Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions

Thomas W. Cusick[*], Pantelimon Stănică[†‡]

February 7, 2002

Abstract

We study the nonlinearity and the weight of the *rotation-symmetric* (*RotS*) functions defined by Pieprzyk and Qu. We give exact results for the nonlinearity and weight of 2-degree *RotS* functions with the help of the semi-bent functions and we give the generating function for the weight of the 3-degree *RotS* function. Based on the numerical examples and our observations we state a conjecture on the nonlinearity and weight of the 3-degree *RotS* function.

Keywords: Boolean functions; nonlinearity; bent; semi-bent; hash functions

1 Motivation

Hash functions are used to map a large collection of *messages* into a small set of *message digests* and can be used to generate efficiently both signatures and message authentication codes, and they can be also used as one-way functions in key agreement and key establishment protocols. There are two approaches to the study of hash functions:

[*]State University of New York at Buffalo, Department of Mathematics, Buffalo, NY 14260-2900, e-mail: cusick@math.buffalo.edu

[†]Auburn University Montgomery, Department of Mathematics, Montgomery, AL 36124-4023, e-mail: stanica@strudel.aum.edu

[‡]The second author is on leave from the Institute of Mathematics of Romanian Academy, Bucharest, Romania

Information Theory and *Complexity Theory*. The first method provides unconditional security – an enemy cannot attack such systems even if he/she has unlimited computing power. Unfortunately, this is still a theoretical approach and is generally impractical [1]. In the second method based on complexity theory, some assumptions are made on the computing power of the enemy or the weaknesses of the existing systems and algorithms. The best we can hope for is to estimate the computing power necessary for the attacker to break the algorithm. Recent progress in interpolation cryptanalysis [4] and high order differential cryptanalysis [5] has shown that the algebraic degree is an important factor in the design of cryptographic primitives. In fact, in [5] the algebraic degree is the crucial parameter in determining how secure certain cryptosystems are against higher order differential attacks. Together with propagation, differential and nonlinearity profile, resiliency, correlation-immunity, local and global avalanche characteristics they form a class of design criteria which we have to consider in the design of such primitives.

In [6], Pieprzyk and Qu studied some functions, which they called *rotation-symmetric* (*RotS*) as components in the rounds of a hashing algorithm. This is a highly desirable property when efficient evaluation of the function is important, for instance in the implementation of MD4, MD5 or HAVAL, since we can reuse evaluations from previous iterations. It turns out that the degree-two *RotS* function takes $\frac{3n-1}{2}+6(m-1)$ operations (additions and multiplications) to evaluate in m consecutive rounds of a hashing algorithm. In [5] the authors showed how to break in less than 20 milli-seconds

a block cipher that employs quadratic Boolean functions as its S-boxes and is provably secure against linear and differential attacks. Therefore, it is necessary to employ higher degree *RotS* functions in our algorithms. To protect from differential attack, we need *RotS* functions with high nonlinearity. In this paper we aim to continue the study started by Pieprzyk and Qu [6] on the 2-degree *RotS* functions by investigating these in the even dimension and also, by constructing the 3-degree *RotS* functions and proving some results about their weights and nonlinearity. Our techniques apply in principle to *RotS* functions of degree $k > 3$, but it becomes increasingly difficult to calculate the weight of the functions. If Conjecture 12 below (for the case $k = 3$) could be proved, then significant progress for larger k might be possible.

2 Preliminaries

Let $n \geq 6$ be a positive integer and $W_n = \{0,1\}^n$ be the space of binary vectors. Denote $\alpha_0 = (0,\ldots,0,0), \alpha_1 = (0,\ldots,0,1), \ldots, \alpha_{2^n-1} = (1,\ldots,1)$. We use the lexicographical order on the sequence α, that is $\alpha_0 < \alpha_1 < \cdots < \alpha_{2^n-1}$. The Boolean functions will be written in their algebraic normal form (when $\alpha = (a_1,\ldots,a_n)$) as

$$f(x) = \oplus_{\alpha \in W_n} c_\alpha x_1^{a_1} \cdots x_n^{a_n},$$

where $c_\alpha \in W_1$. The truth table of f is the binary sequence

$$f = (v_1, v_2, \ldots, v_{2^n}), \tag{1}$$

where the bits $v_1 = f\big((0,\ldots,0)\big)$, $v_2 = f\big((0,\ldots,0,1)\big), \ldots$. We shall identify the function f with its vector representation in (1). We call a function *balanced* if the number

of ones is equal to the number of zeros in its truth table. The *Hamming weight* of a binary vector v, denoted by $wt(v)$ is defined as the number of ones it contains. The *Hamming distance* between two functions $f, g : W_n \to W_1$, denoted by $d(f, g)$ is defined as $wt(f \oplus g)$. The nonlinearity of a function f, denoted by N_f is defined as

