Higher-order differential properties of KECCAK and *Luffa*[⋆]

Christina Boura[1,2], Anne Canteaut[1] and Christophe De Cannière[3]

[1] SECRET Project-Team - INRIA Paris-Rocquencourt - B.P. 105
78153 Le Chesnay Cedex - France
[2] Gemalto - 6, rue de la Verrerie - 92447 Meudon sur Seine - France
[3] Department of Electrical Engineering ESAT/SCD-COSIC, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium.
Christina.Boura@inria.fr, Anne.Canteaut@inria.fr, Christophe.DeCanniere@esat.kuleuven.be

Abstract. In this paper, we identify higher-order differential and zero-sum properties in the full KECCAK-f permutation, in the *Luffa* v1 hash function, in components of the *Luffa* v2 algorithm, while also in the full inner permutations of Grøstl and ECHO. These structural properties rely on a new bound on the degree of iterated permutations with a nonlinear layer composed of parallel applications of a number of balanced Sboxes. These techniques yield zero-sum partitions of size 2^{1590} for the full KECCAK-f permutation and several observations on the *Luffa* hash family. We first show that *Luffa* v1 applied to one-block messages is a function of 255 variables with degree at most 251. This observation leads to the construction of a higher-order differential distinguisher for the full *Luffa* v1 hash function, similar to the one presented by Watanabe *et al.* on a reduced version. We show that similar techniques can be used to find all-zero higher-order differentials in the *Luffa* v2 compression function, but the additional blank round destroys this property in the hash function. Finally, some zero-sum partitions for the permutations of the AES-based candidates Grøstl and ECHO are exhibited.

Keywords. Hash functions, degree, higher-order differentials, zero-sums, SHA-3.

1 Introduction

The algebraic degrees of some hash function proposals and of their building blocks have been studied for analyzing their security. In particular, the fact that some inner primitive in a hash function has a relatively low degree can often be used to construct higher-order differential distinguishers, or zero-sum structures. This direction has been investigated in [1,14,4] for three SHA-3 candidates, Luffa, Hamsi and KECCAK. Here, we show how to deduce a new bound for the degree of iterated permutations for a special category of SP-networks. This category includes functions that have for non-linear layer, a number of smaller balanced Sboxes. This class of functions is though quite general. From one side, it includes functions with a big number of small S-boxes of 3 or 4 bits, and from the other side, as every non-linear permutation can be considered as a sort of "big" Sbox, in can include functions with a small number of big-sized Sboxes. This bound shows in particular that the degree grows in a much smoother way than expected when it approaches the number of variables.

For instance, this new bound enables us to find zero-sum partitions for the full inner permutations of the hash functions KECCAK [3], Grøstl [9] and ECHO [2] and for the *Luffa* v1 hash function [6]. Furthermore, by applying a technique similar to that used in [14], and by combining it with the results given by the new bound, we show that the degree of the *Luffa* v2 compression function [7] is slightly lower than expected. This also enables us to find distinguishers for the Q_j permutations and for the compression function of *Luffa* v2. These results do not seem to affect the security of *Luffa* v2, but are another confirmation of the fact that the internal components of *Luffa* do not behave as ideal random functions.

[⋆] Partially supported by the French Agence Nationale de la Recherche through the SAPHIR2 project under Contract ANR-08-VERS-014.

The rest of the paper is organized as follows. In Section 2, a new bound on the degree of iterated permutations is presented when the nonlinear layer consists of several parallel applications of smaller balanced Sboxes. Section 3 recalls how a low algebraic degree can be exploited for mounting higher-order differential distinguishers and zero-sum distinguishers. An application to the full KECCAK-f permutation is presented in Section 4, while applications to the *Luffa* hash family are described in Section 5. We conclude in Section 6 with applications to the full inner permutations of ECHO and Grøstl.

2 A new bound on the degree of some iterated permutations

In the whole paper, the addition in $\mathbb{F}_2^n$, *i.e.* the bitwise exclusive-or will be denoted by $+$, while $\oplus$ will be used for denoting the direct sum of subspaces of $\mathbb{F}_2^n$.

A *Boolean function f of n variables* is a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. It can be expressed as a polynomial, called *algebraic normal form*. The *degree* of f, denoted by *deg*(f), is the degree of its algebraic normal form. Moreover, the *degree* of a vectorial function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ is defined as the highest degree of its coordinates. The Hamming weight of a Boolean function, f, is denoted by $\mathrm{wt}(f)$. It corresponds to the number of x such that $f(x) = 1$. Any function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ is said to be *balanced* if each element in $\mathbb{F}_2^m$ has exactly 2^{n-m} preimages under F.

In this paper, we are interested in estimating the degree of a composed function $G \circ F$. Obviously, we can bound the degree of the composition $G \circ F$ by $\deg(G \circ F) \leq \deg(G)\deg(F)$. Though, this trivial bound is often very little representative of the real degree of the permutation, in particular if we are trying to estimate the degree after a high number of rounds. A first improvement of the trivial bound was provided by Canteaut and Videau [8] when the values occurring in the Walsh spectrum of F are divisible by a high power of 2, *i.e.* if the values $\mathrm{wt}(\varphi_b \circ F + \varphi_a)$ for all $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^m$ are divisible by a high power of 2, where φ_a denotes the linear function $x \mapsto a \cdot x$.

Theorem 1. *[8] Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ such that all values*

$$\mathrm{wt}(\varphi_b \circ F + \varphi_a), \quad a, b \in \mathbb{F}_2^n, b \neq 0$$

are divisible by 2^ℓ, for some integer ℓ. Then, for any $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$, we have

$$\deg(G \circ F) \leq n - 1 - \ell + \deg(G).$$

In particular, this result applies to the functions composed of a nonlinear layer followed by a linear permutation, where the nonlinear layer is defined by the concatenation of m smaller balanced Sboxes $S_1, \ldots, S_m$, defined over $\mathbb{F}_2^{n_0}$, $n_0 \geq 2$. Indeed, since all elements $\mathrm{wt}(\varphi_b \circ S_i + \varphi_a)$ for all smaller functions $S_1, \ldots, S_m$ are divisible by 2, then we deduce that, for the whole permutation, $\mathrm{wt}(\varphi_b \circ F + \varphi_a)$ is divisible by 2^{2m-1}. We will show here how this bound can be further improved in this particular case. The result mainly comes from the following observation.

Proposition 1. *Let F be a balanced function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, and let k be an integer with $1 \leq k \leq m$. Then, all products of k coordinates of F have the Hamming weight 2^{n-k}.*

In particular, if $k < n$, the product of any k coordinates of F has degree at most $(n - 1)$.

