A. V. Kurochkin, On some properties of an XSL-network,
Mat. Vopr. Kriptogr., 2019, Volume 10, Issue 2, 117–124

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ
2019 Т. 10 № 2 С. 117–124

УДК 519.719.2 DOI https://doi.org/10.4213/mvk289

On some properties of an XSL-network

A. V. Kurochkin

Technical Committee for Standardization "Cryptography and security mechanisms" (*TC 26*)*, Moscow, Russia*

Получено 06.II.2018

Abstract. A new type of distinguishing property, named the zero-sum property has been presented by Aumasson and Meier in 2009. It has been applied to the inner permutation of the hash function Keccak. In this paper we present a modified algorithm for zero-sums searching for constructions having an XSL structure, with some restrictions on the linear transformation. The algorithm is applied to a representative of Photon hash functions family.

Key words: zero-sum, algebraic degree, Photon, hash-function

О некоторых свойствах XSL-сетей

А. В. Курочкин

Технический комитет по стандартизации «Криптографическая защита информации», Москва, Россия

Аннотация. В 2009 году Омассон и Мейер предложили новый тип различителя, названный свойством нулевой суммы, и применили его для анализа внутренней подстановки хэш-функции Keccak. В настоящей работе представлен модифицированный алгоритм поиска нулевых сумм для примитивов, построенных на основе XSL схемы с некоторыми ограничениями на линейное преобразование. Рассматривается возможность применения алгоритма к одному из представителей семейства хэш-функций Photon.

Ключевые слова: нулевые суммы, алгебраическая степень, семейство Photon, хэш-функция

Citation: *Mathematical Aspects of Cryptography*, 2019, v. 10, № 2, pp. 117–124 (Russian)

1. Zero-sum

In this section we introduce the notion of zero-sums, describe the algorithm for zero-sums searching and its modification. As an example we apply this algorithm to the hash function Photon-196.

1.1. Intoduction

Definition 1 ([1])**.** Let F be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. A zero-sum for F of size (dimension) K is a subset $\{x_1, \ldots, x_K\} \subset \mathbb{F}_2^n$ of elements which sum to zero and such that their images under F also sum to zero, i. e.

$$\bigoplus_{i=1}^{K} x_i = \bigoplus_{i=1}^{K} F(x_i) = 0,$$

where the sum is defined by the addition in $\mathbb{F}_2^n$ (and in $\mathbb{F}_2^m$), i. e., the bitwise exclusive-or. Since it is expected that a randomly chosen function does not have many zero-sums, the existence of several such sets of inputs may be seen as a distinguishing property of F.

Definition 2. Let $K \subset \mathbb{F}_2^n$ be a subspace. Denote

$$\underline{K} = \{K \oplus a \,|\, a \in \mathbb{F}_2^n\}.$$

That is $\underline{K}$ is the set of all coset classes by K.

Statement 1 ([1])**.** *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ and V be the subspace of dimension $(\deg F) + 1$. Then for every $V' \subseteq \underline{V}$*

$$\bigoplus_{v \in V'} F(x \oplus v) = 0.$$

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 [3]. The only information needed for finding such zero-sums on the iterated permutation is an upper bound on the algebraic degrees of both the round transformation and its inverse. As a rule, the upper bound for the algebraic degree of nonlinearity of the iterative mapping is calculated, as in [4].

More precisely, we suppose that F is a function which operates on an n-bit state, and that F is composed of n_r transformations:

$$F = R_{n_r} \circ \cdots \circ R_1.$$

Let $d_1 < n$ be the algebraic degree of the function composed of the last r_1 transformations, i. e.

$$F_{r_1} = R_{n_r} \circ \cdots \circ R_{n_r - r_1 + 1},$$

and let $d_2 < n$ be the degree of the inverse of the first $r_2 = (n_r - r_1)$ transformations, i. e.

$$G_{r_2} = R_1^{-1} \circ \cdots \circ R_{r_2}^{-1}.$$

Then, we can find many zero-sums of size 2^{d+1} where $d = \max(d_1, d_2)$ as follows:

- Choose a set of $(n - d - 1)$ bits in the intermediate state after r_2 rounds, and fix them to an arbitrary value.
- For each of the 2^{d+1} possible intermediate states z obtained when the nonfixed $(d+1)$ bits take all possible values, compute r_2 rounds backwards in order to obtain the 2^{d+1} input states $x = G_{r_2}(z)$.