$$\min_{\phi \in A_n} d(f, \phi),$$

where A_n is the class of all affine function on W_n. We say that f satisfies the *propagation criterion (PC)* with respect to c if

$$\sum_{x \in W_n} f(x) \oplus f(x \oplus c) = 2^{n-1}. \tag{2}$$

If f satisfies the PC with respect to all vectors of weight 1, f is called an *SAC* (*Strict Avalanche Criterion*) function. If the above relation happens for any c with $wt(c) \leq s$, we say that f satisfies $PC(s)$, and if $s = n$, then we say that f is a *bent function*. If two functions g, h, on W_n, satisfy $g(x) = h(Ax \oplus a) \oplus (b \cdot x) \oplus c$ with $a, b \in W_n, c \in W_1$, and A a $2k \times 2k$ nonsingular matrix, we say that g is *affinely equivalent* to h.

Definition 1. *The class of* rotation-symmetric *(*RotS*) functions includes all Boolean functions $f : W_n \to W_1$ such that $f(x_1, \ldots, x_n) = f(\rho(x_1), \ldots, \rho(x_n))$, where $\rho(x_i) = x_{i+1}$, and $x_{n+1} := x_1$.*

As in [6], we denote by ρ the permutation $\rho(i) = i + 1, \rho(n) = 1$. By abuse of notation we use the same letter for the transformation which acts on each variable by $\rho(x_i) = x_{i+1}, \rho(x_n) = x_1$. By $\hat{g}$ we mean $(-1)^g$. We define the *Walsh-Hadamard*

transform of a function g on W_n to be the map $\hat{\mathcal{F}}_{\hat{g}} : W_n \to \mathbf{R}$,

$$\hat{\mathcal{F}}_{\hat{g}}(w) = \sum_{x \in W_n} \hat{g}(x)(-1)^{w \cdot x}.$$

The *correlation value* between g and h it is defined by

$$c(g,h) = 1 - \frac{d(g,h)}{2^{n-1}}.$$

If U is a string of bits, then $\bar{U}$ denotes the complemented string with 0 and 1 interchanged. If X is a 4-bit block or a string of blocks, by $(X)_u$ or X_u we shall mean the string obtained by concatenation of u copies of X. The concatenation of two strings u, v will be denoted by uv or $u||v$. Now we define two sets of 4-bit strings

$$\begin{aligned} T_1 = \{A &= 0,0,1,1;\ \bar{A} = 1,1,0,0;\ B = 0,1,0,1; \bar{B} = 1,0,1,0; \\ C &= 0,1,1,0;\ \bar{C} = 1,0,0,1;\ D = 0,0,0,0; \bar{D} = 1,1,1,1\} \end{aligned}$$

and

$$\begin{aligned} T_2 = \{U &= 1,0,0,0;\ \bar{U} = 0,1,1,1;\ V = 0,0,0,1;\ \bar{V} = 1,1,1,0; \\ X &= 0,1,0,0;\ \bar{X} = 1,0,1,1;\ Y = 0,0,1,0;\ \bar{Y} = 1,1,0,1\}. \end{aligned}$$

3 Second Degree Rotation-Symmetric Function

In [6] the authors proved that the homogeneous rotation symmetric function of degree 2, $f_2 = x_1x_l + x_2x_{l+1} + \cdots x_nx_{n+l-1}$, (the subscript w is taken as $((w-1) \bmod n) + 1$) has good nonlinearity and good avalanche properties. Precisely, they proved that the Hamming weight satisfies $2^{n-2} \leq wt(f_2) \leq 2^n - 2^{n-2}$, the nonlinearity $N_{f_2} \geq 2^{n-2}$ and

if n is odd, then $N_{f_2} = 2^{n-1} - 2^{\frac{n-1}{2}}$. Also, f_2 is balanced and satisfies the PC with respect to all vectors α of weight $0 < wt(\alpha) < n$. In particular f_2 is an SAC function. By using the normal form of the function, if n odd, the truth table of f_2 is found using $\frac{3n-1}{2}2^n$ operations (additions and multiplications). We would like to point out that efficient computation of truth tables is necessary if the corresponding functions are to be used in cryptographic protocols.

