A new criterion for avoiding the propagation of linear relations through an Sbox

Christina Boura, Anne Canteaut

A new criterion for avoiding the propagation of linear relations through an Sbox*

Christina Boura[1,2] and Anne Canteaut[1]

[1] SECRET Project-Team - INRIA Paris-Rocquencourt - B.P. 105
78153 Le Chesnay Cedex - France
[2] Gemalto - 6, rue de la Verrerie - 92190 Meudon - France.
Christina.Boura@inria.fr, Anne.Canteaut@inria.fr

Abstract In several cryptographic primitives, Sboxes of small size are used to provide nonlinearity. After several iterations, all the output bits of the primitive are ideally supposed to depend in a nonlinear way on all of the input variables. However, in some cases, it is possible to find some output bits that depend in an affine way on a small number of input bits if the other input bits are fixed to a well-chosen value. Such situations are for example exploited in cube attacks or in attacks like the one presented by Fuhr against the hash function Hamsi. Here, we define a new property for nonlinear Sboxes, named (v, w)-linearity, which means that 2^w components of an Sbox are affine on all cosets of a v-dimensional subspace. This property is related to the generalization of the so-called Maiorana-McFarland construction for Boolean functions. We show that this concept quantifies the ability of an Sbox to propagate affine relations. As a proof of concept, we exploit this new notion for analyzing and slightly improving Fuhr's attack against Hamsi and we show that its success strongly depends on the (v, w)-linearity of the involved Sbox.

Keywords. Sbox, Boolean function, linear relations, Maiorana-McFarland construction, hash functions.

1 Introduction

In the construction of symmetric primitives such as block ciphers and hash functions, nonlinear functions are iterated to provide confusion. In particular, it is required that all the outputs of the primitive depend in a nonlinear way on the inputs. However, it might happen that some output bits can be expressed in an affine way as a function of a small number of input bits, when the other input bits are fixed to some well-chosen values. Clearly, the sizes of the corresponding sets of inputs and outputs provide a measure of the induced weaknesses: such a

* Partially supported by the French Agence Nationale de la Recherche through the BLOC project under Contract ANR-11-INS-011.

property always holds for any input set of size 1, but it should be avoided for larger sets. Actually, in such a situation, an attacker would be able to derive some conditional relations of algebraic degree 1 between some inputs and some outputs of the primitive and to exploit them in a cryptanalysis, like a cube attack [9] or an attack similar to the one presented by Fuhr on Hamsi [12]. However, it is often difficult to determine whether such affine relations exist and even more difficult to find them. Furthermore, from the designer's point of view, it is not easy to understand how such relations can be avoided at a low implementation cost, especially without increasing the number of rounds.

Our Contributions. In this paper, we show that the number of affine relations between input bits and output bits after several rounds of an SPN construction depends on a new linearity measure of the Sbox, that we call (v, w)-*linearity*. The parameters (v, w) quantify the ability of the Sbox to propagate affine relations. More precisely, a vectorial function S from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$ is (v, w)-linear, if there exist a subspace $V \subset \mathbf{F}_2^n$ with $\dim V = v$ and a subspace $W \subset \mathbf{F}_2^m$ with $\dim W = w$ such that all Boolean functions $x \mapsto \lambda \cdot S(x)$, for $\lambda \in W$, have degree at most 1 on all cosets of V. We show that the (v, w)-linear functions correspond to the functions which follow the generalized Maiorana-McFarland construction [18] applied to vectorial functions. In other words, the use of Sboxes obtained by this construction, which have been extensively studied for instance in [19,6,21,14], introduces some weaknesses into a cryptographic primitive, which might be exploited by a cube attack or by an attack like [12].

As a proof of concept, we analyze and slightly improve Fuhr's attack against Hamsi with the new insights brought by this notion. Most notably, we show that the feasibility of this attack mainly depends on the (v, w)-linearity of the Hamsi Sbox. By classifying 4-bit Sboxes in terms of (v, w)-linearity, we exhibit the families of Sboxes which considerably reduce the success of the attack.

The rest of the paper is organized as follows. In Section 2 we introduce the notion of (v, w)-linearity and present some general properties. We characterize, in this same section, the (v, w)-linear functions for certain values of (v, w) and we exhibit a classification of 4-bit Sboxes with respect to this new criterion. In Section 3, we recall the principle of the second preimage attack by Fuhr against the hash function Hamsi. Section 4 points out that the notion of (v, w)-linearity for the involved Sbox brings a new insight on Fuhr's attack. In particular, a more extensive use of this notion enables us to slightly improve the attack against Hamsi. We also investigate the feasibility of the attack for all possible choices of the 4-bit Sbox. We refer to [2] for further details, especially on the classification of 4-bit Sboxes and on the algorithms we used for finding affine relations for Hamsi-256.

2 The Notion of (v, w)-linearity

When we consider an Sbox, i.e., a vectorial function with several output coordinates, some of its cryptographic properties are derived from the properties of its components, in the following sense.

Definition 1. *[20] Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. The* components *of S are the linear combinations of its coordinates, i.e., the Boolean functions of n variables $S_\lambda : x \mapsto \lambda \cdot S(x)$, where $\lambda \in \mathbf{F}_2^m$ and S_0 is the null function.*

In the following, we often consider the *restriction* of S to an (affine) subspace $a+V$ of $\mathbf{F}_2^n$. This restriction corresponds to the function $x \in V \mapsto S(a+x)$, and it can be identified with a function of $\dim V$ variables.

2.1 Definition and Link with the Maiorana-McFarland Construction

Definition 2. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. Then, S is said to be* (v, w)-linear *if there exist two linear subspaces $V \subset \mathbf{F}_2^n$ and $W \subset \mathbf{F}_2^m$ with $\dim V = v$ and $\dim W = w$ such that, for all $\lambda \in W$, S_λ has degree at most 1 on all cosets of V.*

Obviously, a function S that is (v, w)-linear is equally (v, i)-linear for every $1 \leq i < w$. Similarly, it is (i, w)-linear for every $1 \leq i < v$.

Any Boolean function f which is linear on all cosets of a v-dimensional subspace V can be written as

$$f(x, y) = \pi(x) \cdot y + h(x) \quad \text{with } (x, y) \in U \times V,$$

where U is a supplementary subspace of V, π is a function from U to $\mathbf{F}_2^v$ and h is a Boolean function from U to $\mathbf{F}_2$. This construction is a well-known generalization of the so-called Maiorana-McFarland construction of bent functions [18]. This class has been generalized to vectorial functions in [19] and studied by several authors, e.g. [6,21,14]. Then, it follows that an Sbox is (v, w)-linear if and only if its 2^w components corresponding to W define a function which is equivalent to a vectorial Maiorana-McFarland function, in the sense of the following proposition.

Proposition 1. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$, and V and W two linear subspaces $V \subset \mathbf{F}_2^n$ and $W \subset \mathbf{F}_2^m$ with $\dim V = v$ and $\dim W = w$. Then, S is (v, w)-linear w.r.t. (V, W) if and only if the function S_W corresponding to all components $S_\lambda, \lambda \in W$, can be written as*

$$S_W(x, y) = M(x)y + G(x)$$

where $\mathbf{F}_2^n$ is the direct sum of U and V, G is a function from U to $\mathbf{F}_2^w$ and $M(x)$ is a $w \times v$ binary matrix whose coefficients are Boolean functions defined on U.