Proof. Let $(f_1, \ldots, f_m)$ denote the coordinates of F. Let I be any subset of $\{1, \ldots, m\}$ of size k, and let F_I be the function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^k$ whose coordinates are the $f_i, i \in I$. Since F_I is balanced, the multiset $\{F_I(x), x \in \mathbb{F}_2^n\}$ consists of all elements in $\mathbb{F}_2^k$, each one with multiplicity 2^{n-k}. Therefore, there exist exactly 2^{n-k} values of x such that $F_I(x)$ is the all-one vector, or equivalently such that $\prod_{i \in I} f_i = 1$. ◇

We then deduce the following theorem.

Theorem 2. *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ corresponding to the concatenation of m smaller balanced Sboxes, $S_1, \ldots, S_m$, defined over $\mathbb{F}_2^{n_0}$. Let δ_k be the maximal degree of the product of any k coordinates of anyone of these smaller Sboxes. Then, for any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2^\ell$, we have*

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{\gamma} \ , \tag{1}$$

where

$$\gamma = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i} \ .$$

Most notably, we have

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{n_0 - 1} \ .$$

Moreover, if $n_0 \geq 3$ and all Sboxes have degree at most $n_0 - 2$, we have

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{n_0 - 2} \ ,$$

Proof. Let us denote by π the product of d output coordinates of F. Some of the coordinates involved in π may belong to the same Sbox. Then, for any i, $1 \leq i \leq n_0$, we denote by x_i the integer corresponding to the number of Sboxes for which exactly i coordinates are involved in π. Obviously, we have

$$\deg(\pi) \leq \max_{(x_1, \ldots, x_{n_0})} \sum_{i=1}^{n_0} \delta_i x_i$$

where the maximum is taken over all vectors $(x_1, \ldots, x_{n_0})$ satisfying

$$\sum_{i=1}^{n_0} i x_i = d \text{ and } \sum_{i=1}^{n_0} x_i \leq m \ .$$

Then, we have

$$\begin{aligned}
\gamma \deg(\pi) - d &\leq \gamma \sum_{i=1}^{n_0} \delta_i x_i - \sum_{i=1}^{n_0} i x_i \\
&\leq (\gamma - 1) n_0 x_{n_0} + \sum_{i=1}^{n_0 - 1} (\gamma \delta_i - i) x_i \\
&\leq (\gamma - 1) n_0 \sum_{i=1}^{n_0} x_i - \sum_{i=1}^{n_0 - 1} ((\gamma - 1) n_0 - \gamma \delta_i + i) x_i \\
&\leq (\gamma - 1) n - \sum_{i=1}^{n_0 - 1} ((\gamma - 1) n_0 - \gamma \delta_i + i) x_i \\
&\leq (\gamma - 1) n \ ,
\end{aligned}$$

where the last inequality comes from the fact that all coefficients in the sum are positive. Actually, we have

$$(\gamma - 1) n_0 - \gamma \delta_i + i = \gamma (n_0 - \delta_i) - (n_0 - i) \geq 0$$

by definition of γ. Thus, since $\gamma \deg(\pi) - d \leq (\gamma - 1) n$, we deduce that

$$\gamma \left(n - \deg(\pi) \right) \geq n - d \ .$$

Now, we first show that, if all Sboxes are balanced, then $\gamma \leq n_0-1$. Indeed, for any $1 \leq i \leq n_0-1$, we have

$$\frac{n_0 - i}{n_0 - \delta_i} \leq \frac{n_0 - 1}{1} ,$$

since we know from Proposition 1 that the degree of the product of $(n_0 - 1)$ coordinates of a balanced $n_0 \times n_0$ Sbox cannot be equal to n_0, and thus $\delta_i \leq n_0 - 1$. Also, we can prove that, if the degrees of all Sboxes satisfy $\deg S < n_0 - 1$, then $\gamma \leq n_0 - 2$. Indeed, for $i = 1$, we have

$$\frac{n_0 - i}{n_0 - \delta_i} = \frac{n_0 - 1}{n_0 - \delta_1} \leq \frac{n_0 - 1}{2} \leq n_0 - 2$$

since $n_0 \geq 3$. Similarly, for any i, $2 \leq i < n_0$, we get from Proposition 1 that $\delta_i \leq n_0 - 1$ and so we deduce that

$$\frac{n_0 - i}{n_0 - \delta_i} \leq n_0 - i \leq n_0 - 2 .$$

$\diamond$

It is worth noticing that Bound (1) and the trivial bound are in some sense symetric. Indeed, we have

$$\frac{\deg(G \circ F)}{\deg G} \leq \max_{1 \leq i < n_0} \frac{\delta_i}{i} \quad \text{and} \quad \frac{n - \deg(G \circ F)}{n - \deg G} \geq \left(\max_{1 \leq i < n_0} \frac{n_0 - i}{n_0 - \delta_i} \right)^{-1} .$$

In other words, when representing $\deg(G \circ F)$ as a function of $\deg G$, the trivial bound states that the degree of $G \circ F$ is upper-bounded by a line through the origin with coefficient $\deg F$. When representing the "degree deficiency" $(n - \deg(G \circ F))$ as a function of $(n - \deg G)$, (1) states that the degree deficiency of $G \circ F$ is lower-bounded by a line through the origin with coefficient γ^{-1}. This can be observed on Figure 1 where the parameters correspond to the inverse of Keccak permutation.

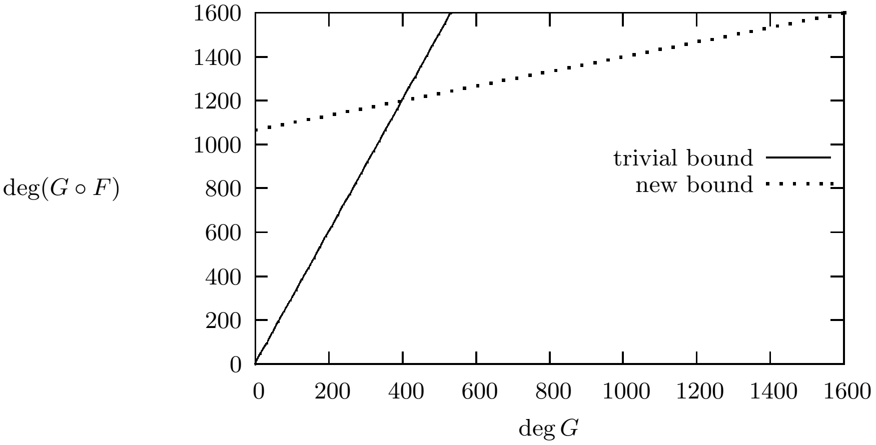

Fig. 1. Evolution of the degree of $G \circ F$ where F is a 1600-variable function composed of 320 cubic permutations over $\mathbb{F}_2^5$.

3 Distinguishing properties related to the algebraic degree

3.1 Higher-order derivatives

The algebraic degree of a permutation F provides some particular distinguishers, which correspond to the values of any derivative of F with respect to a subspace of $\mathbb{F}_2^n$ with dimension $(\deg(F)+1)$. This result comes from the following property of higher-order derivatives of a function.