If n is even and $l = \frac{n}{2} + 1$, then f_2 is constant. In this section we improve the previous results (if $l \neq \frac{n}{2} + 1$) in the following

Theorem 2. *If f_2 is defined on W_n, with $n = 2k$, then it is not bent. The nonlinearity is*

$$N_{f_2} = 2^{2k-1} - 2^k,$$

and the truth table of f_2 can be displayed using only $2^{n-3} - 2$ operations (additions and multiplications). Furthermore, for any n, the weights of f_2 are given by

$$wt(f_2) = 2^{n-1} - 2^{\frac{n}{2}-1}\left(1 + (-1)^n\right). \tag{3}$$

The proof of the previous theorem needs some preparations. For $n \geq 3$, let $t_n = x_1x_2 + x_2x_3 + \cdots + x_{n-2}x_{n-1} + x_{n-1}x_n$. The following lemma proves to be useful

Lemma 3. *If n is even, then t_n is a bent function.*

Proof. We have

$$\begin{aligned} t_{2k} =& x_2(x_1 + x_3) + x_4(x_3 + x_5) + \cdots \\ &+ x_{2k-2}(x_{2k-3} + x_{2k-1}) + x_{2k}x_{2k-1}. \end{aligned}$$

By taking the transformation

$$X_{2i} = x_{2i} \text{ and } X_{2i-1} = x_{2i-1} + x_{2i+1},$$

$$X_{2k-1} = x_{2k-1},\ i = 1, 2, \ldots, k-1,$$

we see that t_{2k} is affinely equivalent to a bent function in the Maiorana-McFarland class (see [3]), therefore it is also bent. □

We say (see [2]) that g on W_{2k+1} is *semi-bent*, if there is a bent function g_0 on W_{2k} with

$$g = g_0||g_1,$$

where $g_1(x) = g_0(Ax \oplus a) \oplus 1$, A is a nonsingular $2k \times 2k$ matrix and a is any vector in W_{2k}.

In [2], the authors prove the following results (see Theorem 18, Corollary 21 and Theorem 16), which will be used in this paper.

Lemma 4. *Any semi-bent function g on W_{2k+1} is balanced, $N_g = 2^{2k} - 2^k$, for any $w^* \in W_{2k+1}$, the correlation value between g and the linear function $l_{w^*}(x) = w^* \cdot x$ is 0 or $\pm 2^{-k}$, and*

$$\#\{w^* \in W_{2k+1} |\, c(g, l_{w^*}) = 0\} = 2^{2k} = \#\{w^* \in W_{2k+1} |\, c(g, l_{w^*}) = \pm 2^{-k}\}.$$

Lemma 5. *Let g on W_{2k+1} be a semi-bent function with $A = I$ and $a = (1, 1, \cdots, 1)$. Then g satisfies $PC(2k)$.*

We define $l_b(x) = b \cdot x$. The following results belong to Preneel [7].

Lemma 6. *If g is the concatenation $g_0||g_1$, $w^* = (w, w_{n+1}) \in W_{n+1}$, then*

$$\hat{\mathcal{F}}_{\hat{g}}(w^*) = \hat{\mathcal{F}}_{\hat{g}_0}(w) + (-1)^{w_{n+1}}\hat{\mathcal{F}}_{\hat{g}_1}(w).$$

Lemma 7. *For h on W_n, $a, b \in W_n, c \in W_1$ and a $2k \times 2k$ nonsingular matrix A, define g by $g(x) = h(Ax \oplus a) \oplus l_b(x) \oplus c$. Then,*

$$\hat{\mathcal{F}}_{\hat{g}}(w) = (-1)^c(-1)^{(A^{-1}a, w\oplus b)}\hat{\mathcal{F}}_{\hat{h}}((A^{-1})^t(w \oplus b)).$$

It is not very difficult to observe (see also [6]) that any 2-degree rotation-symmetric function in n variables is affinely equivalent to $f_2^n = f_2 = x_1x_2 \oplus x_2x_3 \oplus \cdots \oplus x_{n-1}x_n \oplus x_nx_1$. The following lemma is useful for the analysis of an algorithm we display later to evaluate f_2^{2k} fast.

Lemma 8. *Each monomial of degree 2 can be written in the form (1) as*

$$\begin{aligned}
x_ix_j &= \left(D_{2^{n-i-2}}\left(D_{2^{n-j-2}}\bar{D}_{2^{n-j-2}}\right)_{2^{j-i-1}}\right)_{2^{i-1}}, \text{if } 1 \le i < j \le n-2,\\
x_ix_{n-1} &= \left(D_{2^{n-i-2}}A_{2^{n-i-2}}\right)_{2^{i-1}},\\
x_ix_n &= \left(D_{2^{n-i-2}}B_{2^{n-i-2}}\right)_{2^{i-1}},\\
x_{n-1}x_n &= V_{2^{n-2}}.
\end{aligned} \tag{4}$$

Proof. From the self explanatory truth-table

x_1	x_2	x_3	$\cdots$	x_{n-1}	x_n
0	0	0	$\cdots$	0	0
0	0	0	$\cdots$	0	1
0	0	0	$\cdots$	1	0
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
1	1	1	$\cdots$	0	1
1	1	1	$\cdots$	1	0
1	1	1	$\cdots$	1	1

(5)

we get $x_1x_2 = D_{2^{n-2}-2^{n-4}}\bar{D}_{2^{n-4}}$, $x_1x_3 = D_{2^{n-3}}\left(D_{2^{n-5}}\bar{D}_{2^{n-5}}\right)_2$, etc., hence the result.

