ bits, the generic attack for finding an internal collision (*i.e.*, two messages which lead to the same final internal state) requires to hash around $2^{\frac{n}{2}}$ messages, corresponding to at least 46×2^{512} calls to the round permutation. Actually, because of the padding and of the fact that each message block is inserted at four different rounds, we cannot search for colliding internal states which correspond to different rounds.

The generic collision attack (*i.e.*, for finding two messages with the same hash value) for h-bit message digests requires to hash around $2^{\frac{h}{2}}$ messages, and requires at least $46 \times 2^{\frac{h}{2}}$ calls to the round permutation. Its time complexity basically corresponds to the cost of $2^{\frac{h}{2}}$ hashing.

3.2. General principle of the internal collision attack. Our attack against Maraca consists in finding two padded messages of the same length which lead to the same internal state. The attack exploits the fact that the inner permutation Perm has a relatively high ∇_{Perm}. This section first describes the general principle of the attack and exhibits the underlying property of the inner permutation. However, we will show that the time or the memory complexity of the attack might be higher than for the generic collision attack in some cases. This might be overcome by exploiting some algebraic structure of the inner permutation.

We consider two sets of padded messages using a given key $K \in \mathbf{F}_2^n$. Since all considered messages before padding are composed of 49 elements in $\mathbf{F}_2^n$, all of them are post-padded with the same value, pad, which only depends on K and on the message length. This value does not play any role in the attack since it is the same for all messages and it is involved in the computation after the internal states collide. Both sets of padded messages are defined as follows:

$$\mathcal{A} = \{\mathcal{M}_a = (K, a, 0^{47}, m, \mathrm{pad}),\ a \in \mathbf{F}_2^n\}$$

and

$$\mathcal{B} = \{\mathcal{M}_b = (K, b, 0, \gamma, 0^{45}, m, \mathrm{pad}),\ b \in \mathbf{F}_2^n\}$$

where γ and m are two fixed elements in $\mathbf{F}_2^n$ which will be defined later and where 0^i denotes the all-zero sequence in $\mathbf{F}_2^{ni}$.

Let S_a (resp. S_b) denote the internal state obtained at the beginning of Round 49 when $\mathcal{M}_a$ (resp. $\mathcal{M}_b$) is hashed. We aim at finding a collision on the internal state at Round 49, before the second application of Perm, as depicted on Figure 2. Round 49 for $\mathcal{M}_a$ (resp. $\mathcal{M}_b$) actually consists of the following operations:

- add m to the current internal state;
- apply Perm to the internal state;
- add 0 (resp. γ''') to the internal state;
- apply two additional iterations of Perm.

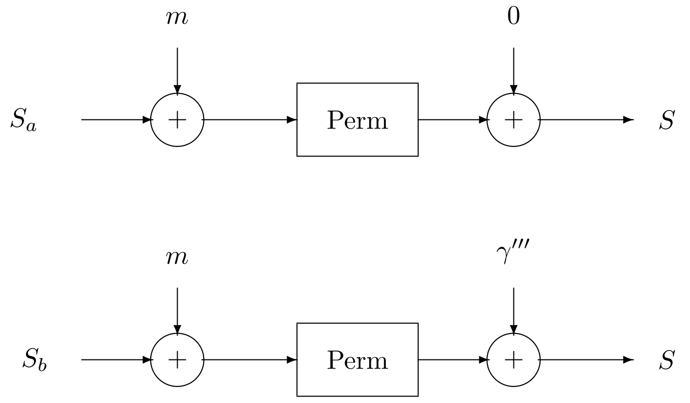

FIGURE 2. Beginning of Round 49 for $\mathcal{M}_a$ (top) and $\mathcal{M}_b$ (bottom)

This comes from the fact that all message blocks M_i, $3 \leq i \leq 48$, in $\mathcal{M}_a$ vanish, implying that there is no message insertion after the first application of Perm at Round 49. All message blocks M_i, $3 \leq i \leq 48$, in $\mathcal{M}_b$ vanish except $M_3 = \gamma$, implying that γ''', corresponding to γ rotated by 6×128, is xored to the internal state after the first application of Perm at Round 49.

Then, all message blocks which are inserted after Round 49 are equal for both message sets. Thus, an internal collision occurs as soon as we are able to find three message blocks a, b and m which satisfy

$$\mathrm{Perm}(S_a + m) = \mathrm{Perm}(S_b + m) + \gamma'''. \tag{3.1}$$

It is worth noticing that both S_a and S_b are independent of m.