Proof. Let $(\lambda_1, \ldots, \lambda_w)$ be a basis of W. Clearly, S is (v, w)-linear w.r.t. (V, W) if and only if, for any $1 \leq i \leq w$, S_{λ_i} can be written as

$$S_{\lambda_i}(x, y) = \pi_i(x) \cdot y + g_i(x) \; .$$

Let G denote the function from U to $\mathbf{F}_2^w$ whose w coordinates correspond to g_i, $1 \leq i \leq w$, and let $M(x)$ denote the $w \times v$ matrix whose i-th row correspond to the w coordinates of $\pi_i(x)$. Then, the previous condition can be equivalently written as

$$S_W(x, y) = M(x)y + G(x) \ .$$

□

2.2 General Properties

It directly follows from the definition that (v, w)-linear functions have some weaknesses with respect to the usual cryptographic properties. In particular, the algebraic degree and the nonlinearity of some components of the Sbox both decrease when v increases. Indeed, an upper bound on the degree of the components of S can be directly deduced from Proposition 1.

Proposition 2. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. If S is (v, w)-linear w.r.t. (V, W), then all its components S_λ, $\lambda \in W$ have degree at most $n + 1 - v$.*

We now show that the (v, w)-linearity provides an upper bound on the nonlinearity of the function, i.e., on its distance to the set of all affine functions. The following notation will be extensively used. For any Boolean function f of n variables, we denote by $\mathcal{F}(f)$ the following value related to the Hamming weight of f:

$$\mathcal{F}(f) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)} = 2^n - 2wt(f) \ .$$

This quantity is just the discrete Fourier transform (aka., Walsh transform) at point 0 of the sign function $(-1)^f$.

Definition 3. *The* Walsh spectrum *of an Sbox S from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$ is the multiset*

$$\mathcal{W}(\mathcal{S}) = \{\mathcal{F}(S_\lambda + \varphi_\alpha), \alpha \in \mathbf{F}_2^n, \lambda \in \mathbf{F}_2^m \setminus \{0\}\} \ ,$$

where φ_α denotes the n-variable linear function $x \mapsto \alpha \cdot x$. The nonlinearity *of S is the Hamming distance between the set of its nontrivial components $\{S_\lambda, \lambda \neq 0\}$ and the set of all affine functions. It is given by*

$$2^{n-1} - \frac{1}{2}\mathcal{L}(S) \quad \textit{where} \quad \mathcal{L}(S) = \max_{\alpha \in \mathbf{F}_2^n, \lambda \neq 0} |\mathcal{F}(S_\lambda + \varphi_\alpha)| \ .$$

Proposition 3. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. If S is (v, w)-linear, then $\mathcal{L}(S) \geq 2^v$.*

Proof. The result comes from the fact that the linearity of a Boolean function f, $\mathcal{L}(f)$, is lower-bounded by the linearity of any of its restrictions to a subspace (see e.g. Corollary V.3 in [4]). Since the restriction of S_λ, $\lambda \in W$, to V is affine, it has linearity 2^v. □

The notion of (v, w)-linearity is also related to the notion of *normality* introduced by Dobbertin [11], and then generalized by Charpin [7] as follows: a Boolean function f of n variables is said to be *weakly v-normal*, if it is affine on an (affine) subspace V of dimension v. However, $(v, 1)$-linearity is a stronger requirement than weak v-normality since the component of S needs to have degree at most 1 on all cosets of V while weak normality requires this property on a single coset.

It is worth noticing that the two conditions derived from Propositions 2 and 3, i.e., $\deg f \leq n+1-v$ and $\mathcal{L}(f) \geq 2^v$ are not sufficient for guaranteeing that f is $(v, 1)$-linear. For instance, it has been shown in [5] that the Boolean function of 14 variables, $f(x) = \mathrm{Tr}(\alpha x^{57})$ with $\alpha \in \mathbf{F}_4 \setminus \mathbf{F}_2$, is not 7-weakly normal. Then, this function is not $(7, 1)$-linear while it has degree 4 and satisfies $\mathcal{L}(f) = 2^7$.

It is known that the Boolean functions which are affinely equivalent to a Maiorana-McFarland bent function can be characterized by their second-order derivatives [8]. The situation is similar for vectorial functions. In the following, we denote by $D_a S$ the derivative of a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$, i.e., $D_a S$ is the function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$ defined by $D_a S(x) = S(x + a) + S(x)$.

Proposition 4. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$. Then, S is (v, w)-linear w.r.t. (V, W) if and only if the function S_W corresponding to all components $S_\lambda, \lambda \in W$ is such that all its second-order derivatives, $D_\alpha D_\beta S_W$ with $\alpha, \beta \in V$ vanish.*

Proof. Let U denote a supplementary subspace of V.

- If S is (v, w)-linear w.r.t. (V, W), then for any $x \in U$,

  $$S_W(x, y) = M(x)y + G(x),$$

  where $M(x)$ is a $w \times v$ matrix and G a function from U to $\mathbf{F}_2^w$. It follows that, for any $\alpha, \beta \in V$, we have

  $$D_\alpha D_\beta S_W(x, y) = S_W(x, y) + S_W(x, y+\alpha) + S_W(x, y+\beta) + S_W(x, y+\alpha+\beta) = 0 .$$

- Conversely, if the second-order derivatives of S_W, $D_\alpha D_\beta S_W$ with $\alpha, \beta \in V$, vanish, then for any $x \in U$, the function F_x from V to $\mathbf{F}_2^w$ defined by $F_x(y) = S_W(x, y)$ is such that all its second-order derivatives vanish. However, if a function has degree at least 2, then it has at least one second-order derivative which does not vanish. It follows that, for any $x \in U$, S_W has degree at most 1 on $x + V$. □

2.3 $(v, 1)$-linear Functions

In the following, we focus on $(v, 1)$-linear functions since the highest value of v such that S is $(v, 1)$-linear is a relevant parameter. Actually, as seen in Propositions 2 and 3, this value provides bounds on the degree and on the nonlinearity of the corresponding component: $\deg f \leq n + 1 - v$ and $\mathcal{L}(f) \geq 2^v$. Obviously, any function is $(1, 1)$-linear. Then, we first consider $(2, 1)$-linear functions. From Proposition 4, a Boolean function is $(2, 1)$-linear if and only if one of its second-order derivatives vanishes. We now give a sufficient condition for this property.

Proposition 5. *Let f be a balanced Boolean function of n variables, n even, with $\deg(f) \leq 3$. Then f is $(2,1)$-linear.*

Proof. Since f is balanced, it is obviously not bent. Then, by definition, f has at least one derivative, say $D_\alpha f$, that is not balanced. Since $\deg(f) \leq 3$, we have that $\deg(D_\alpha f) \leq 2$. If $\deg(D_\alpha f) < 2$, then $D_\beta D_\alpha f$ vanishes for at least all values of β in a (affine) hyperplane. Thus, we deduce from Proposition 4 that f is $(2,1)$-linear. Suppose now that $\deg(D_\alpha f) = 2$ and consider its restriction to a hyperplane H such that $\alpha \notin H$. Let g denote this restriction, *i.e.* $g = D_\alpha f_{|H}$. This restriction is a quadratic function of $(n-1)$ variables that is not balanced (since its Hamming weight is half of the Hamming weight of $D_\alpha f$). Since n is even, $n-1$ is odd and thus g cannot be bent. Therefore, g has at least one derivative that is constant. That is, there exists some $\beta \in \mathbf{F}_2^n$ such that $D_\beta D_\alpha f$ is constant. Though, a quadratic function is balanced if and only if it has a derivative equal to 1. Therefore, $D_\beta D_\alpha f$ is the all-zero function. □

Most notably, it follows that all nontrivial components of a permutation of $\mathbf{F}_2^4$ are $(2,1)$-linear.