Definition 1. *[13] Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. For any $a \in \mathbb{F}_2^n$ the* derivative *of F with respect to a is the function $D_aF(x) = F(x+a) + F(x)$. For any k-dimensional subspace V of $\mathbb{F}_2^n$ and for any basis of V, $\{a_1, \dots, a_k\}$, the k-th order derivative of F with respect to V is the function defined by*

$$D_V F(x) = D_{a_1} D_{a_2} \dots D_{a_k} F(x) = \sum_{v \in V} F(x+v), \forall x \in \mathbb{F}_2^n \; .$$

It is well-known that the degree of any first-order derivative of a function is strictly less than the degree of the function. This simple remark, which is exploited in higher-order differential attacks [11], implies that for every subspace V of dimension $(\deg F + 1)$,

$$D_V F(x) = \sum_{v \in V} F(x+v) = 0, \quad \text{for every } x \in \mathbb{F}_2^n.$$

3.2 Zero-sum structures

The existence of zero-sum structures is a distinguishing property which has been recently investigated by Aumasson and Meier [1], Knudsen and Rijmen [12] and by Boura and Canteaut [4].

Definition 2. *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. A* zero-sum *for F of size K is a subset $\{x_1, \dots, x_K\} \subset \mathbb{F}_2^n$ of elements which sum to zero and for which the corresponding images by F also sum to zero,* i.e.,

$$\sum_{i=1}^{K} x_i = \sum_{i=1}^{K} F(x_i) = 0 \; .$$

It has been shown in [4] that any function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ has zero-sums of size less than or equal to 5. However, when F is a permutation over $\mathbb{F}_2^n$, a much stronger property, named *zero-sum partition*, can be investigated.

Definition 3. *Let P be a permutation from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. A* zero-sum partition *for P of size $K = 2^k$ is a collection of 2^{n-k} disjoint zero-sums $X_i = \{x_{i,1}, \dots, x_{i,2^k}\} \subset \mathbb{F}_2^n$* i.e.,

$$\bigcup_{i=1}^{2^{n-k}} X_i = \mathbb{F}_2^n \text{ and } \sum_{j=1}^{2^k} x_{i,j} = \sum_{j=1}^{2^k} P(x_{i,j}) = 0, \; \forall 1 \le i \le 2^{n-k} \; .$$

Here, we focus on the search for zero-sum partitions coming from structural properties of the permutation P, when P is an iterated permutation of the form

$$P = R_r \circ \dots \circ R_1,$$

where all R_i are simpler permutations over $\mathbb{F}_2^n$, named the *round permutations.* The fact that the permutation used in a hash function does not depend on any secret parameter allows to exploit the previous property starting from the middle, *i.e.*, from an intermediate internal state.

This property was used by Aumasson and Meier [1] and also by Knudsen and Rijmen in the case of a known-key property of a block cipher [12]. The only information needed for finding such zero-sums on the iterated permutation using this first approach is an upper bound on the algebraic degrees of both the round transformation and its inverse.

More precisely, we consider $P = R_r \circ \ldots \circ R_1$, and we choose some integer t, $1 \leq t \leq r$. We define the following functions involved in the decomposition of P: F_{r-t} consists of the last $(r-t)$ round transformations, *i.e.*, $F_{r-t} = R_r \circ \ldots \circ R_{t+1}$ and G_t consists of the inverse of the first t round transformations, *i.e.*, $G_t = R_1^{-1} \circ \ldots \circ R_t^{-1}$. Then, as detailed in [1] and in [4], we can find many zero-sum partitions for P of size 2^{d+1} where $d = \max(\deg(F_{r-t}), \deg(G_t))$.

Besides the degree of the round transformation, it has been shown in [4] that some properties of the linear layer in the round transformation may also be exploited for constructing zero-sum partitions, in particular when the nonlinear layer of the round transformation consists of parallel applications of smaller functions defined over $\mathbb{F}_2^{n_0}$. In the following, we denote by B_i, $0 \leq i < m$, the n_0-dimensional subspaces corresponding to the inputs of these smaller Sboxes, *i.e.*,

$$B_i = \langle e_{n_0 i}, \ldots, e_{n_0 i + n_0 - 1} \rangle$$

where $e_0, \ldots, e_{n-1}$ denotes the canonical basis of $\mathbb{F}_2^n$ and where the positions of the n bits in the internal state are numbered such that the n_0-bit Sboxes apply on n_0 consecutive input variables. Then, it was shown in [4] that it is possible to extend a number of zero-sum partitions that have been found for t rounds, to $t+1$ rounds, without increasing the complexity.

Proposition 2. *[4] Let d_1 and d_2 be such that $\deg(F_{r-t-1}) \leq d_1$ and $\deg(G_t) \leq d_2$. Let us decompose the round transformation after t rounds into $R_{t+1} = A_2 \circ \chi \circ A_1$ where both A_1 and A_2 have degree 1 and χ corresponds to the concatenation of m smaller permutations defined over $\mathbb{F}_2^{n_0}$. Let $\mathcal{I}$ be any subset of $\{0, \ldots, m-1\}$ of size $\lceil (d+1)/n_0 \rceil$, let*

$$V = \bigoplus_{i \in \mathcal{I}} B_i$$

and W be its complement. Then, the sets

$$X_a = \{(G_t \circ A_1^{-1})(a+z),\ z \in V\},\quad a \in W$$

form a zero-sum partition of $\mathbb{F}_2^n$ of size 2^k, with $k = n_0 \lceil \frac{d+1}{n_0} \rceil$, for the r-round permutation P.

The complexity of the attacks based on the zero-sum problem is discussed in details in [4], where in particular, attention is paid on zero-sums of small size.

4 Application to the Keccak-*f* permutation

4.1 The Keccak-*f* permutation

Keccak [3] is one of the fourteen hash functions selected for the second round of the SHA-3 competition. Its mode of operation is the sponge construction. The inner primitive in Keccak is a permutation, composed of several iterations of very similar round transformations. Within the Keccak-family, the SHA-3 candidate operates on a 1600-bit state, which is represented by a 3-dimensional binary matrix of size 5 × 5 × 64. Then, the state can be seen as 64 parallel slices, each one containing 5 rows and 5 columns. The permutation in Keccak is denoted by Keccak-$f[b]$, where b is the size of the state. So, for the SHA-3 candidate, $b = 1600$.

The number of rounds in KECCAK-f[1600] was 18 in the original submission, and it has been updated to 24 for the second round. Every round R consists of a sequence of 5 permutations modifying the state:

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta.$$

The functions θ, ρ, π, ι are transformations of degree 1 providing diffusion in all directions of the 3-dimensional state. Then, keeping the same notation as in the previous section, we have $A_1 = \pi \circ \rho \circ \theta$, which is linear and $A_2 = \iota$, which corresponds to the addition of a constant value. Therefore, the linear part of $A = A_1 \circ A_2$ corresponds to $L = \pi \circ \rho \circ \theta$. The nonlinear layer, χ, is a quadratic permutation which is applied to each row of the 1600-bit state. In other words, 320 parallel applications of χ_0 are implemented in order to provide confusion. The inverse permutation, denoted by χ^{-1}, is a permutation of degree 3.