Equation (3.1) with $x = S_a + m$ and $\delta = \gamma'''$ shows that finding an internal collision for both previously described message sets is equivalent to finding a pair (S_a, S_b) of internal states in $\mathbf{F}_2^n$ such that

$$\exists x \in \mathbf{F}_2^n,\ \ \mathrm{Perm}(x + S_a + S_b) + \mathrm{Perm}(x) = \delta, \tag{3.2}$$

for a fixed value of δ chosen by the attacker.

Equivalently, the attack consists in finding a pair (S_a, S_b) of internal states such that $(S_a + S_b) \in \mathcal{D}_{\text{Perm}}(\delta)$. As a comparison, the generic birthday attack for finding an internal collision consists in finding a pair (S_a, S_b) of internal states in $\mathbf{F}_2^n$ such that $S_a + S_b = 0$. Then, δ will be chosen such that $\mathcal{D}_{\text{Perm}}(\delta)$ has the largest possible size, *i.e.,* such that

$$\#\mathcal{D}_{\text{Perm}}(\delta) = \nabla_{\text{Perm}}.$$

Then, randomly choosing

$$N_a = N_b = \frac{2^{\frac{n}{2}}}{\sqrt{\nabla_{\text{Perm}}}}$$

messages in $\mathcal{A}$ and in $\mathcal{B}$ enables us to find a pair of internal states (S_a, S_b) at the beginning of Round 49 with $S_a + S_b \in \mathcal{D}_{\text{Perm}}(\delta)$. The data complexity of our attack, *i.e.* the number of calls to the hash function, is therefore smaller than the data complexity of the generic internal collision attack as soon as $\nabla_{\text{Perm}} > 1$, *i.e.,* as soon as Perm is not of degree 1. In the case where the size of the internal state, n, is larger that the length h of the message digest, as in Maraca, our attack leads to a collision attack with data complexity smaller than the generic collision attack if $\nabla_{\text{Perm}} > 2^{n-h}$. Note that, in our attack, each call to the hash function actually corresponds to 49 calls to the round function since the first 49 blocks in each message $\mathcal{M}_a$ and $\mathcal{M}_b$ have to be proceeded but message block 0 is constant and has to be evaluated only once. As a comparison, the generic collision attack requires at least 46 calls to the round functions (and 30 additional calls to Perm) for each message which is hashed.

Time complexity of the general attack. However, if the set of input differences $\mathcal{D}_{\text{Perm}}(\delta)$ does not have any particular structure, determining whether two internal states are such that $S_a + S_b \in \mathcal{D}_{\text{Perm}}(\delta)$ might be very time-consuming. The only general strategy which may have time complexity lower than $2^{\frac{n}{2}}$ consists in storing all N_a values of S_a and all N_b values of S_b in two tables. Then, all N_aN_b differences must be computed and compared to the elements in $D(\delta)$. This procedure has time complexity

$$N_aN_b \log(\nabla_{\text{Perm}}) = 2^n \frac{\log(\nabla_{\text{Perm}})}{\nabla_{\text{Perm}}}.$$

The attack is then faster than the generic internal collision attack only if $\nabla_{\text{Perm}} > 2^{\frac{n}{2}}$, and it is faster than the generic collision attack only if $\nabla_{\text{Perm}} > 2^{n-\frac{h}{2}}$. But, in general, comparing all differences $S_a + S_b$ with the elements of $\mathcal{D}_{\text{Perm}}(\delta)$ requires the storage of $\mathcal{D}_{\text{Perm}}(\delta)$, which needs an amount of memory higher than the complexity of the generic attack. However, this memory complexity can be much lower in some cases. For instance, if Perm corresponds to the concatenation of several copies of a smaller permutation P of $\mathbf{F}_2^k$ (even if it is followed by an affine permutation), then the attacker has to store the elements in

$$\mathcal{D}_P(\delta') = \{\alpha \in \mathbf{F}_2^k,\ \ \exists x \in \mathbf{F}_2^k,\ \ P(x+\alpha) + P(x) = \delta'\}$$

only, for some $\delta' \in \mathbf{F}_2^k$.