The other extremal case of $(n-1,1)$-linear Boolean functions can be completely characterized. Indeed, it can be shown that the necessary conditions on the degree and nonlinearity of an $(n-1,1)$-linear Boolean function (Propositions 2 and 3) are sufficient.

Proposition 6. *Let f be a Boolean function of n variables. Then, f is $(n-1,1)$-linear if and only if $\deg f \leq 2$ and $\mathcal{L}(f) \geq 2^{n-1}$. Moreover, if $\deg(f) = 2$ and $\mathcal{L}(f) \geq 2^{n-1}$, there exist exactly three distinct hyperplanes H such that f has degree at most 1 on both H and $\bar{H}$.*

Proof. The fact that any $(n-1,1)$-linear function has degree at most 2 and linearity greater than or equal to 2^{n-1} is derived from the previous propositions. Conversely, let us consider a quadratic Boolean function f (we assume that $\deg f = 2$ since the result is trivial for affine or constant functions). Any quadratic function f satisfies $\mathcal{L}(f) = 2^{\frac{n+h}{2}}$ where $0 \leq h < n$ is the dimension of the linear space of f, $LS(f)$ (see e.g. [4, Appendix 1]):

$$LS(f) = \{a \in \mathbf{F}_2^n \ : \ D_a f : x \mapsto f(x+a) + f(x) \text{ is constant}\} .$$

Moreover, the set

$$LS^0(f) = \{a \in \mathbf{F}_2^n \ : \ D_a f = 0\}$$

is a subspace of $LS(f)$ of dimension either $\dim LS(f)$ or $(\dim LS(f) - 1)$. Since $\mathcal{L}(f) = 2^{n-1}$, there are exactly 4 values of α such that $|\mathcal{F}(f + \varphi_\alpha)| = 2^{n-1}$, and exactly three among these four have the same sign. Now, we will prove that these four values are the elements of $\beta + LS(f)^\perp$, where $\beta = 0$ if $\dim LS^0(f) = \dim LS(f)$, and $\beta \in LS^0(f)^\perp \setminus LS(f)^\perp$ otherwise. We get from Lemma V.2 in [4] that

$$\sum_{\alpha \in LS(f)^\perp} \mathcal{F}^2(f + \varphi_{\alpha+\beta}) = 2^2 \sum_{e \in LS(f)} (-1)^{\beta \cdot e} \mathcal{F}(D_e f)$$

$$= 2^2 \left(\sum_{e \in LS^0(f)} \mathcal{F}(D_e f) - \sum_{e \in LS(f) \setminus LS^0(f)} \mathcal{F}(D_e f) \right) = 2^{2n} .$$

Therefore, all four $\mathcal{F}^2(f + \varphi_{\alpha+\beta})$, $\alpha \in LS(f)^\perp$, are equal to 2^{2n-2}. Now, since

$$2\mathcal{F}((f + \varphi_\beta)|_{H_a}) = \mathcal{F}(f + \varphi_\beta) + \mathcal{F}(f + \varphi_{\beta+a})$$

and

$$2\mathcal{F}((f + \varphi_\beta)|_{\bar{H}_a}) = \mathcal{F}(f + \varphi_\beta) - \mathcal{F}(f + \varphi_{\beta+a})$$

we deduce that f is linear both on H_a and $\bar{H}_a$ for some $a \neq 0$, if and only if there exist some u_1, u_2, u_3 such that

$$\mathcal{F}(f + \varphi_{u_1}) = \mathcal{F}(f + \varphi_{u_2}) = \mathcal{F}(f + \varphi_{u_3}) = (-1)^b 2^{n-1}$$

and

$$\mathcal{F}(f + \varphi_{u_1+u_2+u_3})(-1)^{b+1} 2^{n-1} .$$

Moreover, a can be any element in $\{u_1 + u_2, u_1 + u_3, u_2 + u_3\}$. Therefore, we get that f is linear both on H_a and $\bar{H}_a$ if and only if a is a nonzero element of $LS(f)^\perp$. □

If we focus on Sboxes which guarantee the best resistance to linear attacks, i.e., on permutations S of $\mathbf{F}_2^n$ with $\mathcal{L}(S) \leq 2^{\lceil \frac{n+1}{2} \rceil}$, then, for $n = 4$, we deduce from the previous propositions that any 4-bit permutation is $(2, 1)$-linear, and that it is $(3, 1)$-linear if and only if it has maximal nonlinearity and a quadratic component. For larger values of n, the situation is different. For instance, we can prove the following.

Corollary 1. *Let S be a permutation of $\mathbf{F}_2^n$ with the best known nonlinearity, that is $\mathcal{L}(S) \leq 2^{\lceil \frac{n+1}{2} \rceil}$. Then, if $n \geq 5$, S is not $(n-1, 1)$-linear.*

Proof. If S has a component that is $(n-1, 1)$-linear, then we deduce from Proposition 3 that

$$n - 1 \leq \left\lceil \frac{n+1}{2} \right\rceil \leq \frac{n}{2} + 1.$$

Consequently, $\frac{n}{2} \leq 2$ and thus $n \leq 4$. □

2.4 Classification of 4-bit Sboxes

Many symmetric primitives are based on 4-bit balanced Sboxes. Several classifications of these Sboxes have been previously provided. We can for example mention the classification by De Cannière [3], the one provided by Leander and Poschmann [17] and another one by Saarinen [22]. In particular in [17], the authors have proved that, for affine equivalence, there are exactly 16 classes of 4-bit permutations which are *optimal* in terms of resistance against both linear and differential attacks. Here, we go one step further in this classification, and

consider the notion of (v, w)-linearity for those 16 classes. Actually, the number of pairs (V, W) such that an Sbox is (v, w)-linear w.r.t (V, W) is invariant under affine equivalence.

The previous result shows that the number of quadratic components of the Sbox plays an important role for $(n-1, w)$-linearity. For instance, for a permutation of $\mathbf{F}_2^4$ which is optimal for linear cryptanalysis, we have proved that the number of pairs (V, W) with $\dim V = 3$ and $\dim W = 1$ such that S is $(3, 1)$-linear w.r.t to (V, W) is equal to $3Q$, where Q is the number of quadratic components of S. Therefore, we first focus on the number of quadratic components for a permutation of $\mathbf{F}_2^4$.

All classes of 5-variable Boolean functions for affine equivalence have been exhibited in [1]. From this classification, since the 4-variable Boolean functions can be seen as a subset of the functions in 5 variables, it can be deduced that any of the 2^{15} possible Boolean functions of four variables with degree at most 3 is equivalent to one of the five functions given in Table 1. This table also provides the corresponding Walsh spectra since affine equivalence preserves the multiset composed of the magnitudes of all Walsh coefficients, *i.e.* functions belonging to the same equivalence class, have the same multiset $\mathcal{W}(f) = \{|\mathcal{F}(f + \varphi_a)|,\ a \in \mathbf{F}_2^n\}$.

Table1. Number of occurrences of each value in the Walsh spectrum of any of the five equivalence classes for the 4-variable Boolean functions of degree at most 3.