We need to define a numbering for the $n = 1600$ bits of the internal state of KECCAK-f. We associate to the bit of the state positioned at the intersection of the i-th row and the j-th column of the k-th slice, *i.e.*, to the element (i, j, k), $0 \leq i \leq 4$, $0 \leq j \leq 4$, $0 \leq k \leq 63$, the number $25k + 5j + i$. We recall that the elements of the form $(0, 0, z)$ are found in the center of each slice. Then, the 5-dimensional subspace corresponding to the j-th row in the k-th slice, $0 \leq j \leq 4$, $0 \leq k \leq 63$, is defined by

$$B_{5k+j} = \langle e_{25k+5j}, e_{25k+5j+1}, e_{25k+5j+2}, e_{25k+5j+3}, e_{25k+5j+4} \rangle.$$

4.2 Zero-sum partitions for the full KECCAK-*f* permutation

We apply here Theorem 2 to the KECCAK-f round permutation, which is denoted by R. For any F,

$$\deg(F \circ R) = \deg(F \circ \chi) \leq n - \frac{n - \deg(F)}{3}$$

and

$$\deg(F \circ R^{-1}) = \deg((F \circ L^{-1}) \circ \chi^{-1}) \leq n - \frac{n - \deg(F)}{3}$$

by using that the inverse of χ has degree 3. By combining this bound with the trivial bound, we get the bound presented in Table 1 on the degree of several iterations of the round permutation of KECCAK-f and of its inverse. The results for the first rounds are obtained by using the trivial bound and from the round in bold letters and on all the results are due to the new bound. With this new bound, we can use the technique presented in Proposition 2 for finding zero-sum partitions for the full KECCAK-f permutation. Namely, we consider the intermediate states after the linear layer $L = \pi \circ \rho \circ \theta$ in the 11-th round. Let us choose any subspace V in $\mathbb{F}_2^{1600}$ corresponding to a collection of 318 rows (out of the 320), implying $\dim V = 1590$. Then, the sets

$$X_a = \{(G_{10} \circ L^{-1})(a + z),\ z \in V\},\ \ a \in \mathbb{F}_2^{1600}\ ,$$

where G_{10} denotes the inverse of the first 10 rounds, form a zero-sum partition of size 2^{1590} for the full KECCAK-f permutation. This comes directly from Proposition 2 and from the fact that the inverse of the first 10 rounds of the permutation have degree at most $1589 < \dim V$, and that the last 13 rounds have degree at most $1578 < \dim V$.

5 Application to the hash function *Luffa*

5.1 The Luffa hash function

The *Luffa* hash function [6,7] is also a Round-2 candidate of the NIST SHA-3 competition. Its mode of operation is based on a variant of the sponge design. The internal state in *Luffa*

Table 1. Upper bounds on the degree of several rounds of Keccak-f and of its inverse.

forward		backward	
# rounds	bound on deg(R^r)	# rounds	bound on deg(R^{-r})
1	2	1	3
2	4	2	9
3	8	3	27
4	16	4	81
5	32	5	243
6	64	6	729
7	128	**7**	**1309**
8	256	8	1503
9	512	9	1567
10	1024	10	1589
11	**1408**	11	1596
12	1536	12	1598
13	1578	13	1599
14	1592		
15	1597		
16	1599		

consists of w 256-bit words where w equals $3, 4$ and 5 for the output lengths 256, 384 and 512 bits respectively. At each iteration, a 256-bit message block is processed by applying a linear message injection function *MI*. Then, a permutation is applied to the output as follows: the state is split into w 256-bit words and w parallel 256-bit permutations Q_j are applied to each word independently.

The internal state of each permutation Q_j is now divided in 8 words of 32 bits, denoted by $a_0, \ldots, a_7$. Each permutation consists of an input tweak applied only once at the beginning of each permutation and 8 rounds of a round transformation `Step`. The `Step` function consists of a nonlinear transformation called `SubCrumb`, a linear transformation `MixWord` and an addition of constants `AddConstant`. The nonlinear part `SubCrumb` consists of 64 parallel applications of a 4×4 cubic permutation.

Finally, a finalization step is applied. It consists of several iterations of a blank round with fixed message `0x0...00` followed by a linear output function *OF*. In *Luffa* v1, a blank round with message block `0x0...00` is applied at the beginning of the finalization, only if the number of padded message blocks is strictly greater than one. In *Luffa* v2 such a blank round is always applied, in order to prevent higher-order differential attacks.

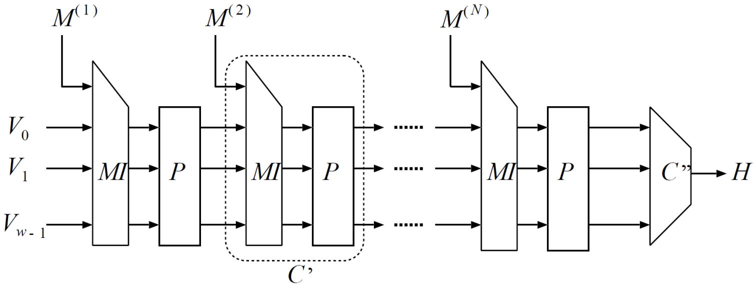

Fig. 2. The *Luffa* construction

The `SubCrumb` Permutation. The input of every Sbox has four bits, every one coming from a different word a_k: S substitutes the ℓ-th bits of a_0, a_1, a_2, a_3 (or a_4, a_5, a_6, a_7) by a 4×4 Sbox of degree 3. The Sbox used in the original submission, *Luffa* v1, was

$$S_1[16] = \{7, 13, 11, 10, 12, 4, 8, 3, 5, 15, 6, 0, 9, 1, 2, 14\} \ ,$$

but the terms of degree 3 in the first three coordinates of this Sbox are similar. This property has been exploited in [14] for showing that the degree of Q_j does not grow as expected. In particular, Q_j reduced to 5 rounds out of 8 has degree at most 130, and the sum of the first two coordinates of Q_j after 6 rounds has degree at most 214. In order to avoid these unsuitable properties, the designers have modified the Sbox according to the strategy detailed in [5]. The new Sbox used in *Luffa* v2, is then

$$S_2[16] = \{13, 14, 0, 1, 5, 10, 7, 6, 11, 3, 9, 12, 15, 8, 2, 4\} \ ,$$

and the algebraic normal forms of its outputs are

$$\begin{aligned}
y_0 &= 1 + x_0 + x_1 + x_1x_2 + x_0x_3 + x_1x_3 + x_0x_1x_3 + x_0x_2x_3\\
y_1 &= x_0 + x_3 + x_0x_1 + x_1x_2 + x_0x_3 + x_1x_3 + x_0x_1x_3 + x_0x_2x_3\\
y_2 &= 1 + x_1 + x_3 + x_0x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_0x_1x_2 + x_0x_1x_3\\
y_3 &= 1 + x_1 + x_2 + x_0x_3 + x_0x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_0x_1x_2 + x_0x_1x_3
\end{aligned}$$