Let us now investigate different choices for Perm and their impacts on the complexity of our attack. Since the attack is faster than the generic attack if $\nabla_{\text{Perm}} > 2^{n-\frac{h}{2}}$, we deduce that this will be always the case if $\Delta_{\text{Perm}} \leq 2^{\frac{h}{2}}$. In the case where Perm consists of 128 copies of a permutation P of $\mathbf{F}_2^8$, like in Maraca, and for $h = 512$, this implies that Maraca is broken by our attack as soon as $\Delta_P \leq 4$. It is worth noticing that this is obviously not a necessary condition.

Attack against Maraca using the inverse permutation. A natural choice for the permutation P of $\mathbf{F}_2^8$ is the inverse function over $\mathbf{F}_{2^8}$ as in the AES, or any linearly equivalent permutation. It has been shown in Example 2.8 that the inverse function P over $\mathbf{F}_{2^s}$ satisfies

$$\#\mathcal{D}_P(\delta) = 2^{s-1} - 1$$

for any nonzero $\delta \in \mathbf{F}_{2^s}$. Then, with Maraca's parameters, $\nabla_{\mathrm{Perm}} = (2^7 - 1)^{128} = 2^{894.5}$. Our attack then requires to hash

$$N_a = N_b = 2^{64.7}$$

messages in $\mathcal{A}$ and $\mathcal{B}$. It is faster than the generic collision attack since examining all differences $(S_a + S_b)$ requires

$$128 \times 895 \times 2^{129.4} = 2^{146} \text{ operations}$$

and the memory cost is roughly 2^{76} bits. Therefore, if P is replaced by the inverse function in Maraca, our attack is efficient and its complexity is lower than the complexity of the generic attack when the length of the message digest exceeds 292.

Attack against Maraca using the original permutation. However, the permutation P which has been originally chosen in Maraca has not been so carefully designed regarding to differential attacks. The highest value for $\#D_P(\delta)$ is 21, and it is obtained for 20 output differences $\delta \in \mathbf{F}_2^8$. An example of a such an output difference is $\delta = \mathtt{0x3}$. Then, we deduce that $\nabla_{\mathrm{Perm}} = (21)^{128}$, which implies that the previously described attack is not faster than the generic collision attack.

3.3. Exploiting the algebraic structure of $\mathcal{D}_{\mathrm{Perm}}(\delta)$. Determining whether $S_a + S_b \in \mathcal{D}_{\mathrm{Perm}}(\delta)$ for all (S_a, S_b) is much easier when $\mathcal{D}_{\mathrm{Perm}}(\delta)$ has a simple algebraic structure.

When $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is an affine subspace or contains a large affine subspace. The simplest case is when $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is an affine subspace. Since Perm is a permutation, $\mathcal{D}_{\mathrm{Perm}}(\delta)$ does not contain 0, implying that $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is a coset of a linear subspace V. Let W be such that $V \oplus W = \mathbf{F}_2^n$. Then, we consider the case where

$$\mathcal{D}_{\mathrm{Perm}}(\delta) = c + V,\ c \in W.$$

Now, all pairs (S_a, S_b) with $S_a + S_b \in \mathcal{D}_{\mathrm{Perm}}(\delta)$ can be found by storing the list of all the elements s_a in W corresponding to the restrictions of S_a to W. Then, for each S_b, the attacker computes $s_b = (S_b)_W$ and she checks whether $s_b + c$ belongs to the list where c is the constant defining the affine subspace.

Then, when $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is an affine subspace of dimension d, the time complexity of the attack is $2(n-d)N_a = 2(n-d)2^{\frac{n-d}{2}}$. It requires the storage of a list of $(n-d)2^{\frac{n-d}{2}}$ bits. The attack then improves the generic collision attack if $d > n-h$.

It is worth noticing that the attack only exploits the fact that any element in the considered affine subspace belongs to $\mathcal{D}_{\mathrm{Perm}}(\delta)$. Therefore, the same attack can be mounted if $\mathcal{D}_{\mathrm{Perm}}(\delta)$ contains an affine subspace V of dimension d. In both cases, we have $N_a = N_b = 2^{\frac{n-d}{2}}$.