The set of these input states is then the zero-sum of size 2^{d+1} of a function of degree d_2 and thus it vanishes. Now, the images of these input states under F correspond to the images of the intermediate states z under F_{r_1}. Then, by computing r_1 rounds forwards, we obtain 2^{d+1} output states. The set of these output states is the zero-sum of size 2^{d+1} of F_{r_1}, of degree less than d. Thus, this sum vanishes, implying that x forms a zero-sum. It is worth noticing that this technique provides several zero-sums having a particular property. Actually, for a given choice of the $(n - d - 1)$ fixed bits in the intermediate state, taking all possible values for the corresponding constant leads to 2^{n-d-1} zero-sums of sizes 2^{d+1} which form a partition of the input space into zero-sums.

1.2. Modified algorithm

In this subsection we introduce new definitions and prove Statement 2 below which allows us to modify the algorithm for zero-sum searching.

Definition 3. Let $\overline{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_2^n$ and $\{n_1, n_2, \ldots, n_t\}$ (where $0 < n_1 < n_2 < \cdots < n_t < n$) be a fixed set of natural numbers which splits the vector $\overline{a}$ into consecutive subvectors, i. e.

$$\overline{a} = (a_0, a_1, \ldots, a_{n_1-1}) || (a_{n_1}, \ldots, a_{n_2-1}) || \ldots || (a_{n_t}, \ldots, a_{n-1}) \stackrel{\text{def}}{=}$$
$$\stackrel{\text{def}}{=} (\overline{a_{0,n_1-1}}, \overline{a_{n_1,n_2-1}}, \ldots, \overline{a_{n_t,n-1}}).$$

Such splitting of vector $\overline{a}$ we shall call *a partition of the form* $(n_1, \ldots, n_t)$, and if all subvectors are of equal length d we denote that partition by (d). If the partition refers to a particular vector, then we write $\overline{a}(n_1, \ldots, n_t)$ or $\overline{a}(d)$ respectively.

Definition 4. Let $V_1 \subseteq \mathbb{F}_2^n$ and $V_2 \subseteq \mathbb{F}_2^n$ be subspaces. We say that $H\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ maintains the structure of the subspace V_1 if for any $V_1' \in \underline{V_1}$ there is a $V_2' \in \underline{V_2}$ such that $H(V_1') = V_2'$; we denote this property by $H(V_1 \to V_2)$.

Definition 5. We say that a subspace $V \subseteq \mathbb{F}_2^n$ is of block type $i_0, i_1, \ldots, i_t$ and is consistent with the partition $(n_1, \ldots, n_t)$ (with $n_0 = 0$ and $n_{t+1} = n$), if

$$V = \{(\overline{a_{n_0, n_1 - 1}}, \overline{a_{n_1, n_2 - 1}}, \ldots, \overline{a_{n_t, n_{t+1} - 1}})\},$$

where

$$\overline{a_{n_j, n_{j+1}-1}} \begin{cases} \text{runs through all possible values from } \mathbb{F}_2^{n_j - n_{j-1}} & \text{if} \quad i_j = 1, \\ = \overline{0} \in \mathbb{F}_2^{n_j - n_{j-1}} & \text{if} \quad i_j = 0, \end{cases}$$

for $j = 0, 1, \ldots, t$. Denote such subspaces by $V^{(n_1, \ldots, n_t)}(i_0, i_1, \ldots, i_t)$ and if the partition has the form (d), then by $V^{(d)}(i_0, i_1, \ldots, i_t)$.

Definition 6. We say that a transformation $G\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ has a block structure if there exists at least one subspace of block type V concordant with the partition $(n_1, \ldots, n_t)$ such that

$$G(V \to V'), \text{ where } V' \text{ is a subspace of block type concordant with some partition } (m_1, \ldots, m_l),\ l \leqslant t.$$

Statement 2. *Let* $G\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ *be an iterative transformation of the form*

$$G = G_1 \circ G_2 \circ \cdots \circ G_{2 \cdot t}, \quad \textit{where} \quad G_i = X_i \circ S \circ L, \qquad i = \overline{1, 2t}.$$

For the calculation of the zero-sum the values of the round constants do not matter, so it is possible to put $X_i = X$, $i = \overline{1, 2t}$. *Then if there exist subspaces* $V, V' \subset \mathbb{F}_2^n$ *such that*

$$S \circ L \circ X \circ S(V \to V'),$$

then in order to find the zero-sum it is sufficient to construct a subspace of smaller dimension.

