On APN functions EA-equivalent to permutations

Valeriya Idrisova

Sobolev Institute of Mathematics, Novosibirsk State University,
Academgorodok, Novosibirsk, Russia

BFA-2017, Os, Norway

Definitions

A *vectorial Boolean function* is an arbitrary mapping F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. Every vectorial function can be represented as set of m *coordinate* Boolean functions in n variables: $F = (f_1, ..., f_m)$.

A vectorial function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ is called *2-to-1 function* if it's vector of values consists of 2^{n-1} different elements and F takes every value twice.

In this work we consider the case $m = n$.

Definitions

A vectorial function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ is called an *APN function* if, for every nonzero a and every b in $\mathbb{F}_2^n$, the equation $F(x) + F(x + a) = b$ has at most two solutions.

The notion of an APN function function was proposed by K. Nyberg [1]. It is also known that APN functions, in particular, inverse function $F(x) = x^{2^n - 2}$, were investigated starting from 1968 by V. Bashev and B. Egorov in USSR.

[1]Nyberg K. Differentially uniform mappings for cryptography // Eurocrypt 1993, Lecture Notes in Computer Science, 1994 V. 765. P. 55–64.

The Big APN problem

APN functions cause a great interest, and many articles are devoted to studying their properties, but there are still a lot of interesting open problems. State of art in the area of APN functions and reviews of opened questions can be found, for example, in the following sources [2], [3]

[2] Carlet C. Open Questions on Nonlinearity and on APN Functions (Proc. of the 5th International Workshop WAIFI 2014, Gebze, Turkey, September, 2014).// Lecture Notes in Computer Science, 2015, Vol. 9061, P. 83–107.

[3] Budaghyan L. Construction and Analysis of Cryptographic Functions. Springer International Publishing, 2014.

The Big APN problem

One of the most interesting problems in this area is constructing bijective APN functions in even dimensions. There was a conjecture that such functions do not exist (it was proved for $n = 4$), but in 2009 J.F.Dillon et al.[4] presented the first APN permutation for n $= 6$.

This question is still open for the greater dimensions and it is referred as **"The Big APN problem"**.

[4] McQuistan M. T., Wolfe A. J., Browning K. A., Dillon J. F. An apn permutation in dimension six.// American Mathematical Society, 2010 V. 518. P. 33–42.

The Big APN problem

Many interesting approaches in investigations of this problem. were proposed. One of them, using decomposition of S-boxes, lead to new APN permutations, CCZ-equivalent to the found by Dillon et.al.[5]

The first APN permutation was constructed using non-bijective CCZ-equivalent APN function (so-called Kim function). In this work we investigate special functions EA-equivalent to permutations. More precisely, we consider 2-to-1 APN functions F such that $F + L$ is a permutation for some linear functions L.

---

[5] Perrin L., Udovenko A., Biryukov A. Cryptanalysis of a Theorem: Decomposing the Only Known Solution to the Big APN Problem.// Advances in Cryptology – CRYPTO 2016. CRYPTO 2016. Lecture Notes in Computer Science, vol 9815. Springer

2-to-1 functions

Theorem 1. For every 2-to-1 vectorial Boolean function F in n variables there exists at least one vectorial Boolean function G such that every coordinate Boolean function of G is balanced or constant and $H = F + G$ is a permutation.

This fact implies the following. If F is an APN function and G is affine, then H is an APN permutation, since F and H are EA-equivalent.

The algorithm

In this work we present an algorithm for searching 2-to-1 APN functions. This algorithm can be divided into two steps.

On the first step we obtain symbol sequences that potentially represents the vector of values for some 2-to-1 APN function.

On the second step we put binary vectors in correpondence to the symbols in the generated sequences such that obtained 2-to-1 functions are APN.

The algorithm

The first step.

Consider the vector of values of an arbirtrary 2-to-1 vectorial function. The definition of an APN function implies certain restrictions on its structure. In particular, for any non-zero $a \in \mathbb{F}_2^n$ and any different x_1 and x_2 from $\mathbb{F}_2^n$ such that $x_1 + a \neq x_2$ the following relation holds $F(x_1 + a) + F(x_1) \neq F(x_2 + a) + F(x_2)$.

The algorithm

On the first step of the algorithm we build all possible symbol sequences, satisfying the restrictions mentioned above. Let us call them *admissible sequences.*

For example, the sequence $\alpha\ \alpha\ \beta\ \beta\ \theta\ \epsilon\ \theta\ \epsilon$ is not admissible, since for $a = 001$ holds $F(000 + 001) + F(000) = \alpha + \alpha = 000$ and $F(010 + 001) + F(010) = \beta + \beta = 000$, that contradicts these restrictions.