Then the substitution by S is given by

$$\begin{aligned}
b_{3,\ell}||b_{2,\ell}||b_{1,\ell}||b_{0,\ell} &= S[a_{3,\ell}||a_{2,\ell}||a_{1,\ell}||a_{0,\ell}], \quad 0 \le \ell < 32,\\
b_{7,\ell}||b_{6,\ell}||b_{5,\ell}||b_{4,\ell} &= S[a_{7,\ell}||a_{6,\ell}||a_{5,\ell}||a_{4,\ell}], \quad 0 \le \ell < 32.
\end{aligned}$$

in *Luffa* v1. In *Luffa* v2, the order of the last four input words is modified when entering the Sbox in order to break the symmetries exploited in [14]:

$$\begin{aligned}
b_{3,\ell}||b_{2,\ell}||b_{1,\ell}||b_{0,\ell} &= S[a_{3,\ell}||a_{2,\ell}||a_{1,\ell}||a_{0,\ell}], \quad 0 \le \ell < 32,\\
b_{4,\ell}||b_{7,\ell}||b_{6,\ell}||b_{5,\ell} &= S[a_{4,\ell}||a_{7,\ell}||a_{6,\ell}||a_{5,\ell}], \quad 0 \le \ell < 32.
\end{aligned}$$

The `MixWord` Permutation. `MixWord` is a linear permutation of two words. If $z_0, \dots, z_7$ are the 8 words of the state after the application of `Step` we have that

$$\begin{aligned}
(z_0, z_4) &= \texttt{MixWord}(b_0, b_4),\\
(z_1, z_5) &= \texttt{MixWord}(b_1, b_5),\\
(z_2, z_6) &= \texttt{MixWord}(b_2, b_6),\\
(z_3, z_7) &= \texttt{MixWord}(b_3, b_7).
\end{aligned}$$

5.2 Algebraic degree of the Q_j permutation and its inverse

We now show that the approach used in [14] still applies to some extent to the *Luffa* v2 nonlinear function, and that this approach can be combined with Theorem 2 in order to find a new upper bound on the degree of several iterations of the `Step` function.

The remarkable property comes from the fact that the sum of the four coordinates of S_2 has degree 2 only:

$$d = y_0 + y_1 + y_2 + y_3 = 1 + x_1 + x_2 + x_0x_1 + x_0x_3 \ .$$

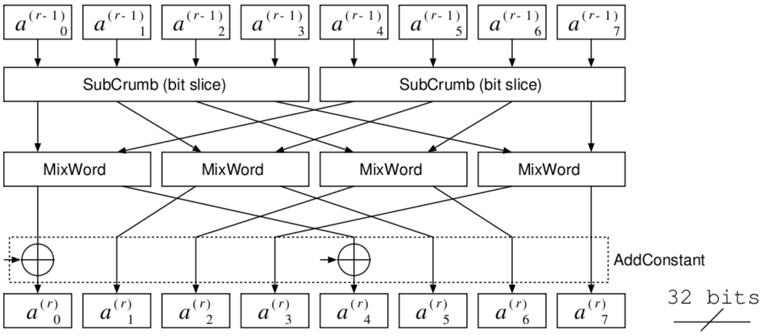

Fig. 3. The `Step` function

Let $x_i^r = \left(x_{i,\ell}^r\right)_{0\le\ell<32}$ denote the output words of r rounds of `Step`, and let $d_{0,\ell}^r$ (resp. $d_{4,\ell}^r$) denote the sum $x_{0,\ell}^r + x_{1,\ell}^r + x_{2,\ell}^r + x_{3,\ell}^r$ (resp. $x_{4,\ell}^r + x_{5,\ell}^r + x_{6,\ell}^r + x_{7,\ell}^r$). Now, let us consider the sum of any two distinct monomials of degree 3 in 4 variables. Any such two monomials share two variables. Then, if we denote by d the sum of all four variables, we obtain that

$$\begin{aligned} x_i x_j x_k + x_i x_j x_{k'} &= x_i x_j x_k + x_i x_j (x_i + x_j + x_k + d) \\ &= x_i x_j x_k + x_i x_j + x_i x_j + x_i x_j x_k + x_i x_j d \\ &= x_i x_j d \ . \end{aligned}$$

It follows that, since all coordinates of the Sboxes S_2 contain an even number of distinct monomials of degree 3, the degrees of their outputs (and then the degree of the output of $(r+1)$ rounds) satisfy

$$\deg x_{i,\ell}^{r+1} \le 2 \max_{0\le i\le 3} \deg x_{i,\ell}^r + \deg d_{0,\ell}^r < 3 \max_{0\le i\le 3} \deg x_{i,\ell}^r, \ \forall 0 \le i \le 3 \ . \tag{2}$$

Moreover, this property holds for any ordering of the inputs and outputs of the Sbox, implying

$$\deg x_{i,\ell}^{r+1} \le 2 \max_{4\le i\le 7} \deg x_{i,\ell}^r + \deg d_{4,\ell}^r < 3 \max_{4\le i\le 7} \deg x_{i,\ell}^r, \ \forall 4 \le i \le 7 \ .$$

Now, since the linear layer consists of the same function applied to all pairs of words (b_k, b_{k+4}) for $0 \le k \le 3$ separately, we deduce that

$$\begin{aligned} d_{0,\ell}^{r+1} &= x_{0,\ell}^{r+1} + x_{1,\ell}^{r+1} + x_{2,\ell}^{r+1} + x_{3,\ell}^{r+1} \\ &= \sum_{i=0}^{3} \texttt{MixWord}_{0,\ell}(b_i, b_{i+4}) \\ &= \texttt{MixWord}_{0,\ell}\left(\sum_{i=0}^{3} b_i, \sum_{i=0}^{3} b_{i+4}\right) \end{aligned}$$

and

$$d_{4,\ell}^{r+1} = \texttt{MixWord}_{1,\ell}\left(\sum_{i=0}^{3} b_i, \sum_{i=0}^{3} b_{i+4}\right) \ .$$

Therefore, the degrees of $d_{0,\ell}^{r+1}$ and of $d_{4,\ell}^{r+1}$ correspond to the degrees of the sum of the four coordinates of the Sboxes, implying

$$\deg d_{i,\ell}^{r+1} \le 2 \max_{i\le j\le i+3} \deg x_{j,\ell}^r \ , i \in \{0, 4\} \ . \tag{3}$$

Both recurrence relations (2) and (3) lead to the bounds presented in Table 2 on the degrees of several iterations of `Step` for the new nonlinear layer (*i.e.* for the new Sbox S_2 and the ordering of the input variables).