When $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is included in an affine subspace. In the case where the largest affine subspace included in $\mathcal{D}_{\mathrm{Perm}}(\delta)$ has dimension $d \leq n - h$, then the time complexity of our attack exceeds the time complexity of the generic collision attack. In this case, the existence of a larger (affine) subspace V of dimension d which contains many elements of $\mathcal{D}_{\mathrm{Perm}}(\delta)$ can be used as a sieve for selecting the pairs (S_a, S_b) whose differences belong to $\mathcal{D}_{\mathrm{Perm}}(\delta)$. The attack then aims at finding a pair (S_a, S_b) such that $(S_a + S_b) \in (\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V)$. The data complexity has now increased to

$$N_a = N_b = \frac{2^{\frac{n}{2}}}{\sqrt{\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right)}}$$

which improves the generic collision attack if

$$\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right) > 2^{n-h}.$$

But, the time complexity is much lower. Actually, once the much smaller list of pairs with difference in V has been obtained, all differences $(S_a + S_b)$ from this list can be exhaustively computed until a difference in $\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V$ is found. The sieving phase selects

$$\frac{N_a N_b}{2^{n-d}} = 2^d \frac{1}{\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right)}$$

pairs (S_a, S_b) among the $2^n \frac{1}{\#(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V)}$ possible pairs. The overall time complexity is then

$$\frac{2(n-d)2^{\frac{n}{2}}}{\sqrt{\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right)}} + \frac{2^d \log_2(\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right))}{\#\left(\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V\right)},$$

where the last term is the cost for checking whether a difference in the previous list belongs to $\mathcal{D}_{\mathrm{Perm}}(\delta) \cap V$. The attack is then faster than the generic collision attack as soon as the proportion of elements in V which belong to $\mathcal{D}_{\mathrm{Perm}}(\delta)$, *i.e.* $2^{-d}\#\left(\mathcal{D}(\delta) \cap V\right)$ exceeds $2^{-\frac{h}{2}}$.

3.4. Attack on Maraca-512. The previously described situation corresponds to the situation of Maraca. Actually, since the first three coordinates of P, P_i, $1 \leq i \leq 3$, are linear, we have that, for any $\delta \in \mathbf{F}_2^8$, $\mathcal{D}_P(\delta)$ is included in a 5-dimensional affine subspace. Thus, for the complete inner permutation Perm, there is an input difference $\delta \in \mathbf{F}_2^n$, such that $\#\mathcal{D}_{\mathrm{Perm}}(\delta) = (21)^{128}$ and $\mathcal{D}_{\mathrm{Perm}}(\delta)$ is included in an affine subspace V of dimension 640. Note that this is a particular case of the attack described in the previous section where it was allowed that some elements of $\mathcal{D}_{\mathrm{Perm}}(\delta)$ do not belong to V. With the parameters used in Maraca, the attack requires to compute the internal states at the beginning of Round 49 for

$$N_a = N_b = 2^{230.9}$$

messages in $\mathcal{A}$ and in $\mathcal{B}$. Using this subspace, we are able to find all pairs (S_a, S_b) whose differences belong to V. The average number of such pairs (S_a, S_b) is

$$\frac{N_a N_b}{2^{384}} = 2^{78}.$$

Now, for those 2^{78} favorable pairs of internal states, we have to check whether $(S_a + S_b)$ belongs to $\mathcal{D}_{\mathrm{Perm}}(\delta)$. This occurs with probability

$$\frac{\#\mathcal{D}_{\mathrm{Perm}}(\delta)}{2^{5\times 128}} = 2^{-78}.$$

Once such a pair has been found, we can pick up a value of x which makes possible to obtain the desired output difference from the input difference $S_a + S_b$. Such an x can be constructed as an element in $(\mathbf{F}_2^8)^{128}$, $(\mu_1, \ldots, \mu_{128})$, defined by

$$P(\mu_i + (S_a)_i) + P(\mu_i + (S_b)_i) = \delta_i$$

where S_a, S_b and δ are seen as elements in $(\mathbf{F}_2^8)^{128}$.

This procedure then leads to a pair of messages $\mathcal{M}_a \in \mathcal{A}$ and $\mathcal{M}_b \in \mathcal{B}$ such that

$$\mathrm{Perm}(S_a + m) = \mathrm{Perm}(S_b + m) + \gamma''',$$

i.e., to an internal collision after Round 49. Since all the blocks which must be inserted in the following rounds are the same for both messages, we clearly obtain an internal collision after the computation of the hash value. The attack then requires fewer than $2^{232} \times 49 = 2^{237.5}$ calls to the round function. The memory complexity is $2^{239.5}$ bits. From the previous analysis, we deduce that the overall time complexity is $2^{240.5}$ operations, which is clearly less than for the generic collision attack when the length of the message digest is greater than or equal to 512. Then, Maraca with message digest of length 512 can be considered as broken.