The algorithm

Let us consider lexicographically ordered sequence $\alpha_1, \alpha_1, \alpha_2, \alpha_2, \cdots, \alpha_{2^{n-1}}, \alpha_{2^{n-1}}$ whose elements would form the admissible sequences.

Let us denote the set of all admissible sequences of the length 2^n by M_n. As a first symbol of the first sequence let us take an element α_1. On j-th step, $j = 1, ..., 2^n - 1$, for every sequence from M_n of length j we build all possible sequences of length $j + 1$ adding a new element, such that the following two conditions hold:

The algorithm

1. The added element coincides with previous j elements of considered sequence, or it is lexicographically the smallest elements amongst new elements.

2. Let i_1 and i_2 be the different natural numbers, denoting positions in obtained sequence of length $j+1$ where $1 \leqslant i_1, i_2 \leqslant j+1$. Let x_{i_1} and x_{i_2} — be the corresponding binary representations of i_1 and i_2. Then for all non-zero vectors a of length n the pair of symbols on positions x_{i_1} and $x_{i_1}+a$, and the pair of symbols on positions x_{i_2} and $x_{i_2}+a$, are different (when $x_{i_1} \neq x_{i_2}+a$).

Sequences obtained on j-th step of the length $j+1$ are added into M_n, initial sequence of length j is deleted. This step of the algorithm finishes when all the sequences in M_n have length 2^n.

Examples of generated symbol sequences

For $n = 3$: $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_3\ \alpha_4\ \alpha_2\ \alpha_4\ \alpha_1)$

For $n = 4$: $(\alpha_1\ \alpha_1\ \alpha_2\ \alpha_3\ \alpha_2\ \alpha_4\ \alpha_3\ \alpha_5\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ \alpha_7\ \alpha_8\ \alpha_6\ \alpha_8)$

For $n = 5$: $(\alpha_1\ \alpha_2\ \alpha_1\ \alpha_3\ \alpha_2\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ \alpha_8\ \alpha_9\ \alpha_{10}\ \alpha_9\ \alpha_{11}$
$\alpha_{12}\ \alpha_4\ \alpha_3\ \alpha_8\ \alpha_{13}\ \alpha_{14}\ \alpha_{15}\ \alpha_{15}\ \alpha_{11}\ \alpha_{16}\ \alpha_6\ \alpha_{12}\ \alpha_5\ \alpha_{10}\ \alpha_7\ \alpha_{14}\ \alpha_{16}\ \alpha_{13})$

For $n = 6$: $(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ \alpha_8\ \alpha_3\ \alpha_5\ \alpha_9\ \alpha_9\ \alpha_{10}\ \alpha_6\ \alpha_{11}$
$\alpha_1\ \alpha_{10}\ \alpha_2\ \alpha_4\ \alpha_7\ \alpha_{12}\ \alpha_8\ \alpha_{12}\ \alpha_{13}\ \alpha_{14}\ \alpha_{13}\ \alpha_{11}\ \alpha_{14}\ \alpha_{15}\ \alpha_{16}\ \alpha_{17}\ \alpha_{18}\ \alpha_{19}$
$\alpha_{20}\ \alpha_{21}\ \alpha_{22}\ \alpha_{23}\ \alpha_{24}\ \alpha_{18}\ \alpha_{19}\ \alpha_{25}\ \alpha_{24}\ \alpha_{20}\ \alpha_{26}\ \alpha_{27}\ \alpha_{28}\ \alpha_{29}\ \alpha_{30}\ \alpha_{29}$
$\alpha_{31}\ \alpha_{30}\ \alpha_{28}\ \alpha_{31}\ \alpha_{32}\ \alpha_{32}\ \alpha_{25}\ \alpha_{26}\ \alpha_{22}\ \alpha_{27}\ \alpha_{21}\ \alpha_{23}\ \alpha_{16}\ \alpha_{15}\ \alpha_{17})$

The algorithm

The second step.

To get 2-to-1 an APN function we assign binary vectors to the symbols from the obtained sequences on the second step. In general, we need to choose 2^{n-1} vectors from $\mathbb{F}_2^n$ and put in correspondence with each from 2^{n-1} symbols in the considered admissible sequence.

For $n = 3$ there are the following property, that allow to obtain all possible 2-to-1 APN functions:

Lemma 1. An admissible sequence with assigned vectors b_1, b_2, b_3, b_4 from $\mathbb{F}_2^3$ is 2-to-1 APN function if and only if for these vectors the following relation holds $b_1 + b_2 + b_3 + b_4 \neq 0$.