Table 2. Upper bounds on the algebraic degree of the output of r iterations of `Step` for *Luffa* v2 (and comparison with the results obtained in [14] for *Luffa* v1).

r	*Luffa* v2		*Luffa* v1	
	$\deg x_{i,\ell}^r$	$\deg d_{i,\ell}^r$	$\deg x_{i,\ell}^r$	$\deg d_{i,\ell}^r$
1	3	2	3	2
2	8	6	8	5
3	22	16	20	13
4	60	44	51	33
5	164	120	130	84
6		-		214

Now, for $r \geq 6$, we apply Theorem 2, exploiting the fact that `Step` is the composition of a linear layer and of several parallel applications of a smaller balanced Sbox of degree 3 defined over $\mathbb{F}_2^4$. Then, for any G, we have

$$\deg(G \circ \texttt{Step}) \leq \frac{512 + \deg(G)}{3} ,$$

implying

$$\max_{i,\ell} \deg(x_{i,\ell}^r) \leq \frac{512 + \max_{i,\ell} \deg(x_{i,\ell}^{r-1})}{3} \text{ and } \max_{i,\ell} \deg(d_{i,\ell}^r) \leq \frac{512 + \max_{i,\ell} \deg(d_{i,\ell}^{r-1})}{3} .$$

These new bounds are given in Table 3.

Table 3. Upper bounds on the algebraic degree of the output of r iterations of `Step` for *Luffa* v1 and *Luffa* v2.

r	*Luffa* v2		*Luffa* v1	
	$\deg x_{i,\ell}^r$	$\deg d_{i,\ell}^r$	$\deg x_{i,\ell}^r$	$\deg d_{i,\ell}^r$
1	3	2	3	2
2	8	6	8	5
3	22	16	20	13
4	60	44	51	33
5	164	120	130	84
6	225	210	214	198
7	245	240	242	236
8	252	250	251	249

It is worth noticing that the same upper bounds hold for the degree of r iterations of the inverse of `Step` in *Luffa* v2 since the algebraic normal form of the inverse of S_2 is

$$\begin{aligned}
y_0 &= x_0 + x_2 + x_3 + x_2x_3 \\
y_1 &= 1 + x_3 + x_0x_1 + x_0x_2 + x_0x_3 + x_2x_3 + x_0x_2x_3 + x_1x_2x_3 \\
y_2 &= x_1 + x_2 + x_3 + x_0x_1 + x_1x_2 + x_0x_3 + x_1x_3 + x_0x_1x_3 + x_0x_1x_2 + x_0x_2x_3 + x_1x_2x_3 \\
y_3 &= x_1 + x_2 + x_3 + x_0x_2 + x_2x_3 + x_0x_1x_2 + x_0x_1x_3 \ .
\end{aligned}$$

Then, the sum of the four coordinates of S_2^{-1} is equal to

$$1 + x_0 + x_2 + x_1x_2 + x_1x_3 + x_2x_3$$

and has degree 2 only. Moreover, all four coordinates of S_2^{-1} have an even number of monomials of degree 3. Then, the previously described technique for upper-bounding the degree of several iterations of the round function applies similarly when computing the inverse.

5.3 Higher-order differentials for the compression function of *Luffa* v2

The compression function in *Luffa* v2 takes as input a $256w$-bit chaining value and a 256-bit message block and it outputs a new $256w$-bit chaining value, where w equals $3, 4$ and 5 when the output length is 256, 384 and 512. Then, we have proved that this function has degree at most 252, while it is expected from its construction to have degree 255.

A first consequence is the existence of all-zero higher-order differentials for the full compression function of *Luffa* v2, similar to those found in [14] for *Luffa* v1 reduced to 7 steps. Let us choose a position ℓ_0 among the 32 possible positions in a word, $0 \leq \ell_0 < 32$, and let us consider any coset of the linear subspace V of the set of all possible message blocks defined as

$$V = \langle e_{i,\ell}, 0 \leq i \leq 7,\ \ell \neq \ell_0 \rangle \ .$$

Then, V has dimension 248. For any fixed chaining value, the message injection function MI stabilizes the subspaces $\langle e_{i,\ell}, 0 \leq i \leq 7 \rangle$, implying that the input of each Q_j is a coset of V. Now, the tweak function at the beginning of each Q_j rotates the least significant four words by a number of bits depending on j. Its output then corresponds to a coset of a subspace V', which is the direct sum of 4-dimensional subspaces of the form $\langle e_{i,\ell}, 0 \leq i \leq 3 \rangle$ or $\langle e_{i,\ell}, 4 \leq i \leq 7 \rangle$. Since the first nonlinear layer applies to those 4-dimensional subspaces separately, it stabilizes the structure of V'. Therefore, the output of the first iteration of Step in each Q_j varies in a coset of a subspace of dimension 248. Then, the outputs of the compression function, *i.e.*, after 8 iterations of Step, sum to zero when the message block varies in any coset of V, since $\dim V = 248 > 246 > \deg(\mathtt{Step}^7)$. This observation holds for any size of the hash value. It should be noted that, by nature, this algebraic property is very different from the properties exploited in previously known distinguishers on the compression function of *Luffa* v2 (e.g. [10]).

5.4 Zero-sum partitions for the Q_j permutations

We consider the subspace V generated by the first 23 bits in a given word, that is

$$V = \langle e_{i_0,0}, e_{i_0,1}, \ldots, e_{i_0,22} \rangle \ ,$$

for some $0 \leq i_0 \leq 7$. Then, we can show that the sets

$$X_a = \{\mathtt{Tweak}_j^{-1} \circ \left(\mathtt{Step}^{-1}\right)^4 (a + z),\ z \in V\}, a \in \mathbb{F}_2^{256} \ .$$

form a zero-sum partition of size 2^{23} for each Q_j.

We first consider any coset of V as input of 4 rounds of Step. Then, the 23 active bits in V correspond to the inputs of 23 different Sboxes and thus the first round has degree 1. As 3 iterations of Step have degree at most 22, we deduce that

$$\sum_{x \in X_a} Q_j(x) = 0 \ .$$

We now focus on the backward computation. We first take the image of V under the inverse of the linear application `MixWord`. As all the variables are in the same word a_{i_0}, after the application of the inverse of the linear layer, all words are constant except the words of index i_0 and $(i_0 + 4)$. But the bits of the words a_{i_0} and a_{i_0+4} all go to different Sboxes, implying that the first round backwards is linear. As we have proven that the inverse of 3 iterations of `Step` has degree at most 22, we deduce that

$$\sum_{x \in X_a} x = 0 \ .$$

There exist $\binom{32}{23} \times 8 = 2^{27.7}$ such zero-sum partitions for each Q_j corresponding to all possible choices for V, *i.e.*, for all possible choices for i_0 and for the 23 positions within the word of index i_0.