4. Algebraic structure of $D_F(\delta)$ and generalized crooked functions

In the light of the previously described attack, it seems important to characterize the permutations F having some $\mathcal{D}_F(\delta)$ which coincide (or almost coincide) with a large affine subspace. A very particular case has been investigated in [**BdF98, vDdF00**] where the notion of crooked permutations have been introduced. Here, we recall this notion in the more general sense defined by Kyureghyan [**Kyu07**] which also includes the case where the function is not a permutation, and then where $\mathcal{I}m(D_\beta F)$ is a linear subspace of codimension 1.

Definition 4.1. [**BdF98, Kyu07**] A function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ is said to be *crooked* if, for any nonzero $\beta \in \mathbf{F}_2^n$, $\mathcal{I}m(D_\beta F)$ is a linear or affine subspace of codimension 1.

It is known that all crooked permutations are almost bent functions [**CC03**, Lemma 5], which are a particular case of APN functions depending on an odd number of variables. However, it is highly conjectured that the crooked functions exactly correspond to the quadratic APN functions. This has been proved in [**Kyu07**] in the case of monomial functions and in [**BK08**] in the case of binomials.

But, in our case, we are interested in the case where $\mathcal{D}_F(\delta)$ is an (affine) subspace but we do not require its codimension to be 1. This generalization then intends to capture some functions with a slightly larger differential uniformity, typically functions with $\Delta_F \leq 8$.

Definition 4.2. A function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ is said to be *crooked of codimension d* if, for any nonzero $\beta \in \mathbf{F}_2^n$, $\mathcal{I}m(D_\beta F)$ is an (affine) subspace of codimension d. In particular, crooked functions of codimension 1 correspond to the classical crooked functions as previously defined.

A weaker notion, which has been used in our attack against Maraca, corresponds to the situation where $\mathcal{I}m(D_\beta F^{-1})$ is not an (affine) subspace but is included in an (affine) subspace. Such situations are captured by the following weakened definition.

Definition 4.3. A function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ is said to be *weakly crooked of codimension* d, $d \geq 1$, if, for any nonzero $\beta \in \mathbf{F}_2^n$, $\mathcal{I}m(D_\beta F)$ is included in an affine subspace of codimension d.

For instance, all quadratic functions are weakly crooked of codimension d for some d. Obviously, any weakly crooked function of codimension d is also weakly crooked function of codimension d' for all $d' \leq d$. Then, the relevant parameter is the largest d such that F is weakly crooked function of codimension d. For instance, the inverse of the permutation P of $\mathbf{F}_2^8$ which is used in Maraca is weakly crooked of codimension 3.

It is worth noticing that, when F is a crooked (resp. weakly crooked) permutation of codimension d, all $\mathcal{I}m(D_\beta F)$ are (resp. are included in) affine subspaces, *i.e.*, cosets of linear subspaces.

(Weakly) crooked functions are obviously related to the functions whose components have some linear structures, in the sense of the following definition.

Definition 4.4. Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. An element $a \in \mathbf{F}_2^n$ is called *a linear structure for* F if $D_a F$ is constant. Clearly, the set of all linear structures for F is a linear space.

In the following, we define the components of a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ like in [**Nyb95**].

Definition 4.5. Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. The linear combinations of the coordinates of F are the Boolean functions

$$f_\lambda : x \in \mathbf{F}_2^n \mapsto \lambda \cdot F(x),\ \lambda \in \mathbf{F}_2^n,$$

where $x \cdot y$ denotes the usual dot product. The functions f_λ are called the *components* of F.

Proposition 4.6. *Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. Let a be a nonzero element in $\mathbf{F}_2^n$ and V a subspace of codimension d. Then,*

$$\mathcal{I}m(D_a F) \subset \gamma + V$$

for some $\gamma \in \mathbf{F}_2^n$ if and only if a is a linear structure of the components f_λ for all $\lambda \in V^\perp$. Moreover, for all λ in $V^\perp$, $D_a f_\lambda = \lambda \cdot \gamma$.

Proof. The result is directly deduced from the following fact.

$$\mathcal{I}m(D_a F) \subset \gamma + V$$

if and only if, for any $\lambda \in V^\perp$, we have

$$\lambda \cdot D_a F(x) = D_a f_\lambda(x) = \lambda \cdot \gamma,\ \forall x \in \mathbf{F}_2^n.$$

□

Kyureghyan proved [**Kyu07**, Corollary 6] that the linear space of any nonzero component of a monomial permutation is equal to $\{0\}$ except for quadratic permutations. We then deduce the following generalization of her result on the characterization of monomial crooked permutations.