The algorithm

For larger dimensions the condition $b_{i_1} + b_{i_2} + b_{i_3} + b_{i_4} \neq 0$ for every four vectors of chosen 2^{n-1} vectors could have been also sufficient for obtaining APN function, but the following statement holds for $n \leqslant 6$:

Lemma 2. For any subset $K = \{b_1, ..., b_{2^{n-1}}\}$ in $\mathbb{F}_2^n$ there exist the set of indices i_1, i_2, i_3, i_4 such that the sum $b_{i_1} + b_{i_2} + b_{i_3} + b_{i_4}$ is equal to zero.

The algorithm

The exhaustive search through all possible sets of vectors can be divided into two parts. The first one is to choose 2^{n-1} vectors from $\mathbb{F}_2^n$. The second is to search through all possible permutations for every chosen set of vectors.

There is the conjecture that allow us to reduce the second step in this search.

Hypothesis 1. If for all $\binom{2^n}{2^{n-1}}$ lexicographically ordered sets of vectors the given admissible sequence is not APN then there is no 2-to-1 APN function with such a structure of vector of values.

If the conjecture is true then the following statement holds:

Hypothesis 2. There is no 2-to-1 APN functions in dimension 4.

Examples for $n = 5$

We have found for $n = 5$ the examples of 2-to-1 functions EA-equivalent to all known permutations (up to affine equivalence):

$F_1 =$ (0 23 5 21 12 31 0 14 8 17 5 7 17 9 26 7 12 15 21 15 8 28 27 9 28 27 22 26 23 22 31 14)

$F_2 =$ (0 5 29 31 24 23 9 16 5 15 10 4 12 16 23 30 26 4 30 14 31 24 22 14 22 9 15 29 0 26 12 10)

$F_3 =$ (0 27 25 5 11 26 30 25 2 12 0 29 17 27 12 4 11 4 29 24 26 2 18 17 24 10 30 18 5 14 14 10)

$F_4 =$ (0 16 27 12 12 22 6 27 6 24 3 26 30 10 10 25 0 3 18 22 26 19 25 23 30 19 18 24 16 23 13 13)

$F_5 =$ (0 29 26 0 6 17 13 29 3 16 4 16 18 11 4 26 1 14 7 15 20 17 3 1 15 14 20 18 13 6 7 11)

Examples for $n = 5$

Corresponding linear functions such that the sum $F_i + L_i$ is a permutation:

$$L_1 = (x_5, x_1+x_2+x_3, x_2+x_3+x_4+x_5, x_1+x_3+x_4+x_5, x_1+x_2+x_3+x_4)$$

$$L_2 = (x_1 + x_3 + x_4, x_1 + x_3 + x_4, x_1 + x_4 + x_5, x_3 + x_4, x_3 + x_4)$$

$$L_3 = (x_4 + x_5, x_1 + x_3 + x_4 + x_5, x_1 + x_2, x_2 + x_4 + x_5, x_1 + x_2 + x_4)$$

$$L_4 = (x_4 + x_5, x_3 + x_4, x_1 + x_3, x_1 + x_2 + x_3, x_2 + x_3 + x_4 + x_5)$$

$$L_5 = (x_4 + x_5, x_4 + x_5, x_1 + x_2 + x_3 + x_5, x_1 + x_2, x_1 + x_3)$$

Examples for $n = 6$

An example of 2-to-1 APN-function that is EA-equvalent to APN-permutation (Dillon et.al.):

$F = (54\ 52\ 48\ 57\ 14\ 39\ 34\ 0\ 63\ 45\ 45\ 0\ 2\ 33\ 32\ 28\ 55\ 1\ 6\ 46\ 5\ 46\ 28\ 8\ 37\ 57\ 5\ 19\ 2\ 25\ 48\ 32\ 17\ 54\ 58\ 58\ 33\ 1\ 34\ 14\ 51\ 21\ 8\ 29\ 55\ 12\ 30\ 29\ 27\ 19\ 21\ 37\ 17\ 40\ 63\ 52\ 40\ 27\ 51\ 12\ 6\ 30\ 39\ 25)$

Corresponding linear function such that the sum $F + L$ is a permutation:

$L = (x_1 + x_2 + x_6, x_1 + x_2 + x_6, x_1 + x_2 + x_4 + x_6, x_1 + x_2 + x_6, x_1 + x_2 + x_4 + x_6, x_4 + x_6)$

Further research

Our further research will be devoted to the following open questions:

1. To find conditions on the linear functions that can give APN permutations from 2-to-1 APN functions.

2. To study the existence of iterative constructions of APN permutations based on 2-to-1 functions.

3. To find new APN-functions that are not CCZ-equivalent to the known classes, using this approach.

Thank you for your attention!