		Walsh spectrum				
class	representative	±16	±12	±8	±4	0
I	$x_1x_2x_3$		1		7	8
II	$x_1x_2x_3 + x_1x_4$			2	8	6
III	x_1x_2			4		12
IV	$x_1x_2 + x_3x_4$				16	0
V	0	1				15

Proposition 7. *Let S be a permutation of $\mathbf{F}_2^4$ having no affine or constant component. Then, S has c_I components of the class I, c_{II} components of the class II and c_{III} components of the class III, with*

$$c_I + c_{II} + c_{III} = 15.$$

Moreover, the number Q of quadratic components of S (i.e., of components of degree exactly 2) is equal to c_{III} and is of the form $Q = 2^r - 1$, $0 \leq r \leq 4$. It is characterized by the Walsh spectrum of S (see Definition 3):

$$Q = W_{12} + \frac{1}{2}W_8 - 15\ ,$$

where W_i denotes the number of occurrences of i in $\mathcal{W}(S)$. Most notably, S and S^{-1} have the same number of quadratic components.

Proof. As S is a permutation, all of its components are of degree at most 3 and are equivalent to one of the five above classes. By hypothesis, as S does not have any constant or affine component, S has no component of the class V. Moreover, the number of components of degree 2 is equal to the number of components of degree at most 2. Similarly, as all the non-trivial components of a permutation are balanced, S has no component of the class IV. The number of non-trivial components of the permutation S is equal to 15. Therefore, $c_I + c_{II} + c_{III} = 15$. The class III corresponds to quadratic functions. Consequently, c_{III} represents the number of quadratic components Q of S. As the values of λ such that $\deg S_\lambda \leq 2$ form a vectorial subspace of $\mathbf{F}_2^4$, Q has the form $2^r - 1$. According to Table 1 we have that

$$\begin{aligned} W_{12} &= c_I \\ W_8 &= 2c_{II} + 4c_{III} = 30 - 2c_I + 2c_{III} \end{aligned}$$

implying that the number of quadratic components of S is given by $c_{III} = W_{12} + \frac{1}{2}W_8 - 15$. Finally, as the inverse permutation of S has the same set $(W_i)_{0 \leq i \leq 16}$ as S, the two permutations have the same number of quadratic components. □

We have carried out an exhaustive search among all the permutations of $\mathbf{F}_2^4$ in order to determine all possible 4-tuples $(c_I, c_{II}, c_{III}, c_V)$. All possible configurations can be found in Appendix A of [2]. Then, we have exhibited some permutations with $Q \in \{0, 1, 3, 7, 15\}$. But, permutations with $Q = 15$ satisfy $c_V = 1$, i.e., every quadratic permutation of $\mathbf{F}_2^4$ has one non-trivial component of degree 1. There exist permutations with 7 quadratic components and optimal nonlinearity, but they do not guarantee optimal resistance to differential attacks.

(v, w)-linearity of Optimal 4-bit Sboxes

We concentrate now on *optimal* permutations of $\mathbf{F}_2^4$, i.e., permutations which guarantee an optimal resistance against linear and differential attacks. The exhaustive search over all 16 classes of such Sboxes in [17] shows that there are 8 classes of optimal Sboxes with $Q = 0$, 4 with $Q = 1$, and 4 with $Q = 3$. For each of the 16 classes of optimal Sboxes, Table 2 gives, for each pair (v, w), the number $N_{(v,w)}$ of subspaces V such that the Sbox is (v, w)-linear w.r.t. (V, W).

Since all optimal Sboxes have at most 3 quadratic components, we deduce from Proposition 6 that they cannot be $(3, 3)$-linear, i.e. $N_{(3,3)} = 0$.

The fact that, for all these Sboxes, $N_{(2,1)} = 35$ comes from the following result.

Proposition 8. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ of degree at most 3. Then, for any pair (a, b) of elements in $\mathbf{F}_2^n$, there exists some nonzero $\lambda \in \mathbf{F}_2^n$ such that $D_a D_b S_\lambda = 0$.*

Equivalently, for any 2-dimensional subspace $V \subset \mathbf{F}_2^n$, there exists at least one nonzero $\lambda \in \mathbf{F}_2^n$ such that S is $(2, 1)$-linear w.r.t $(V, \{0, \lambda\})$.

Table2. Number $N_{(v,w)}$ of subspaces V of dimension v for which there exists a w-dimensional W such that G_i is (v,w)-linear with respect to (V,W), for the 16 optimal Sboxes G_i described in [17].

		(v,w)							
	Q	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)
G_0	3	35	19	5	0	7	1	0	0
G_1	3	35	23	3	0	7	1	0	0
G_2	3	35	23	3	0	7	1	0	0
G_3	0	35	5	0	0	0	0	0	0
G_4	0	35	5	0	0	0	0	0	0
G_5	0	35	5	0	0	0	0	0	0
G_6	0	35	5	0	0	0	0	0	0
G_7	0	35	5	0	0	0	0	0	0
G_8	3	35	19	5	0	7	1	0	0
G_9	1	35	13	0	0	3	0	0	0
G_{10}	1	35	13	0	0	3	0	0	0
G_{11}	0	35	5	0	0	0	0	0	0
G_{12}	0	35	5	0	0	0	0	0	0
G_{13}	0	35	5	0	0	0	0	0	0
G_{14}	1	35	13	0	0	3	0	0	0
G_{15}	1	35	11	1	0	3	0	0	0

Proof. The first statement is proved by contradiction as follows. Suppose that there exists a pair (a,b) such that $D_aD_bS_\lambda \neq 0$ for all $\lambda \neq 0$. This situation can occur only if $\langle a,b\rangle$ has dimension 2. Obviously, all the (2^n-1) functions $D_aD_bS_\lambda$, $\lambda \neq 0$, are distinct since $D_aD_bS_{\lambda_1} + D_aD_bS_{\lambda_2} = D_aD_bS_{\lambda_1+\lambda_2}$. Let U be a supplementary subspace of $\langle a,b\rangle$. Then, the whole function $D_aD_bS_\lambda$ is determined by its restriction to U since $D_aD_bS_\lambda(x) = D_aD_bS_\lambda(x+v)$ for any $v \in \langle a,b\rangle$. Then, because $\deg D_aD_bS_\lambda \leq 1$, the number of distinct and nonzero $D_aD_bS_\lambda$ corresponds to the number of nonzero affine functions of $(n-2)$ variables, which is equal to $(2^{n-1}-1)$. This leads to a contradiction since the (2^n-1) functions $D_aD_bS_\lambda$ are all distinct. The equivalent formulation in terms of $(2,1)$-linearity is a direct consequence of Proposition 4. □

The next proposition explains why $N_{(2,3)} = 0$ when $Q = 0$.