□

Using (4) we obtain

$$\begin{aligned} f_2 \;=\; & x_1x_2 \oplus x_2x_3 \oplus \cdots x_{n-1}x_n \oplus x_nx_1 = \\ & D_{2^{n-3}}\left(D_{2^{n-4}}\bar{D}_{2^{n-4}}\right) \oplus \left(D_{2^{n-4}}\left(D_{2^{n-5}}\bar{D}_{2^{n-5}}\right)\right)_2 \oplus \\ & \left(D_2\left(D\bar{D}\right)\right)_{2^{n-4}} \oplus (DA)_{2^{n-3}} \oplus (DB)_{2^{n-3}} \oplus V_{2^{n-2}} = \\ & g \oplus (DC)_{2^{n-3}} \oplus V_{2^{n-2}} = g \oplus (V\bar{U})_{2^{n-3}} \oplus D_{2^{n-3}}B_{2^{n-3}}, \end{aligned} \tag{6}$$

where g is the sum of the first $n-3$ strings of length 2^n.

For a string u of length $2^s, s \geq 4$, we denote by $\tilde{u}$, the string obtained by complementing the second half, that is the last 2^{s-1} bits of u. It is not difficult to observe that the following algorithm will output $f_2 = G_1||G_2||G_3$.

Algorithm f2.

step 3*:* $g_1^3 \leftarrow VY, g_2^3 \leftarrow X\bar{U}$

step *s:* $g_i^s \leftarrow g_i^{s-1}||\tilde{g}_i^{s-1}, i = 1, 2$

output*:* $G_1 \leftarrow g_1^{n-4}, G_2 \leftarrow g_2^{n-5}$, $G_3 = \bar{G}_4$, *where* $G_4 = \tilde{G}_2$, *and write* $f_2 = G_1||G_2||G_3$

For instance, the first three steps of the algorithm will produce

$$G_1 \leftarrow ((VY)(V\bar{Y}))((VY)(\bar{V}Y))$$

$$G_2||G_3 \leftarrow (X\bar{U}XU)(\bar{X}UXU).$$

We are ready now to prove the main result of this section.

Proof of Theorem 2. Using the above algorithm, we deduce that the *RotS* function on W_n of degree 2 can be evaluated in $n-2$ steps, which requires

$$(1+2^1+\cdots+2^{n-5})+(1+2^1+\cdots+2^{n-6})+2^{n-5}=2^{n-3}-2$$

operations, since at each step s we complement 2^{s-2} bits.

Take an example, say $f_2^5 = VYV\bar{Y}X\bar{U}\bar{X}\bar{U} = t_5 + x_1x_5$ on W_5. We see that $f_2^5 = t_4(\mathbf{x})||t_4(\mathbf{x}\oplus\mathbf{1})\oplus 1$, therefore it is semi-bent.

It is very easy to see that

$$f_2^{2k+1} = t_{2k}(\mathbf{x}_{2k})||(t_{2k}(\mathbf{x}_{2k})+x_1+x_{2k}).$$

But

$$\begin{aligned}\overline{t_{2k}(\mathbf{x}_{2k})+x_1+x_{2k}} &= \sum_{i=1}^{2k-1} x_ix_{i+1}+x_1+x_{2k}+1 = \\ &= \sum_{i=1}^{2k-1}(x_i+1)(x_{i+1}+1) = t_{2k}(\mathbf{x}_{2k}\oplus\mathbf{1}),\end{aligned}$$

therefore $f_2^{2k+1} = t_{2k}(\mathbf{x}_{2k})||(t_{2k}(\mathbf{x}_{2k}\oplus\mathbf{1})\oplus 1)$ is semi-bent. By Lemma 5, f_2^{2k+1} satisfies the propagation criterion for all weights $1\le w\le 2k$.

Similarly,

$$f_2^{2k} = t_{2k-1}(\mathbf{x}_{2k-1})||(t_{2k-1}(\mathbf{x}_{2k-1})+x_1+x_{2k-1}).$$

Now, we shall use Lemma 6 to compute the nonlinearity of f_2^{2k}. First, we observe that

$$t_{2k+1} = t_{2k}(\mathbf{x}_{2k})||(t_{2k}(\mathbf{x}_{2k})+x_{2k}).$$

Take $A = I$ and $a = (1, 0, 1, 0, \ldots, 1, 0)$. We see that

$$\begin{aligned} t_{2k}(\mathbf{x}_{2k}) + x_{2k} &= t_{2k}(x_1 + 1, x_2, x_3 + 1, \ldots, x_{2k-1}, x_{2k}) \\ &= (x_1 + 1)x_2 + x_2(x_3 + 1) + \cdots + (x_{2k-1} + 1)x_{2k}. \end{aligned}$$

We denote the last expression by $r(\mathbf{x})$. Using Lemma 6 we compute the Walsh-Hadamard transform

$$\hat{\mathcal{F}}_{\hat{r}}(\mathbf{w}_{2k}) = (-1)^{(\mathbf{w},\mathbf{a})}\hat{\mathcal{F}}_{\hat{t}_{2k}}(\mathbf{w}) = \pm 2^k,$$

since by Lemma 3, t_{2k} is bent.