Let us prove this statement. Let nl denote the algebraic degree of the function; put

$$h_1 = \mathrm{nl}(G_1 \circ \cdots \circ G_t), \qquad h_2 = \mathrm{nl}(G_{t+1} \circ \cdots \circ G_{2\cdot t}),$$
$$h_1' = \mathrm{nl}(G_1 \circ \cdots \circ G_{t-1}), \qquad h_2' = \mathrm{nl}(G_{t+2} \circ \cdots \circ G_{2\cdot t}),$$
$$d = \max(h_1, h_2), \qquad d' = \max(h_1', h_2').$$

Using the algorithm for finding zero-sums from Section 1 it is easy to see that the complexity of constructing the zero-sum became $2^{d'}$, instead of 2^d.

Remark. For several cryptoalgorithms (for example Photon, Streebog, Stribob, AES) one can easily show that $S \circ L \circ X \circ S$ transformation is of a block type. For cryptoalgorithms Kuznyechik, Present it is not so easy to demonstrate this property.

1.3. Sponge construction

Sponge functions have been introduced by Bertoni et al. [5] as a new way for constructing hash functions from a fixed permutation. The internal t-bit state of S is composed of the c-bit capacity and the r-bit bitrate $(t = c+r)$, is initialized with some fixed value. Then, after being appropriately padded the message is splitted into r-bit chunks. These r-bit chunks of the message are iteratively processed by XORing them to the bitrate part of the internal state and then applying the permutation P acting on the set of t-bit vectors. Once all message chunks have been handled by this absorbing phase, one successively outputs r bits of the final hash value by extracting them from the bitrate part of the internal state and then applying the permutation P on the internal state (squeezing process).

1.4. Photon

Photon is a family of sponge-like hash function proposals that was recently standardized by ISO. Authors define an AES-like function to be a fixed key permutation P applied on an internal state of d^2 elements of s bits each, which may be represented as a $(d \times d)$ matrix. The permutation P is composed of 12 rounds, each containing four layers as depicted in Fig. 1: AddConstants (Add), SubCells (S), ShiftRows (Row), and MixColumnsSerial(Mix).

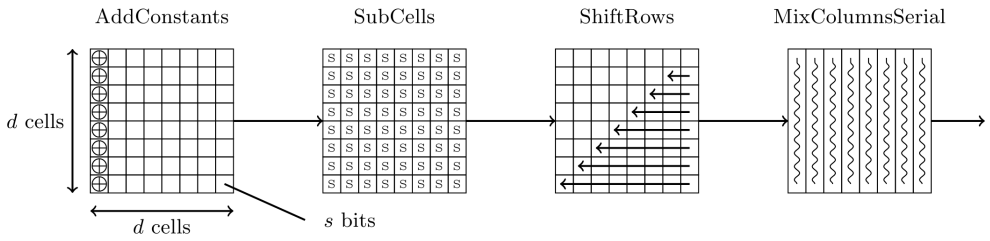

Fig. 1

Table 1 shows the main parameters of the Photon hash function family.

Table 1

	t	d	s	N_r	$IC_d(\cdot)$	irr.polynomial	Z_i coefficients
P_{100}	100	5	4	12	$[0,1,3,6,4]$	x^4+x+1	$(1,2,9,9,2)$
P_{144}	144	6	4	12	$[0,1,3,7,6,4]$	x^4+x+1	$(1,2,8,5,8,2)$
P_{196}	196	7	4	12	$[0,1,2,5,3,6,4]$	x^4+x+1	$(1,4,6,1,1,6,4)$
P_{256}	256	8	4	12	$[0,1,3,7,15,14,12,8]$	x^4+x+1	$(2,4,2,11,2,8,5,6)$
P_{288}	288	6	8	12	$[0,1,3,7,6,4]$	$x^8+x^4+x^3+x+1$	$(2,3,1,2,1,4)$

Table 2 shows the degree of nonlinearity, depending on the number of rounds.

Table 2

number of rounds	1	2	3	4	5	6	7	8	9
P_{100}	3	9	27	75	91	97	99	99	99
P_{144}	3	9	27	81	123	137	141	143	143
P_{196}	3	9	27	81	157	183	191	194	195
P_{256}	3	9	27	81	197	236	249	253	255
P_{288}	7	42	252	282	287	287	287	287	287

Transformation P (here $G_1,\ldots,G_{12}$ corresponds to rounds of Photon; G_6 and G_7 are given explicitly):

$$P = G_1 \circ \cdots \circ G_5 \circ Add_6 \circ S \circ Row \circ Mix \circ Add_7 \circ S \circ Row \circ Mix \circ G_8 \circ \cdots \circ G_{12}.$$