Proposition 9. *Let S be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ such that all its non trivial components have degree exactly $(n-1)$. Then, S is not $(2,n-1)$-linear.*

Proof. Suppose that there exist a hyperplane H and two nonzero distinct elements a and b in $\mathbf{F}_2^n$ such that $D_aD_bS_\lambda = 0$ for all $\lambda \in H$. Let L be a linear permutation which maps a and b to the first two elements of the canonical basis e_1 and e_2. Then, $D_aD_bS(x) = D_{e_1}D_{e_2}(S \circ L^{-1})(L(x))$, implying that $D_{e_1}D_{e_2}(S \circ L^{-1})_\lambda = 0$ for all $\lambda \in H$. Let $\mathcal{M}$ denote the set of all monomials of degree $(n-1)$ of n variables whose second derivative with respect to e_1 and

e_2 vanishes. Then, $|\mathcal{M}| = n-2$. Since all $(S \circ L^{-1})_\lambda$, $\lambda \in H \setminus \{0\}$ have degree $(n-1)$, all their ANF contain a sum of monomials of $\mathcal{M}$, and all these $(2^{n-1}-1)$ sums must be distinct. However, this situation cannot occur since there are only $2^{|\mathcal{M}|}-1 = 2^{n-2}-1$ such sums. □

Moreover, a counting argument shows that for 4-bit permutations with optimal nonlinearity,

$$N_{(2,2)}+2N_{(2,3)}+4N_{(2,4)} = 5+8Q \text{ and } N_{(3,1)}+2N_{(3,2)}+4N_{(3,3)}+8N_{(3,4)} = 3Q \ .$$

Indeed, let us denote by A_w (resp. B_w) the number of subspaces V of dimension 2 (resp. dimension 3) such that w is the highest dimension such that S is (v,w)-linear w.r.t. (V,W) for some W of dimension w. Then,

$$N_{(2,i)} = \sum_{w=i}^{4} A_w \text{ and } N_{(3,i)} = \sum_{w=i}^{4} B_w \ .$$

On the other hand, if S_λ is quadratic, it belongs to Class III identified in Table 1, implying that it is $(2,1)$-linear w.r.t. 19 subspaces of dimension 2, and $(3,1)$-linear w.r.t. 3 hyperplanes. If S_λ has degree 3, then it belongs to Class II, and has three zero second-order derivatives. Then,

$$A_1+3A_2+7A_3+15A_4 = 3(15-Q)+19Q \text{ and } B_1+3B_2+7B_3+15B_4 = 3Q \ .$$

Since $N_{(2,1)} = 35$ from Proposition 8, we deduce that

$$35+2N_{(2,2)}+4N_{(2,3)}+8N_{(2,4)} = 45+16Q$$

and

$$N_{(3,1)}+2N_{(3,2)}+4N_{(3,3)}+8N_{(3,4)} = 3Q \ .$$

It is also worth noticing that $N_{(3,2)} \in \{0,1,3\}$. Actually, we have proved in Proposition 6 that S is $(3,2)$-linear w.r.t. $(H_a, \langle \lambda_1, \lambda_2 \rangle)$ if and only if a belongs to all three sets $LS(S_\lambda)^\perp$, $\lambda \in \{\lambda_1, \lambda_2, \lambda_1+\lambda_2\}$. Therefore, either all these three $LS(S_\lambda)^\perp$ are distinct, or they share one nonzero element or they are all equal.

From these results, we can deduce the values of $N_{(v,w)}$ in most cases for all 4-bit optimal Sboxes. All these values are provided in Table 2. In particular, all figures for $Q \in \{0,1\}$ can be deduced from the previous propositions. For $Q \geq 3$, the weighted sum of $N_{(3,1)}$ and $N_{(3,2)}$ (resp. of $N_{(2,2)}$, $N_{(2,3)}$ and $N_{(2,4)}$) can be explained theoretically, but a theoretical explanation of their exact individual values remains open. Most notably, Table 2 shows that there are five different behaviours of 4-bit optimal Sboxes with respect to (v,w)-linearity. It is worth noticing here that an Sbox and its inverse do not always have the same behaviour. Indeed, as pointed out in [17], any optimal Sbox G_i belongs to the same equivalence class as its inverse except G_0, G_2, G_{14} and G_{15} which are such that G_0^{-1} belongs to the same class as G_2 and G_{14}^{-1} belongs to the same class as G_{15}. Then, we deduce that, for all Sboxes S in the four classes defined by G_0, G_2, G_{14} and G_{15}, S and S^{-1} do not have the same behaviour regarding (v,w)-linearity.

3 Fuhr's Attack against Hamsi-256

The hash family Hamsi was designed by Küçük [15] in 2008 for the SHA-3 competition. It was among the 14 algorithms that were chosen by the NIST for the second round of the contest. A special feature of this function is that its compression function consists of a small number of rounds of a permutation with a particularly low algebraic degree. These weaknesses have been exploited by Fuhr [12] and by Dinur and Shamir [10] in order to find second preimages for the entire hash function. We show here that Fuhr's attack is related to the (v, w)-linearity of the Sbox used in Hamsi. More precisely, we use this notion for formalizing an important part of the attack in [12], that is the search for affine relations between some input and output bits of the compression function of Hamsi-256. This enables us to slightly improve Fuhr's result and to analyse the influence of the choice of the Sbox on this type of attack.

3.1 Description of Hamsi-256

We start by describing the most important parts of the design of Hamsi-256, the instance of the hash function outputting 256-bit digests. The Hamsi hash function follows the Davies-Meyer construction. In Hamsi-256, the message is padded and cut into 32-bit blocks. A linear code over $\mathbf{F}_4$ is used to expand each 32-bit message block to a 256-bit value $(m_0, \ldots, m_7)$, where every m_i is a 32-bit word. Then, the 256-bit expanded message is combined together with the 256-bit chaining value h_{i-1} and provides a 512-bit state. The inner permutation P is then applied to this 512-bit state, seen as a 4×4 matrix of 32-bit words.

Concatenation: The chaining value $(c_0, \ldots, c_7)$ is concatenated to the message words $(m_0, \ldots, m_7)$ to form a 512-bit state $s = (s_0, \ldots, s_{15})$, seen as a 4×4 matrix. The state s as also the way that the message and the chaining value words are arranged within it are illustrated in Figure 1.

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	s_{15}

m_0	m_1	c_0	c_1
c_2	c_3	m_2	m_3
m_4	m_5	c_4	c_5
c_6	c_7	m_6	m_7

Figure1. Input state of the inner permutation P in Hamsi-256.

The nonlinear permutation P of $\mathbf{F}_2^{512}$ is then applied to this concatenated state. It is composed of three rounds of a permutation R, called the round function. This round function is made up of three different layers of operations. First, some constant values are added to the state. Then, a nonlinear layer corresponding to 128 parallel applications of a 4-bit Sbox S is applied. Finally, the bits of the state are mixed by a linear application L.

The *substitution layer* is based on a 4-bit Sbox S. S is one of the Sboxes used in `Serpent` and is given by

$$S[16] = \{8, 6, 7, 9, 3, 12, 10, 15, 13, 1, 14, 4, 0, 11, 5, 2\}.$$

The algebraic normal forms of its coordinates are

$$\begin{aligned}
y_0 &= x_0x_2 + x_1 + x_2 + x_3 \\
y_1 &= x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2 + x_0x_3 + x_2x_3 + x_0 + x_1 + x_2 \\
y_2 &= x_0x_1x_3 + x_0x_2x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_0 + x_1 + x_3 \\
y_3 &= x_0x_1x_2 + x_1x_3 + x_0 + x_1 + x_2 + 1.
\end{aligned}$$

This Sbox is applied in parallel to the 128 columns of the state. In the first round, due to the way that the state is obtained by concatenation, every Sbox mixes two message bits with two bits coming from the chaining value.

The *diffusion layer* of Hamsi-256 is based on the linear function $L : \mathbf{F}_2^{128} \to \mathbf{F}_2^{128}$ that operates on 32-bit words. In the case of Hamsi-256, this function is called four times in total in every round, one time for every diagonal of the state.