For simplicity we set $t(\mathbf{x}) = t_{2k+1}(\mathbf{x}_{2k+1})$ and $w^* = (w, w_{2k+1})$. Thus,

$$\hat{\mathcal{F}}_{\hat{t}}(w^*) = \hat{\mathcal{F}}_{\hat{t}_{2n}}(w) + (-1)^{w_{2k+1}}\hat{\mathcal{F}}_{\hat{r}}(w) = 0 \text{ or } \pm 2^{k+1}, \tag{7}$$

since r and t_{2k} are bent. Therefore,

$$N_{t_{2k+1}} = 2^{2k} - \frac{1}{2}|\hat{\mathcal{F}}_{\hat{t}_{2k+1}}(w^*)| = 2^{2k} - 2^k.$$

By Lemma 4, $\hat{\mathcal{F}}_{\hat{t}_{2k-1}}(\mathbf{x}_{2k-1}) = 0$ or $\pm 2^k$. Let $v(\mathbf{x}) = t_{2k-1}(\mathbf{x}) + x_1 + x_{2k-1}$. By Lemma 6,

$$\hat{\mathcal{F}}_{\hat{v}}(\mathbf{x}) = (-1)^{(\mathbf{x}\oplus(1,0,\ldots,0,1),\mathbf{0})}\hat{\mathcal{F}}_{\hat{t}_{2k-1}}(\mathbf{x} \oplus (1, 0, \ldots, 0, 1)) = 0 \text{ or } \pm 2^k.$$

Thus, by the same Lemma 6,

$$\hat{\mathcal{F}}_{\hat{f}_2^{2k}}(\mathbf{x}_{2k}) = \hat{\mathcal{F}}_{\hat{t}_{2k-1}}(\mathbf{x}_{2k-1}) + (-1)^{x_{2k}}\hat{\mathcal{F}}_{\hat{v}}(\mathbf{x}) = 0 \text{ or } \pm 2^{k+1},$$

which implies $N_{f_2^{2k}} = 2^{2k-1} - 2^k$. Therefore f_2 is not bent (any bent function in $2k$ variables has nonlinearity $2^{2k-1} - 2^{k-1}$ [2, Th. 13, p. 111]) and the claim on the nonlinearity is proved.

Now, we will evaluate the weights of f_2 for any dimension n. We recall that $f_2 = g_1^{n-1} g_2^{n-2} g_3^{n-2}$. We show that for any s,

$$wt(g_i^s) = 2wt(g_i^{s-2}) + 2^{s-2}, i = 1, 2, 3. \tag{8}$$

Since $g_i^s = g_i^{s-1}\tilde{g}_i^{s-1} = g_i^{s-1} g_i^{s-2} \bar{\tilde{g}}_i^{s-2}$,

$$\begin{aligned} wt(g_i^s) &= wt(g_i^{s-1}) + wt(g_i^{s-2}) + wt(\tilde{g}_i^{s-2}) \\ &= wt(g_i^{s-2}) + wt(\tilde{g}_i^{s-2}) + wt(g_i^{s-1}) + 2^{s-2} - wt(\tilde{g}_i^{s-2}) \\ &= 2\, wt(g_i^{s-2}) + 2^{s-2}, i = 1, 2. \end{aligned} \tag{9}$$

Now, from $g_3^s = \bar{g}_2^{s-1}\tilde{g}_2^{s-1}$, we get

$$\begin{aligned} wt(g_3^s) &= 2^{s-1} - wt(g_2^{s-1}) + wt(\tilde{g}_2^{s-1}) \\ &= 2^{s-1} - wt(g_2^{s-1}) + 2wt(g_2^{s-2}) - wt(g_2^{s-1}) + 2^{s-2} \\ &= 2wt(g_2^{s-2}) - 2wt(g_2^{s-1}) + 2^{s-1} + 2^{s-2} \\ &= wt(g_2^s) - 2wt(g_2^{s-1}) + 2^{s-1}. \end{aligned} \tag{10}$$