It is easy to show that substutition $H = S \circ Row \circ Mix \circ Add_7 \circ S$ has a block structure $H(V_1 \to V_2)$, where

$$V_1 = V^{(4)}(i_0,\ldots,i_{48}), \qquad i_j = \begin{cases} 1, & \text{if } j = 0 \mod (8), \\ 0, & \text{if } j \neq 0 \mod (8), \end{cases} \qquad j = \overline{0,48},$$

$$V_2 = V^{(4)}(i_0,\ldots,i_{48}), \qquad i_j = \begin{cases} 1, & \text{if } \ j = 0 \mod (7), \\ 0, & \text{if } \ j \neq 0 \mod (7), \end{cases} \qquad j = \overline{0,48}.$$

According to Table 2 we have

$$\mathrm{nl}(G_1 \circ \cdots \circ G_6) = \mathrm{nl}(G_7 \circ \cdots \circ G_{12}) = 183$$

and

$$\mathrm{nl}(G_1 \circ \cdots \circ G_5) = \mathrm{nl}(G_8 \circ \cdots \circ G_{12}) = 157.$$

It's obvious that (Fig. 2) $H(V_1 \to V_2)$, where

$$V_1 = V^{(4)}(i_0, \ldots, i_{48}), \qquad i_j = \begin{cases} 0, & \text{if} \quad j = 0 \mod (8), \\ 1, & \text{if} \quad j \neq 0 \mod (8), \end{cases} \qquad j = \overline{0, 48},$$

$$V_2 = V^{(4)}(i_0, \ldots, i_{48}), \qquad i_j = \begin{cases} 0, & \text{if} \quad j = 0 \mod (7), \\ 1, & \text{if} \quad j \neq 0 \mod (7), \end{cases} \qquad j = \overline{0, 48}.$$

In Fig. 2 white color indicates 4-bit words which take all possible values, and black color indicates 4-bit words whose values are fixed.

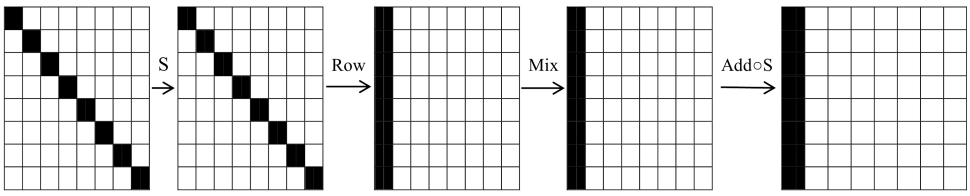

Fig. 2

Using statement 1 and the fact that the dimension of the subspace V_1 is

$$196 - 4 \cdot 7 = 168,$$

one can assert that we have found zero-sums of dimension 168 instead of the declared 183.

Also for the hash function $Photon(256)$ using this algorithm, the complexity of finding zero sums may be reduced from 2^{236} to 2^{224}.

Note that similar results for hash functions were obtained in [6] and [7]. The results of our paper were obtained independently.

2. Conclusion

A modified algorithm for constructing zero-sums was proposed in the paper. The algorithm is applicable to cryptographic algorithms of a specific type and may be applied to the synthesis of cryptographic primitives.

Note that these results may be used for the analysis of block ciphers, but in such cases the construction of zero-sums can not be started from the intermediate states of the algorithm.

References

[1] Aumasson J.-P., Meier W., “Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi”, Presented at the rump session of CHES, 2009.

[2] Guo J., Peyrin T., Poschmann A., “The PHOTON family of lightweight hash function”, CRYPTO 2011, Lect. Notes Comput. Sci., **6841**, 2011, 222–239.

[3] Daemen J., Knudsen L. R., Rijmen V., “The block cipher Square”, FSE 1997, Lect. Notes Comput. Sci., **1267**, 1997, 149–165.

[4] Boura C., Canteaut A., De Canniere C., “Higher-order differential properties of Keccak and Luffa”, FSE 2011, Lect. Notes Comput. Sci., **6733**, 2011, 252–269.

[5] Beroni G., Daemen J., Peeters M., Van Assche G., “Sponge function”, Ecrypt Hash Workshop, 2007.

[6] Wang Q., Grassi L., Rechberger C., “Zero-sum partitions of PHOTON permutations”, Topics in Cryptology – CT-RSA, Lect. Notes Comput. Sci., **10808**, 2018, 279–299.

[7] Dong L., Wu W.-L., Wu S., Zhou J., “Novel method of constructing the zero-sum distinguishers”, *J. Communications*, **33**:11 (2012), 91–99.