The function $L(a, b, c, d)$, with $a, b, c, d \in \mathbf{F}_2^{32}$ can be described as follows:

$$\begin{array}{l|l}
a := a \lll 13 & d := (d \oplus c \oplus (a \ll 3)) \lll 7 \\
c := c \lll 3 & a := (a \oplus b \oplus d) \lll 5 \\
b := (b \oplus a \oplus c) \lll 1 & c := (c \oplus d \oplus (b \ll 7)) \lll 22
\end{array}$$

Truncation and Feed-forward: The truncation $T : \mathbf{F}_2^{512} \to \mathbf{F}_2^{256}$ eliminates the second and the last row of the state:

$$T(s_0, s_1, s_2, \ldots, s_{14}, s_{15}) = (s_0, s_1, s_2, s_3, s_8, s_9, s_{10}, s_{11}).$$

The truncated 256-bit state is then XORed to the previous chaining value h_{i-1} to form h_i.

Notations: Table 3 describes how we have numbered the 512 bits of the state. According to the representation of the Hamsi state seen in Figure 1, we will say that the bit 0 of the state, is the leftmost bit of s_0, 31 is the rightmost bit of s_0, 32 the leftmost bit of s_1, 128 the leftmost bit of s_4, etc.

3.2 Description of Fuhr's Attack

Fuhr described in [12] a method for finding second preimages for Hamsi-256. This cryptanalysis, of complexity equal to $2^{251.3}$ evaluations of the compression

0 31	32. 63	64. 95	96. 127
128 . . . 159	160 . . . 191	192 . . . 223	224 . . . 255
256 . . . 287	288 . . . 319	320 . . . 351	352 . . . 383
384 . . . 415	416 . . . 447	448 . . . 479	480 . . . 511

Table3. Enumeration of the bits of the state.

function, was the first attack on this candidate that had a lower complexity than the generic attack when treating small messages. The key idea in this cryptanalysis consists in finding affine relations between some input bits and some output bits of the compression function, when the other input bits are fixed to a constant value. These relations lead to preimages for the compression function of Hamsi-256. These pseudo-preimages for the hash function are then transformed into second preimages by using a meet-in-the-middle approach.

In order to find affine relations between some input and output bits of the compression function, Fuhr noticed that, for the Hamsi Sbox S,

$$S(1, x, 0, 1 + x) = (1, 0, 0, x), \text{ for every } x \in \mathbf{F}_2, \tag{1}$$

where the least significant bit is the leftmost bit.

With this property in mind, it is possible to choose a set of variables in the following way. If $y \in \mathbf{F}_2^{32}$, we will denote by y^j the j-th bit of y. If the message block after the constant addition in the first round is such that $s_0^j = 1$ and $s_8^j = 0$, then we can define a variable bit $x^j \in \mathbf{F}_2$ and set $s_4^j = x^j$ and $s_{12}^j = 1 + x^j$. Due to relation (1), after the application of the first nonlinear layer, only s_{12}^j will depend on x^j. This has a particular interest as s_{12}^j will be part of the input word d of the linear function L, which has a relatively slow diffusion, much slower than the words a or c. The same applies for the neighboring column of the state, i.e. the words s_1, s_5, s_9, s_{13}. If $s_1^j = 1$ and $s_9^j = 0$, we define the variable bit $y^j \in \mathbf{F}_2$ and set $s_5^j = y^j$ and $s_{13}^j = 1 + y^j$.

For mounting the attack, a message block is randomly picked. The first step is to choose the set of variables $\mathcal{I} = X \cup Y$. For this, the values of s_0, s_1, s_8 and s_9 before the first Sbox layer are computed. If $s_0^j = 1$ and $s_8^j = 0$ then the variable x^j is added to X. In the same way, if $s_1^j = 1$ and $s_9^j = 0$, the variable y^j is added to Y. Once the variable set has been chosen, one has to search for a set of output bits of the compression function $\mathcal{O}$ such that each bit of this set can be expressed as an affine function of the variables of $\mathcal{I}$.

Suppose that such a set has been found and denote $N_{\mathcal{O}} = \#\mathcal{O}$ and $N_{\mathcal{I}} = \#\mathcal{I}$. Let $x_0, \ldots, x_{N_{\mathcal{I}}-1}$ be the elements of $\mathcal{I}$ and $z_0, \ldots, z_{N_{\mathcal{O}}-1}$ the elements of $\mathcal{O}$. Then, if we are given a chaining value h^* it is possible to find preimages for the compression function, i.e. a message block m and a chaining value h, such that $f(h, m) = h^*$, where f is the compression function of Hamsi-256, with the following simple algorithm, described in [12].

1. Choose a message m such that the conditions required by Equation (1) for the positions indicated by the variables of $\mathcal{I}$ are satisfied.
2. Choose a chaining value h such that the conditions required by Equation (1) for the positions indicated by the variables of $\mathcal{I}$ are satisfied.
3. Compute the bits $z_0, \ldots, z_{N_\mathcal{O}-1}$. Compute the coefficients of the affine system.
4. Solve the affine system. If the system has no solution then choose other values for the constant part of h (without modifying the part of h imposed by the conditions (1)) and go to Step 3. If there is still no solution, choose another message m that fulfills the same constraints and go to Step 2.
5. If the affine system has a solution, check whether $f(h, m) = h^*$. This equation has a solution with probability $2^{N_\mathcal{O}-256}$.

The overall complexity of the attack, corresponding to $2^{251.3}$ evaluations of the compression function, has been estimated in [12] by a very precise estimation of the number of binary operations performed during each step of the algorithm.

Searching for Affine Relations for the Compression Function A very important part of the attack in [12] is the search for affine relations between some input and some output bits of the compression function of Hamsi-256.

Due to Relation (1), after one round of computation all the bits of the state depend affinely on the variable bits. However, this is not the case after the second and the third round of the computation, since the initial variables pass through the Sboxes of the last two rounds. Under some conditions though, some output bits of an Sbox can still be expressed as a linear combination of the input variables. The conditions identified in [12] are the following.

1. All but one input bits of the Sbox are constant. If this bit is some affine combination of the initial variables, then this will also be the case for all the four outputs of the Sbox.
2. If all the inputs of the Sbox depend on at most one initial variable, then all the output bits of the Sbox will depend affinely on this variable.
3. If none of the first two situations occurs, this means that there exist at least two inputs to the Sbox that depend in an affine way on at least two different variables. However, by looking at the ANF of the four outputs of the Sbox, it is possible to do the following two observations. The only nonlinear term of the first output bit y_0 is x_0x_2. Thus if this term is an affine combination of the initial variables, this will also be the case for y_0. Equally, if $x_0x_1x_2$ and x_1x_3 are affine in the initial variables, this will also be the case for y_3.

These properties were used by Fuhr in the search for a set of variable bits $\mathcal{I}$ and a set of output bits $\mathcal{O}$ which affinely depend on the variable bits $\mathcal{I}$. In his first paper [12], the number of variable bits $N_\mathcal{I}$ was fixed and then an automated search was launched in order to determine the variable set that would give the largest number $N_\mathcal{O}$ of such output bits. These results could then be used in order to generate the largest possible set of affine relations. By using this method, Fuhr

found for some $\mathcal{I}$ of size $N_{\mathcal{I}} = 7$, $N_{\mathcal{O}} = 14$ affine equations in $\mathcal{I}$ and for $N_{\mathcal{I}} = 8, 11$ affine equations for the compression function. Later, in [13] he improved these results, by finding for $N_{\mathcal{I}} = 8$, 16 affine equations and for $N_{\mathcal{I}} = 9$, 11 affine equations.