The above equation, for $s-1$, produces

$$wt(g_3^{s-1}) = wt(g_2^{s-1}) - 2wt(g_2^{s-2}) + 2^{s-2}. \tag{11}$$

Now, we add (10) plus twice (11), and we get

$$wt(g_3^s) + 2wt(g_3^{s-1}) = wt(g_2^s) - 4wt(g_2^{s-2}) + 2^s.$$

But $wt(g_2^s) = 2wt(g_2^{s-2}) + 2^{s-2}$. By adding the two previous equations we get

$$wt(g_2^{s-2}) = 2^{s-1} + 2^{s-3} - wt(g_3^{s-1}) - \frac{wt(g_3^s)}{2} \tag{12}$$

Replacing (12) into (11), we obtain

$$wt(g_3^{s+2}) = 2wt(g_3^s) + 2^s.$$

This together with (9) will give the following recurrence for the weights of f_2,

$$wt(f_2^n) = 2wt(f_2^{n-2}) + 2^{n-2}. \tag{13}$$

A generating function for the above recurrence is

$$-\frac{32\,\dfrac{z^7}{1-2\,z} + 16\,z^5 + 24\,z^6}{-1+2\,z^2}. \tag{14}$$

We can linearize the recurrence by using the transformation

$$y_n = wt(f_2^n) - 2^{n-1},$$

thus obtaining the recurrence

$$y_n = 2y_{n-2}.$$

Using the above simple recurrence with $wt(f_2^5) = 16$ and $wt(f_2^6) = 24$, we get a closed formula for the weights of f_2 in dimension n, namely

$$2^{n-1} - 2^{\frac{n}{2}-1}\left(1 + (-1)^n\right),$$

and the theorem is proved. □

Remark 9. *In fact, it can be proved that for n odd, f_2 satisfies the PC with respect to all but 4 vectors, namely $(0,\ldots,0)$, $(0,1,\ldots,0,1)$, $(1,0,\ldots,1,0)$, $(1,1,\ldots,1,1)$.*

4 Third Degree Rotation-Symmetric Function

Regarding a *RotS* of arbitrary degree, Pieprzyk and Qu [6] proved that if f_k is an *RotS* function of degree k, then the nonlinearity satisfies $N_{f_k} \geq 2^{n-k}$. As in the case of second degree *RotS* functions, it is easy to observe that any *RotS* function of degree 3 in n variables, $f_3 = f_3^n$, is affinely equivalent to

$$f_3 = x_1x_2x_3 + x_2x_3x_4 + \cdots + x_nx_1x_2. \tag{15}$$

In the first round of a hashing algorithm, the function will consume $3n-1$ operations. This can be reduced to $2n$, if n odd, and $2n-1$ if n even, if we write the function as $f_3 = x_1x_2x_3 + x_3x_4(x_2+x_5) + \cdots$, respectively, $f_3 = (x_1+x_4)x_2x_3 + \cdots$. Thus, to display the truth table of f_3 we need to perform $n \cdot 2^{n+1}$ (if n odd), or $(2n-1)\cdot 2^n$ (if n even) operations (additions, multiplications).

Using a computer program we have determined the nonlinearity of f_3 on W_n, $n \geq 9$, which turns out to be the same as its weight:

n	3	4	5	6	7	8	9
$N_{f_3^n}$	1	4	6	18	36	80	172

We shall assume that $n \geq 10$. Our main result of this section is the following

Theorem 10. *The weights of f_3 satisfy*

$$wt(f_3^n) = 2\left(wt(f_3^{n-2}) + wt(f_3^{n-3})\right) + 2^{n-3}, \tag{16}$$

and the truth table of f_3 can be displayed using only $2^{n-2}+2^{n-4}+2^{n-5}-3\cdot 2^2$ operations (additions and multiplications). Moreover, the generating function for the weight of f_3

is

$$-\frac{8\,\dfrac{z^6}{1-2\,z}+z^3+4\,z^4+4\,z^5}{-1+2\,z^2+2\,z^3}. \tag{17}$$

The series expansion of the above generating function is

$$z^3+4\,z^4+6\,z^5+18\,z^6+36\,z^7+80\,z^8+$$

$$+\quad 172\,z^9+360\,z^{10}+760\,z^{11}+1576\,z^{12}+\mathrm{O}(z^{13}),$$

obtaining once again the weights of f_3^n, for any dimension.

The following result, a generalization of Lemma 8, will be used.