5.5 Higher-order differentials for the full *Luffa* v1 hash function

It is shown in [14] that, when hashing messages of length at most 256 bits, the reduced version of *Luffa* v1 hash function, with 7 out of 8 steps in each Q_j, does not behave as a random function. Actually, if the message block varies in some particular subspace of dimension 216, then some linear combination of the output words of this reduced version of *Luffa* v1 sums to zero. This property comes from the fact that *Luffa* v1 does not perform any blank round for one-block messages, and that, after 6 rounds of `Step`, some linear combinations of the output words have degree at most 214.

Even if the advantage that this property could give to an attacker is unclear, this unsuitable property has led the designers to modify the function for the second round of the SHA-3 competition. In particular, a blank round is performed for any message length in *Luffa* v2.

It turns out that this was probably a prudent decision, as the new upper bound on the degree of Q_j for *Luffa* v1 given in Table 3 now shows that a similar distinguisher can be exhibited for the full *Luffa* v1, since the degree of the two words obtained by

$$(y_0 + y_1 + y_2 + y_3, y_4 + y_5 + y_6 + y_7)$$

after 7 iterations of `Step` is at most 236. We then get a similar distinguisher based on the fact that the corresponding linear combinations of the bits of the hash values sum to zero when the message block varies in some particular subspace of dimension 240.

More interestingly, we have shown that the full *Luffa* v1 hash function, when applied to one-block messages, has degree at most 251 in the 255 bits of the message.

5.6 Degree of the full *Luffa* v2 hash function with chosen IVs

The previous observation does not hold for *Luffa* v2 since a blank round is performed for any message length. However, if we would consider the $256w$ bits of the initial value of *Luffa* v2 as an additional input which can freely be chosen, then we can still make some theoretical observations for the hash function applied to one-block messages. Recall that w equals 3, 4 and 5 for a message digest of 256, 384 and 512 bits respectively.

In this setting, *Luffa* v2 is a function from $(256(w + 1) - 1)$ bits to $128(w - 1)$ bits, where the input bits correspond to the bits of the initial value and of the message block. But, *Luffa* v2 is composed of a linear message injection function, followed by a function G from $256w$ bits to $128(w - 1)$ bits. Therefore, the degree of the $(256(w + 1) - 1)$-bit function *Luffa* v2 is equal to the degree of G and cannot exceed $256w$. Moreover, we can show that the degree of this function is even smaller than $256w$ due to the particular design of the inner permutation.

This new upper bound on the degree of *Luffa* v2 comes from the fact that G can be decomposed as the inner permutation P, *i.e.*, the parallel applications of w independent nonlinear permutations Q_j of $n_0 = 256$ variables with degree less than $(n_0 - 2)$, followed by some rounds of the finalization function `Final`. Moreover, the first 256 bits of the message digest are extracted after a single application of `Final`. Then, using that the finalization function consists of 8 iterations of `Step` and has then degree at most 252, Theorem 2 implies that

$$\begin{aligned}\deg(\mathtt{Final} \circ P) &\leq 256w - \frac{256w - \deg(\mathtt{Final})}{254} \\ &\leq 256w - \frac{256w - 252}{254} \\ &< 256w - (w-1) \ .\end{aligned}$$

For the $(128(w-1))$-bit version of *Luffa* v2, we get that the first 256 output bits of *Luffa* v2 have degree at most $(256w - w)$. This property must be compared to the probability that this property holds for a randomly chosen function from $\mathbb{F}_2^{256(w+1)-1}$ to $\mathbb{F}_2^{256}$ bits. Such a function can be written as a polynomial with coefficients in $\mathbb{F}_{2^{256}}$ and the number of its monomials of degree greater than $(256w - w + 1)$ is

$$\sum_{i=0}^{256+w-1} \binom{256(w+1)-1}{i} \ .$$

Therefore, the probability that a randomly chosen function with the same parameters as *Luffa* v2-256 has degree at most 765 is $2^{-2^{837}}$.

For the 384-bit version (resp. for the 512-bit version), *i.e.*, for $w = 4$ (resp. $w = 5$), we get that *Luffa* v2 has 1280 variables (resp. 1536 variables) and degree at most 1020 (resp. 1275). The probability that this property holds for a randomly chosen function with the same parameters is $2^{-2^{933}}$ (resp. $2^{-2^{1010}}$).

6 Application to the inner permutations of Grøstl and ECHO

We have seen in the previous sections that the new bound can be successfully applied to functions owning their nonlinearity to the parallel application of a big number of small balanced Sboxes. We show in this section that similar results can be obtained by the application of the new bound to some AES-based functions. In some cases, an Sbox is not considered in its classical sense but takes the role of a non-linear permutation applied to a part of the state.

The new bound is thus applied to the permutation P of the second-round candidate ECHO [2] and to the permutation P of the third-round candidate Grøstl [9]. In both cases this application reveales that the algebraic degree grows in a quite smoother way than expected. Further to these results, a number of zero-sum partitions is built for both permutations.

6.1 Application on the Grøstl's inner permutation

Grøstl [9] is a third round candidate of the SHA-3 competition. The Grøstl's family members are iterated hash functions, based on a compression function f, capable of returning message digests of any number of bytes from 1 to 64. The compression function f is built out of two large distinct permutations P and Q. The general design of f can be seen in Figure 6.1.

Grøstl's permutations P and Q are iterated permutations, based on the Rijndael block cipher. They are applied to a 512-bit state for the 256-bit candidate and to a 1024-bit state for the 512-bit candidate. This state can be represented by a matrix of bytes A, with 8 rows and 8

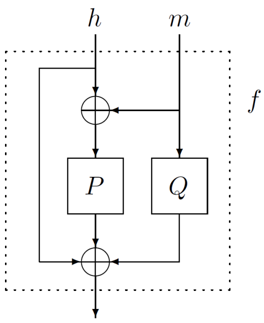

Fig. 4. The Grøstl's compression function

columns for the 256-bit version and 8 rows and 16 columns for the 512 version. In every round R, a total of four transformations inspired from the AES design are performed in the above order:

$$R = \text{MixBytes} \circ \text{ShiftBytes} \circ \text{SubBytes} \circ \text{AddRoundConstant}.$$

The `AddRoundConstant` transformation adds a round dependent constant to every byte of the state matrix A. These constants are different for the P and the Q permutations and this constitues the only difference between them.

The `SubBytes` transformation consists of a non-linear substitution applied to every byte separately. It is identical to the Rijndael's `SubBytes` transformation.

`ShiftBytes` cyclically shifts the bytes within a row to the left by a number of positions. In particular, all bytes in row i are moved σ_i positions to the left, where $\sigma = [0, 1, 2, 3, 4, 5, 6, 7]$ for the 256-bit candidate and $\sigma = [0, 1, 2, 3, 4, 5, 6, 11]$ for the 512-bit candidate.