4 Analysis and Improvement of Fuhr's Attack

We show in this section how to make the search for affine relations between the input and the output bits of the compression function more efficient. Besides the improvement on Hamsi, our approach can similarly be applied to the search for affine relations for any SPN construction using small Sboxes. The success of this part of the work depends, to a large extend, on the quality of the used Sboxes. Our improvements are based on two different directions. The first one concerns the way that the propagation through the Sboxes of the second and the third round is treated. For this, we use the concept introduced in Section 2.

The second direction is related to the way we determine which Sboxes of the first round should be affected and how. Furthermore, another differential property of the Hamsi Sbox is used together with Relation (1) to go through the Sbox layer of the first round.

4.1 Propagation of Affine Relations through the Hamsi Sbox

Let $x = (x_0, x_1, x_2, x_3)$ denote the input to an Sbox and $y = (y_0, y_1, y_2, y_3)$ its output. As described in Section 3.2, Fuhr exploited the following two algebraic properties of the Hamsi Sbox in order to treat the case when at least two input variables of an Sbox are affected by at least two different variables in the second and third round.

- y_0 has degree at most 1 if x_0x_2 has degree at most 1.
- y_3 has degree at most 1 if x_1x_3 and $x_0x_1x_2$ have degree at most 1.

These two properties can be reformulated in the following way (where each vector x of $\mathbf{F}_2^4$ is represented by the integer $(\sum_{i=0}^{3} x_i 2^i)$).

- S_1 is $(3,1)$-linear w.r.t. $(H_\alpha, \langle 1\rangle)$ where H_α denotes the hyperplane $\langle \alpha\rangle^\perp$ for $\alpha \in \{1, 4, 5\}$.
- S_8 is $(2,1)$-linear w.r.t. $(V, \langle 8\rangle)$ for the three 2-dimensional subspaces $V = \langle 1, 8\rangle$, $V = \langle 4, 8\rangle$ and $V = \langle 5, 8\rangle$.

With the notation used in [17] and in Table 2, the Hamsi Sbox is affinely equivalent to G_1. Therefore, there exist 23 subspaces V of dimension 2 for which the Sbox is $(2,2)$-linear and 3 subspaces of dimension 2 on which it is $(2,3)$-linear. For the Hamsi Sbox, all corresponding pairs (V, W) can be deduced from Table 4.

From this table, we can check that, for $\lambda = 1$ (resp. for $\lambda = 8$), the properties given by Fuhr describe the whole list of subspaces V such that S is $(3,1)$-linear

V	list of λ	V	list of λ	V	list of λ	V	list of λ
$\langle 1,2\rangle$	$\{1,e,f\}$	$\langle 2,8\rangle$	$\{1,e,f\}$	$\langle 3,d\rangle$	$\{3,c,f\}$	$\langle 6,8\rangle$	$\{1,4,5,a,b,e,f\}$
$\langle 1,4\rangle$	$\{e\}$	$\langle 2,9\rangle$	$\{1,e,f\}$	$\langle 4,8\rangle$	$\{1,6,7,8,9,e,f\}$	$\langle 6,9\rangle$	$\{4,a,e\}$
$\langle 1,6\rangle$	$\{4,a,e\}$	$\langle 2,c\rangle$	$\{1,e,f\}$	$\langle 4,9\rangle$	$\{e\}$	$\langle 6,a\rangle$	$\{1,e,f\}$
$\langle 1,8\rangle$	$\{1,8,9\}$	$\langle 2,d\rangle$	$\{1,e,f\}$	$\langle 4,a\rangle$	$\{1,2,3,c,d,e,f\}$	$\langle 6,b\rangle$	$\{5,b,e\}$
$\langle 1,a\rangle$	$\{1\}$	$\langle 3,4\rangle$	$\{e\}$	$\langle 4,b\rangle$	$\{e\}$	$\langle 7,8\rangle$	$\{1,6,7\}$
$\langle 1,c\rangle$	$\{f\}$	$\langle 3,5\rangle$	$\{5,b,e\}$	$\langle 5,8\rangle$	$\{1,8,9\}$	$\langle 7,9\rangle$	$\{3,e,f\}$
$\langle 1,e\rangle$	$\{2,d,f\}$	$\langle 3,8\rangle$	$\{1,6,7\}$	$\langle 5,9\rangle$	$\{f\}$	$\langle 7,a\rangle$	$\{1\}$
$\langle 2,4\rangle$	$\{1,e,f\}$	$\langle 3,9\rangle$	$\{1\}$	$\langle 5,a\rangle$	$\{1\}$	$\langle 7,b\rangle$	$\{f\}$
$\langle 2,5\rangle$	$\{1,e,f\}$	$\langle 3,c\rangle$	$\{f\}$	$\langle 5,b\rangle$	$\{2,d,f\}$		

Table4. List of all $\lambda \in \mathbf{F}_2^4$ such that S is $(2,1)$-linear w.r.t. $(V, \langle\lambda\rangle)$, for each subspace V of dimension 2.

(resp. $(2,1)$-linear) w.r.t. to $(V, \langle\lambda\rangle)$. Nevertheless, it appears that S is also $(3,2)$-linear, and $(2,2)$-linear with respect to many other subspaces. In particular, we can see that it is possible to identify other components of S which have also degree at most 1 on the same subspaces. This is very useful in practice, as by using this table we can now guarantee the affine propagation of some components of S that we would have rejected before. For example we can observe that y_1 and y_2 are $(2,1)$-linear with respect to three different subspaces of dimension 2 each. These cases that are not treated at all in [12] can now be used to search for a possible affine propagation of the initial variables through the second and the third round.

4.2 Searching for the Input Variables

In [12], Relation (1) is used in order to ensure the affine propagation through the nonlinear layer of the first round. As we have already mentioned, this property guarantees that after the Sbox layer of the first round, there is at most one variable per *active* Sbox. We name *active* an Sbox that takes at least one variable as input. In the contrary, we call an Sbox *non-active* if its input vector is constant. Moreover, Relation (1) ensures that this unique variable belongs to a word corresponding to the d-input of the linear function L (see Section 3.1). It is easy to see from the description of L that the variables that belong to a word d of the state propagate much slower than the variables in the words a and c. In particular, each variable of a word d affects at most three bits of the state after the application of the linear part. However, the variables of the words b have the same slow propagation as the words d and this property was not exploited in [12]. In this sense, the following property of the Hamsi Sbox appears to be very useful:

$$S(1,x,0,x) = (0,x,1,0), \text{ for every } x \in \mathbf{F}_2. \tag{2}$$

Our aim is to find a set of input variables $\mathcal{I}$ such that the set of output bits $\mathcal{O}$ that are affine in $\mathcal{I}$, is maximized. Then, the most difficult problem is to choose

which Sboxes of the state during the first round will be active. We have used the following approach to solve this problem.

First, we restrict the search to the first 64 Sboxes of the state for the following reason. Equally with the approach in [12], we are searching for a preimage h of a given chaining value h^*. This is why the chosen variable bits of the internal state must be assigned to positions that, after the concatenation, contain variables coming out from the chaining value. By using Relation (1) or Relation (2), this constraint is verified for the first half of the state. On the contrary, this does not hold anymore for the second half, because the positions of the message bits and the chaining value bits are interchanged.