Lemma 11. *The truth table of any monomial $x_{i_1}\cdots x_{i_s}$ of degree s is*

$$\begin{array}{l}
\left(D_{2^{n-i_1-2}}\cdots\left(D_{2^{n-i_s-2}}\bar{D}_{2^{n-i_s-2}}\right)_{2^{i_s-i_{s-1}-1}}\right)_{2^{i_1-1}},\\
\textit{if } 1\le i_1<\cdots<i_s\le n-2,\\
\left(D_{2^{n-i_1-2}}\cdots\left(D_{2^{n-i_{s-1}-2}}M_{2^{n-i_{s-1}-2}}\right)_{2^{i_{s-1}-i_{s-2}-1}}\right)_{2^{i_1-1}},\\
\textit{where } M=A \textit{ or } B \textit{ if } i_s=n-1,\ \textit{respectively } i_s=n,\\
\left(D_{2^{n-i_1-2}}\cdots\left(D_{2^{n-i_{s-2}-2}}V_{2^{n-i_{s-2}-2}}\right)_{2^{i_{s-2}-i_{s-3}-1}}\right)_{2^{i_1-1}},\\
\textit{if } i_{s-1}=n-1 \textit{ and } i_s=n.
\end{array} \tag{18}$$

Proof. Straightforward using the truth table. □

The following algorithm outputs the 3-degree *RotS* function.

Algorithm f3.

step 4*:* $h_1^4 \leftarrow DVDY$ $h_2^4 \leftarrow VDVA$, $h_3^4 \leftarrow XBXC$

step *s:* $h_i^s \leftarrow h_i^{s-1}||\hat{h}_i^{s-1}$

output*:* $H_1 \leftarrow h_1^{n-1}, H_2 \leftarrow h_2^{n-2}, H_3 \leftarrow h_3^{n-3}$, H_4 *is the string obtained from* $\hat{H}_3$ *by complementing its first half, that is* $H_4 = \bar{H}_5$*, where* $H_5 = \tilde{H}_6, H_6 = \hat{H}_3$*. Write* $f_3 = H_1||H_2||H_3||H_4$.

Proof of Theorem 10. Using Lemma 11 in the case $n = 3$, we get

$$x_i x_{i+1} x_{i+2} = \left(D_{2^{n-i-2}} \left(D_{2^{n-i-3}} \left(D_{2^{n-i-4}} \bar{D}_{2^{n-i-3}}\right)\right)\right)_{2^{i-1}}, \tag{19}$$

if $i \leq n-4$, and

$$\begin{aligned} x_{n-3}x_{n-2}x_{n-1} &= (D_3 A)_{2^{n-4}} \\ x_{n-2}x_{n-1}x_n &= (DV)_{2^{n-3}} \\ x_{n-1}x_n x_1 &= D_{2^{n-3}} V_{2^{n-3}} \\ x_n x_2 x_1 &= D_{2^{n-3}+2^{n-4}} B_{2^{n-4}}. \end{aligned} \tag{20}$$

Therefore,

$$\begin{aligned} f_3 = &\sum_{i=1}^{n-4} \left(D_{2^{n-i-2}+2^{n-i-3}+2^{n-i-4}} \bar{D}_{2^{n-i-4}}\right)_{2^{i-1}} \oplus (D_3 A)_{2^{n-4}} \oplus \\ &(DV)_{2^{n-3}} \oplus D_{2^{n-3}+2^{n-4}} B_{2^{n-4}} \oplus D_{2^{n-3}} V_{2^{n-3}} = \end{aligned}$$

$$
\begin{aligned}
= & \sum_{i=1}^{n-4} \left(D_{2^{n-i-2}+2^{n-i-3}+2^{n-i-4}}\bar{D}_{2^{n-i-4}}\right)_{2^{i-1}} \oplus \\
& (DVDY)_{2^{n-4}} \oplus D_{2^{n-3}}V_{2^{n-4}}X_{2^{n-4}} \\
= & \sum_{i=1}^{n-4} \left(D_{2^{n-i-2}+2^{n-i-3}+2^{n-i-4}}\bar{D}_{2^{n-i-4}}\right)_{2^{i-1}} \oplus \\
& (D_3C)_{2^{n-5}} \left(V_3\bar{U}\right)_{2^{n-6}} (X_3Y)_{2^{n-6}} = H_1||H_2||H_3||H_4,
\end{aligned}
$$

where H_1 (on W_{n-1}), H_2 (on W_{n-2}), H_3, H_4 (on W_{n-3}) are defined by the Algorithm $f3$ ($\hat{u}$, on W_j, is the string obtained from u by complementing its last 2^{j-2} bits). We deduce that we need

$$2^2(1+2+\cdots 2^{n-5}) + 2^2(1+2+\cdots 2^{n-6}) +$$

$$2^2(1+2+\cdots 2^{n-7}) + 2^{n-4} + 2^{n-5} = 3\cdot 2^2(2^{n-6}-1) +$$

$$2^{n-3} + 2^{n-4} = 2^{n-2} + 2^{n-4} + 2^{n-5} - 3\cdot 2^2$$

operations to display the truth table of the degree-3 *RotS* function f_3^n.