Proposition 4.7. *A monomial permutation is weakly crooked of codimension d for some d if and only if it has degree 2.*

It is known [**CC03, Kyu07**] that all crooked permutations of codimension 1 are almost bent, that means that their Walsh coefficients

$$\sum_{x\in\mathbf{F}_2^n}(-1)^{f_\lambda(x)+\alpha\cdot x}$$

for all λ and α in $\mathbf{F}_2^n$ take three values only, $\pm 2^{\frac{n+1}{2}}$ and 0. This proof cannot be generalized directly to any codimension since it also involves the number of preimages x of all elements of $\mathcal{I}m(D_\beta F)$, *i.e.*, the number of x such that $D_\beta F(x)=\delta$ for all $\delta\in\mathcal{I}m(D_\beta F)$. This number is known to be 2 in the case of crooked functions of codimension 1, but the fact that all values in $\mathcal{I}m(D_\beta F)$ have the same number of preimages is only true if F has a two-valued differential spectrum. However, even if the complete Walsh spectrum of crooked functions of codimension d cannot be determined in the general case, a lower bound on its maximum value, *i.e.*, an upper bound on the nonlinearity, can be obtained.

Proposition 4.8. *Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. If F is weakly crooked of codimension d, then F has at least a component f_λ, $\lambda\neq 0$, which has a linear space of dimension greater than or equal to d, implying that the highest magnitude of its Walsh coefficients satisfies*

$$\mathcal{L}(F)\geq 2^{\frac{n+d}{2}}.$$

Proof. By hypothesis, for any nonzero $a\in\mathbf{F}_2^n$, there exists a subspace V_a of codimension d such that $\mathcal{I}m(D_aF)\subset\gamma_a+V_a$ for some $\gamma_a\in\mathbf{F}_2^n$. Proposition 4.6 then implies that a is a linear structure for all components f_λ, for $\lambda\in V_a^\perp$. Including the case $a=0$ which is a linear structure for all components, we deduce that

$$\#\{(\lambda,a)\in\mathbf{F}_2^n\times\mathbf{F}_2^n:\ D_af_\lambda=\text{cst}\}\geq 2^d(2^n-1)+2^n.$$

It follows that

$$\begin{aligned}(2^n-1)\max_{\lambda\neq 0}\#\{a\in\mathbf{F}_2^n:\ D_af_\lambda=\text{cst}\} &\geq \#\{(\lambda,a)\in\mathbf{F}_2^n\setminus\{0\}\times\mathbf{F}_2^n:\ D_af_\lambda=\text{cst}\}\\ &\geq 2^d(2^n-1).\end{aligned}$$

Since the set of linear structures is a linear space, there exists at least one component f_λ, $\lambda\neq 0$, which has a linear space of dimension greater than or equal to d. The lower bound on the highest magnitude of the Walsh coefficients of f_λ then follows from [**CCCF00**, Th. 3]. □

However, the question of the generalization of the conjecture on classical crooked function is an open problem.

Open problem 4.9. Does there exist any permutation F over $\mathbf{F}_2^n$ with $\deg(F)>2$ such that F is crooked of codimension d for some $d\geq 1$?

Finally, it must be noticed that our attack requires $\mathcal{D}_{F^{-1}}(\beta)$ to be (included in) an affine subspace for a single nonzero element $\beta\in\mathbf{F}_2^n$, not for all them. In the following, such functions are said to be (weakly) crooked of codimension d with respect to β. It is worth noticing that both notions are equivalent in the case of monomial functions (see Lemma 2.5).

Open problem 4.10. Characterize the permutations F over $\mathbf{F}_2^n$ such that, there exists a nonzero element $a\in\mathbf{F}_2^n$ for which $\mathcal{I}m(D_aF)$ is an affine subspace.