However, it is obvious that we cannot test all the possible pairs $(\mathcal{I}, \mathcal{O})$ because of the high complexity of such a search. For this reason we have adopted a heuristic strategy, that can be found in Appendix B of [2]. This heuristic method exploits the low diffusion through the three rounds of the function for finding good candidates for the input and output sets. An algorithm for obtaining such candidate sets is equally described in [2] (Algorithm 1). Once such candidate sets have been obtained, we launch an automated search, to see which combination of $N_{\mathcal{I}}$ of the input bits in the candidate set gives the largest number of affine output bits. For each test, we check the propagation through the last two rounds by using the relations identified by Table 4. These techniques have led to the following results.

4.3 Results

For $N_{\mathcal{I}} = 9$ input variables. For the 9 Sboxes $\{0, 7, 24, 30, 35, 37, 51, 59, 61\}$, we are fixing the inputs as required by Relation (1) for the Sboxes $\{0, 30, 35, 37\}$ and by Relation (2) for the others. Then the 13 output bits

$$\{6, 8, 43, 78, 262, 278, 313, 320, 343, 345, 350, 355, 380\}$$

depend in an affine way on the 9 input variables. In particular, we are able to find two more affine relations than in [12] for $N_{\mathcal{I}} = 9$ variables.

For $N_{\mathcal{I}} = 10$ input variables. For the 10 Sboxes $\{0, 7, 12, 16, 30, 35, 37, 51, 59, 61\}$, we are fixing the inputs as required by Relation (1) for the Sboxes $\{0, 16, 30, 35, 37\}$ and by Relation (2) for the others. Then the 11 output bits

$$\{6, 8, 43, 78, 278, 313, 320, 343, 345, 350, 380\}$$

depend affinely on the 10 input variables. Here again we find two more output bits than Fuhr in [12].

As we were able to find in both cases a higher number of affine equations than those of the original paper, the overall complexity of the attack should slightly decrease. However a complete complexity evaluation of our attack is a very complex task since it requires to count down the performed number of bitwise operations during all the steps of the attack. This procedure exceeds the scope of this work.

5 Conclusions

We have introduced a new cryptographic property for vectorial Boolean functions, that we call the (v, w)-linearity. This notion can be used as a new measure of linearity for Sboxes and is related to the number of linear relations that propagate through them. As the 4-bit balanced Sboxes are among the most used building-blocks in symmetric primitives, we classify them according to this new criterion. In particular, we analyse the (v, w)-linearity of "optimal" 4-bit permutations, according to the classification of Leander and Poschmann in [17].

For instance, our analysis points out that the Sbox used in Hamsi does not guarantee the best resistance to Fuhr's attack. Indeed, if an Sbox belonging to one of the classes G_3, G_4, G_5, G_6, G_7, G_{11}, G_{12} or G_{13} was used, the good linear and differential properties of the Sbox would still be preserved, but the function would be (v, w)-linear for a smaller value of w. In other words, the Sbox would have fewer components which may remain affine with respect to the input variables. Moreover, the number of 2-dimensional subspaces V such that S is $(2, w)$-linear w.r.t. (V, W) for some W is quite large. This increases the degrees of freedom in the cryptanalysis introduced by Fuhr, while the attack would probably have failed for an Sbox without any quadratic component. In order to verify this in practice, we implemented the same attack on the variant of Hamsi based on some other Sbox. More precisely, we first used the representative Sbox of the class G_3, as this is given in [17] and then, the Sbox S_0 of the finalist of the SHA-3 competition, JH [23]. Indeed, we noticed that in both cases Fuhr's attack failed.

A future line of work would be to determine how the new notion of (v, w)-linearity is related to some other recent attacks. For instance, the invariant subspace attack [16] exploits a similar but stronger property of the 3×3 Sbox used in PRINTcipher: two outputs of this Sbox are constant on a subspace of dimension 1 and on all its cosets (the coset is here determined by the key). Some relation to the resistance to first-order DPA could also be investigated.

Acknowledgments.

We would like to thank María Naya Plasencia for her valuable advices, and Christian Rechberger for very interesting discussions.

References

1. E.R. Berlekamp and L.R. Welch. Weight distributions of the cosets of the (32,6) Reed-Muller code. *IEEE Transactions on Information Theory*, 18(1):203–207, 1972.
2. C. Boura and A. Canteaut. A new criterion for avoiding the propagation of linear relations through an Sbox (Full version). IACR ePrint Report 2013/211, April 2013. `http://eprint.iacr.org/2013/211`.
3. C. De Cannière. *Analysis and Design of Symmetric Encryption Algorithms*. PhD thesis, Katholieke Universiteit Leuven, 2007.

4. A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine. On Cryptographic Properties of the Cosets of $R(1, m)$. *IEEE Transactions on Information Theory*, 47(4):1494–1513, May 2001.
5. A. Canteaut, M. Daum, H. Dobbertin, and G. Leander. Finding nonnormal bent functions. *Discrete Applied Mathematics*, 154(2):202–218, 2006.
6. C. Carlet and E. Prouff. Vectorial Functions and Covering Sequences. In *Finite Fields and Applications - Fq7*, volume 2948 of *Lecture Notes in Computer Science*, pages 215–248. Springer, 2004.
7. P. Charpin. Normal Boolean functions. *J. Complexity*, 20(2-3):245–265, 2004.
8. J.F. Dillon. *Elementary Hadamard Difference sets*. PhD thesis, University of Maryland, 1974.
9. I. Dinur and A. Shamir. Cube Attacks on Tweakable Black Box Polynomials. In *EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.
10. I. Dinur and A. Shamir. An Improved Algebraic Attack on Hamsi-256. In *FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 88–106. Springer, 2011.
11. H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. In *FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 61–74. Springer-Verlag, 1994.
12. T. Fuhr. Finding second preimages of short messages for Hamsi-256. In *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2010.
13. T. Fuhr. *Conception, preuves et analyse de fonctions de hachage cryptographiques*. PhD thesis, Télécom ParisTech, 2011.
14. K. C. Gupta and P. Sarkar. Improved Construction of Nonlinear Resilient S-Boxes. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 466–483. Springer, 2002.
15. Ö. Küçük. The Hash Function Hamsi. Submission to NIST (Round 2), 2009.
16. G. Leander, M. A. Abdelraheem, H. AlKhzaimi, and E. Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2011.
17. G. Leander and A. Poschmann. On the Classification of 4 Bit S-Boxes. In *Arithmetic of Finite Fields - WAIFI 2007*, volume 4547 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2007.
18. R. L. McFarland. A family of noncyclic difference sets. *Journal of Combinatorial Theory, Series A*, 15:1–10, 1973.
19. K. Nyberg. Perfect nonlinear S-boxes. In *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 378–385. Springer-Verlag, 1991.
20. K. Nyberg. S-boxes and round functions with controllable linearity and differential uniformity. In *FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 111–130. Springer-Verlag, 1995.
21. E. Pasalic and S. Maitra. Linear codes in generalized construction of resilient functions with very high nonlinearity. *IEEE Transactions on Information Theory*, 48(8):2182–2191, 2002.
22. M.-J. O. Saarinen. Cryptographic analysis of all 4×4 sboxes. In *Selected Areas in Cryptography - SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2012.
23. H. Wu. The Hash Function JH. Submission to NIST (Round 3), 2011.