We shall evaluate the weight of f_3^s for any s. To do this we will compute the weights of each component of f_3^s. We observe that

$$h_i^s = h_i^{s-1}h_i^{s-2}h_i^{s-3}\bar{h}_i^{s-4}\hat{\bar{h}}_i^{s-4} \text{ and}$$

$$\hat{h}_i^s = h_i^{s-1}h_i^{s-2}\bar{h}_i^{s-3}h_i^{s-4}\hat{h}_i^{s-4}, i = 1,2,3.$$

Therefore, denoting by w_i^s the weight of h_i^s, and by $\hat{w}_i^s$ the weight of $\hat{h}_i^s, i = 1,2,3$, we arrive at the following identities:

$$\hat{w}_i^s = 2w_i^{s-1} + 2w_i^{s-2} - w_s + 2^{s-2}, \tag{21}$$

$$w_i^s = w_i^{s-1} + \hat{w}_i^{s-1}. \tag{22}$$

Using Mathematica[1] we obtained the following results on the weights of f_3^n and of each of the four components on dimensions less than 12.

n	$wt(f_3^n)$	$wt(h_1^{n-1})$	$wt(h_2^{n-2})$	$wt(h_3^{n-3})$	$wt(h_4^{n-3})$
3	1				
4	4				
5	6	2			
6	18	6	4		
7	36	14	8	6	8
8	80	32	18	12	18
9	172	72	40	26	34
10	360	156	84	52	68
11	760	336	180	108	136
12	1576	712	376	220	268

(23)

We have

$$wt(f_3^n) = wt(h_1^{n-1}) + wt(h_2^{n-2}) + wt(h_3^{n-3}) + wt(h_4^{n-3}).$$

We show by induction that

$$wt(h_i^s) = 2\left(wt(h_i^{s-2}) + wt(h_i^{s-3})\right) + 2^{s-4}, i = 1, 2, 3, 4. \tag{24}$$

From the table (23) we have the truth of the claim for the first few cases. Assume (24) true for $s-1$ and we prove it for s. From (21) and (22) and by using the induction step we get

$$wt(h_i^s) = wt(h_i^{s-1}) + wt(\hat{h}_i^{s-1}) = 2\left(wt(h_i^{s-2}) + wt(h_i^{s-3})\right) + 2^{s-3}.$$

Similarly for h_4^s. Adding these relations we get

$$wt(f_3^s) = 2\left(wt(f_3^{s-2}) + wt(f_3^{s-3})\right) + 2^{s-3}. \tag{25}$$

[1] A trademark of *Wolfram Research*

Remark that this equation is true for any $s \geq 6$. Using the table (23), the recurrence (25) and Maple[2], we get the expression of the generating function. □

5 Further Comments and a Conjecture

The generalization of our results by this method of evaluating the function with the help of Lemma 11 becomes very difficult since the sum of terms of a *RotS* function using their truth table had no observable pattern, if the degree is greater than 3.

Based on our numerical examples, we give the following conjecture.

Conjecture 12. *The nonlinearity of f_3^n is the same as its weight.*

References

[1] S. Bakhtiari, R. Safavi-Naini, J. Pieprzyk, Cryptographic Hash Functions: A Survey, Preprint 95-9, Department of Computer Science, The University of Wollongong, 1995.

[2] S. Chee, S. Lee, K. Kim, Semi-bent functions, Adv. in Cryptology - Asiacrypt' 94, LNCS 917, Springer-Verlag, 1995, pp. 107-118.

[3] J.F. Dillon, A survey of bent functions, NSA Technical Journal–unclassified (1972), pp. 191-215.

[4] T. Jakobsen, L. Knudsen, The interpolation attack on block ciphers, *Fast Software Encryption*, LNCS 1267 Springer Verlag, 1997, pp. 28-40.

[2] A trademark of *Waterloo Maple*

[5] S. Moriai, T. Shimoyama and T. Kaneko, Higher order differential attack using chosen higher order differences, *Selected Areas in Cryptography - SAC '98*, LNCS 1556, Springer Verlag, 1999, pp. 106-117.

[6] J. Pieprzyk, C.X. Qu, Fast Hashing and Rotation-Symmetric Functions, Journal of Universal Computer Science **5**, no. 1 (1999), pp. 20-31.

[7] B. Preneel, Analysis and design of cryptographic hash functions, Ph.D. dissertation, Katholieke Universiteit Leuven, 1993.

[8] J. Seberry, X.-M. Zhang, Y. Zheng, Nonlinearity and Propagation Characteristics of Balanced Boolean Functions, Information and Computation **119**, no. 1 (1995), pp. 1-13.

Thomas W. Cusick: *State University of New York at Buffalo, Department of Mathematics, Buffalo, NY 14260-2900, e-mail: cusick@math.buffalo.edu*

Pantelimon Stănică: *Auburn University Montgomery, Department of Mathematics, Montgomery, AL 36124-4023, e-mail: stanica@strudel.aum.edu*