The `MixBytes` transformation, transforms every column of the state matrix independently in a linear way.

The number of rounds is specified to 10 for the 256-bit version and 14 for the 512-bit version.

Bound on the degree of Grøstl's *P* permutation. We will show here that by using Theorem 2 we can bound efficiently the degree of Grøstl's permutation up to 6 rounds, when by using the trivial bound this could be possible only for half of the rounds. As an application we find zero-sum partitions for the permutations P and Q for the 256-bit candidate. As the non-linear transformation used in Grøstl uses the Sbox of Rijndael, one round R of the permutations has degree 7. Then, by applying the Theorem 2 we get that

$$\deg(F \circ R) \leq 512 - \frac{512 - \deg(F)}{7}$$

and

$$\deg(F \circ R^{-1}) \leq 512 - \frac{512 - \deg(F)}{7}.$$

By combining this result with the trivial bound and applying it recursively to the number of rounds we get the following bounds on the degree of $\deg(R^r)$.

The same bounds hold for the inverse transformation.

We can use this new bound in order to find zero-sum partitions for the permutations P and Q, by following the method presented in Proposition 2. As these two permutations differ only in the addition of constants, the technique applied will be the same. That is, let V be a subspace of $\mathbb{F}_2^{512}$ of dimension 509. Then, the sets

$$X_a = \{(R^{-5})(a + z),\ z \in V\},\ \ a \in \mathbb{F}_2^{512}\ ,$$

# rounds	bound on deg(R^r)
1	7
2	49
3	343
4	**487**
5	508
6	511

Table 4. Upper bounds on the degree of several rounds of P and Q.

form a zero-sum partition of size 2^{509} for any of the two Grøstl's permutations.

6.2 Application on the ECHO's inner permutation

The ECHO [2] hash function has been designed by Benadjila et al. for the NIST's SHA-3 competition. It uses the HAIFA mode of operation and it's compression function relies upon a 2048-bit, AES-based permutation P.

The permutation P updates a 2048-bit state, which can be seen as an AES 16×16 state, operating on words of 128 bits. In every round R, three operations modify the state. These are the `BIG SubBytes, BIG ShiftRows` and `BIG MixColumns` transformations. These transformations can be seen, in one way, as generalisations of the three classical AES transformations. In particular,

- `BIG SubBytes` is a non-linear transformation applied independently to every 128-bit cell. It consists of two AES rounds.
- The `BIG ShiftRows` and `BIG MixColumns` transformations are exact analogues of the AES `ShiftRows` and `MixColumns` transformations respectively, with the only difference that they do not operate on bytes but on 128-bit words.

  The number of rounds r, is specified to be 8 for the 256-bit candidate.

The algebraic degree of the permutation P. We will see in this section how the algebraic degree of the permutation P evaluates, as the number of rounds increases. We will show by using the new bound that the degree does not mount the way predicted and reaches its maximum value much later than expected.

The only source of nonlinearity for the P permutation is the `BIG SubBytes` transformation. It can be seen as an application of 16 big 128×128 Sboxes in parallel. Its degree is equivalent to the algebraic degree of the AES transformation reduced to 2 rounds. One round of AES has degree 7 as the Sbox used in AES is of this degree. But two rounds have degree at most 32, as the entries to every AES S-box after the first round, are polynomials of degree 7 in at most 32 variables as a result of the AES diffusion transformation.

By looking now at the ECHO state as a big AES state with 128-bit words instead of bytes, we see that two rounds of P are of degree at most 512, as this is the number of different variables entering every Sbox of the second AES round of the `BIG SubBytes` transformation in the second round of the function.

We know until now that 2 rounds of P are of degree at most 512 both forwards and backwards. Applying the trivial bound to find a bound on the degree of more rounds does not give any satisfactory result. But we will see that by applying Theorem 2, we can get a much better bound for 3 and 4 rounds of the permutation. Here, $n = 2048$ and the degree of every Sbox is at most 32.

So for three rounds of P, if we denote by R^i the iteration by i-times of the round function we get that:

$$\deg(R^3) \leq 2048 - \frac{2048 - 512}{32} \leq 2000,$$

and

$$\deg(R^4) \leq 2048 - \frac{2048 - 2000}{32} \leq 2046.$$

The same bounds hold for the inverse round transformation. So by choosing cosets of a subspace V with dimension 2^{2047} after 4 rounds of the tranformation, we get zero-sum partitions for the entire P permutation.

7 Conclusions

We have found a new bound for the degree of iterated permutations. This improved bound has firstly led to zero-sum distinguishers for the full Keccak-f permutation, while also for the inner permutations of ECHO and Grøstl. Even if the security of the hash functions is not affected, in the case of Keccak our results contradict the so-called hermetic sponge strategy. Additionally, a number of structural properties related to the existence of all-zero higher-order differentials and of zero-sum partitions have been presented for the *Luffa* hash family.

References

1. J.-P. Aumasson and W. Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. Presented at the rump session of Cryptographic Hardware and Embedded Systems - CHES 2009, 2009.
2. R. Benadjila, O. Billet, H. Gilbert, G. Macario-Rat, T. Peyrin, M. Robshaw, and Y. Seurin. SHA-3 Proposal: ECHO. Submission to NIST (Round 2), 2009.
3. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Keccak sponge function family main document. Submission to NIST (Round 2), 2009.
4. C. Boura and A. Canteaut. Zero-sum Distinguishers for Iterated Permutations and Application to Keccak-f and Hamsi-256. In *SAC 2010 - Selected Areas in Cryptography*, Lecture Notes in Computer Science. Springer, 2010. To appear.
5. C. De Cannière, H. Sato, and D.Watanabe. The reasons for the change of Luffa. Supplied with the second round package.
6. C. De Cannière, H. Sato, and D. Watanabe. Hash Function Luffa: Specification. Submission to NIST (Round 1), 2008.
7. C. De Cannière, H. Sato, and D. Watanabe. Hash Function Luffa: Specification. Submission to NIST (Round 2), 2009.
8. A. Canteaut and M. Videau. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 2002.
9. P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and Søren S. Thomsen. Grøstl-a SHA-3 candidate. Submission to NIST (Round 1), 2008.
10. D. Khovratovich, M. Naya-Plasencia, A. Röck, and M. Schläffer. Cryptanalysis of Luffa v2 components. In *SAC 2010 - Selected Areas in Cryptography*, Lecture Notes in Computer Science. Springer, 2010. To appear.
11. L. R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption - FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1995.
12. L.R. Knudsen and V. Rijmen. Known-key distinguishers for some block ciphers. In *Advances in cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.
13. X. Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday.* Kluwer Academic Publishers, 1994.
14. D. Watanabe, Y. Hatano, T. Yamada, and T. Kaneko. Higher Order Differential Attack on Step-Reduced Variants of *Luffa* v1. In *Fast Software Encryption - FSE 2010*, Lecture Notes in Computer Science, pages 270–285. Springer, 2010.