5. Conclusions

We have introduced a new quantity ∇_F, corresponding to the highest cardinality of the image sets of the derivatives of a function and we have pointed out, by a concrete attack against a recent hash function proposal, that the use of a permutation with a high ∇_F might introduce some weaknesses in a cryptographic primitive. Unfortunately, for any permutation, having a high ∇_F is a natural consequence of a good resistance to differential cryptanalysis. For instance, it appears that replacing the original permutation of Maraca by a commonly used Sbox like the inverse function increases its vulnerability. Moreover, our attack also points out that the situation where the image sets of the derivatives coincide (or almost coincide) with affine subspaces is the most favourable case for the attacker. Therefore, the use of crooked permutations (and of the generalizations we have introduced) must be avoided in the design a cryptographic primitive. On the other hand, we believe that our generalization of the notion of crooked functions may be helpful for solving the well-known open problem on the existence of crooked functions of degree greater than 2.

Acknowledgment

The authors would like to thank Pascale Charpin and Gohar Kyureghyan for many helpful discussions.

References

[BCC09] C. Blondeau, A. Canteaut, and P. Charpin, *Differential properties of power functions*, International Journal of Information and Coding Theory (2009), To appear.

[BdF98] T. Bending and D. Fon der Flass, *Crooked functions, bent functions, and distance regular graphs*, Electron. J. Combin. **5** (1998), no. 1, R34.

[BK08] J. Bierbrauer and G. Kyureghyan, *Crooked binomials*, Designs, Codes and Cryptography **46** (2008), no. 3, 269–301.

[BS91] E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology **4** (1991), no. 1, 3–72.

[CC03] A. Canteaut and P. Charpin, *Decomposing bent functions*, IEEE Transactions on Information Theory **49** (2003), no. 8, 2004–19.

[CCCF00] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine, *Propagation characteristics and correlation-immunity of highly nonlinear boolean functions*, Advances in Cryptology - EUROCRYPT'2000, Lecture Notes in Computer Science, vol. 1807, Springer-Verlag, 2000, pp. 507–522.

[CP02] N. Courtois and J. Pieprzyk, *Cryptanalysis of block ciphers with overdefined systems of equations*, Advances in Cryptology - ASIACRYPT'02, Lecture Notes in Computer Science, vol. 2501, Springer-Verlag, 2002, pp. 267–287.

[Dil09] J.F. Dillon, *APN polynomials: an update*, International Conference on Finite fields and applications - Fq9, 2009.

[HLL+00] S. Hong, S. Lee, J. Lim, J. Sung, D. Hyeon Cheon, and I. Cho, *Provable security against differential and linear cryptanalysis for the spn structure*, Fast Software Encryption - FSE 2000, Lecture Notes in Computer Science, vol. 1978, Springer, 2000, pp. 273–283.

[HP08] D. Hertel and A. Pott, *Two results on maximum nonlinear functions*, Designs, Codes and Cryptography **47** (2008), no. 1-3, 225–235.

[Jen08] R. J. Jenkins Jr., *Maraca - algorithm specification*, Submission to NIST, 2008.

[Kas71] T. Kasami, *The weight enumerators for several classes of subcodes of the second order binary Reed-Muller codes*, Information and Control **18** (1971), 369–394.

[Kyu07] G. Kyureghyan, *Crooked maps in* $\mathbb{F}_{2^n}$, Finite Fields and their applications **13** (2007), no. 3, 713–726.

[NK95] K. Nyberg and L.R. Knudsen, *Provable security against a differential attack*, Journal of Cryptology **8** (1995), no. 1, 27–37.

[Nyb93] K. Nyberg, *Differentially uniform mappings for cryptography*, Advances in Cryptology - EUROCRYPT'93, Lecture Notes in Computer Science, vol. 765, Springer-Verlag, 1993, pp. 55–64.

[Nyb95] ———, *S-boxes and round functions with controllable linearity and differential uniformity*, Fast Software Encryption - FSE'94, Lecture Notes in Computer Science, vol. 1008, Springer-Verlag, 1995, pp. 111–130.

[Röc08] A. Röck, *Stream ciphers using a random update function: Study of the entropy of the inner state*, Progress in Cryptology - AFRICACRYPT 2008, Lecture Notes in Computer Science, vol. 5023, Springer, 2008, pp. 258–275.

[vDdF00] E.R. van Dam and D. Fon der Flass, *Codes, graphs, and schemes from nonlinear functions*, Tech. report, Research memorandum, FEW 790, Tilburg University, The Netherlands, May 2000.

INRIA project-team SECRET, B.P. 105, 78153 Le Chesnay Cedex, France
E-mail address: Anne.Canteaut@inria.fr

INRIA project-team SECRET, B.P. 105, 78153 Le Chesnay Cedex, France
E-mail address: Maria.Naya_Plasencia@inria.fr