Math-Net.Ru

Общероссийский математический портал

А. В. Menyachikhin, Изменение корреляционных и разностных характеристик подстановки при ее умножении на транспозицию, *Матем. вопр. криптогр.*, 2020, том 11, выпуск 2, 111–123

DOI: <https://doi.org/10.4213/mvk325>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 35.173.219.12

26 апреля 2023 г., 14:04:03

The change in linear and differential characteristics of substitution after the multiplication by transposition

A. V. Menyachikhin

TVP Laboratories, Moscow

Получено 11.XI.2019

Abstract. It is shown that the linearity and differential uniformity of the substitution multiplied by transposition may be calculated with time complexities $O(2^{2n})$ and $O(2^n)$ respectively. Some heuristic algorithms of constructing s-boxes are optimized in this paper.

Keywords: s-box, substitution, heuristic optimization, linearity, linear spectrum, linear approximation table, differential uniformity, differential spectrum, difference distribution table

Изменение корреляционных и разностных характеристик подстановки при ее умножении на транспозицию

А. В. Менячихин

Лаборатории ТВП, Москва

Аннотация. Установлена связь между элементами корреляционных и между элементами разностных матриц подстановок, находящихся друг от друга на расстоянии Кэли, равном единице. Для таких подстановок предложены алгоритмы эффективного вычисления линейного и разностного спектров. Снижены оценки трудоемкости ряда методов алгоритмической оптимизации s-боксов.

Ключевые слова: s-бокс, подстановка, алгоритмическая оптимизация, линейная характеристика, линейный спектр, корреляционная матрица, разностная характеристика, разностный спектр, разностная матрица

1. Introduction

Constructing s-boxes with excellent cryptographic properties is one of the important problems in modern cryptography. One approach to solve this problem is based on heuristic optimization of some given s-box. The heuristic optimization methods include genetic algorithms (see [3, 9]), hill climbing methods (see [7, 8]), methods of gradient descent (see [6]), spectral-linear and spectral-differential methods (see [4, 5]).

The main problem of using heuristic methods is the high level of their time complexity. The δ_g -parameter and the p_g -parameter of s-box g are the most difficult to calculate. In this paper we introduce new techniques for calculating linearity and differential uniformity of the substitution $h \in S(V_n)$ such that $h = (x, y)g$, where $x, y \in V_n$, $g \in S(V_n)$, in terms of the same parameters of the substitution g .

The paper is organized as follows. Section 2 contains the basic definitions and notation. In section 3 we derive the main propositions and algorithms. The cryptographic applications of the results are discussed in Section 4. Finally, Section 5 presents our conclusions.

2. Definitions and notation

Let $V_n(2) = V_n$ be n -dimensional vector space over the field $\mathbb{F}_2$. Suppose that $V_n^\times = V_n \setminus \{0\}$. Let $S(V_n)$ be the symmetric group on set of 2^n elements. The cardinality of a set A is usually denoted by $|A|$.

We shall use the following operations and notation:

- exclusive-OR (or XOR) is denoted by $\oplus$,
- logical AND is denoted by $\wedge$,
- the scalar product of two elements $x = (x_{n-1}, \dots, x_0)$ and $y = (y_{n-1}, \dots, y_0)$ of $\mathbb{F}_2^n$ is denoted by $\circ$ and is equal to $x \circ y = \bigoplus_{i=0}^{n-1} x_i \wedge y_i$.

Now, we give some basic definitions.

Definition 1. The *linearity* of s-box g is defined as the absolute value of the bias:

$$\delta_g = \max_{\alpha, \beta \in V_n^\times} \delta_{\alpha, \beta}^g,$$

where

$$\delta_{\alpha, \beta}^g = 2^{1-n} \cdot |\{x \in V_n \mid x \circ \alpha = g(x) \circ \beta\}| - 1.$$

The *Linear Approximation Table* (LAT) of s-box g is a $2^n \times 2^n$ matrix T_1 such that $T_1(\alpha, \beta) = \delta_{\alpha, \beta}^g$.

S-boxes with small value of δ_g -parameter offer better resistance against linear attacks.

Definition 2. The *differential uniformity* of s-box g is defined as

$$p_g = \max_{\alpha, \beta \in V_n^\times} p_{\alpha, \beta}^g,$$

where

$$p_{\alpha, \beta}^g = 2^{-n} \cdot |\{x \in V_n \mid g(x \oplus \alpha) \oplus g(x) = \beta\}|.$$

The *Difference Distribution Table* (DDT) of s-box g is a $2^n \times 2^n$ matrix T_2 such that $T_2(\alpha, \beta) = p_{\alpha, \beta}^g$.

S-boxes designed for cryptographic primitives must have a low p_g -parameter value to provide high resistance to differential attacks.

According to [4] we define the linear and the differential spectra of substitution g .

For $g \in S(V_n)$ and for elements $p \in P_{n-1}$ and $\delta \in P_{n-2}$, where

$$P_j = \left\{ \frac{i}{2^j} \mid i = 0, 1, \dots, 2^j \right\}, \quad |P_j| = 2^j + 1, \quad j \in \{1, 2, \dots\},$$

we define the sets

$$D(g, p) = \{(\alpha, \beta) \in V_n^\times \times V_n^\times \mid p_{\alpha, \beta}^g = p\},$$

$$L(g, \delta) = \{(\alpha, \beta) \in V_n^\times \times V_n^\times \mid |\delta_{\alpha, \beta}^g| = \delta\}.$$

Definition 3. The *differential spectrum* of s-box g is defined as

$$D(g) = \{(p, |D(g, p)|) \mid p \in P_{n-1}\}, \quad |D(g)| = 2^{n-1} + 1.$$

Definition 4. The *linear spectrum* of s-box g is defined as

$$L(g) = \{(\delta, |L(g, \delta)|) \mid \delta \in P_{n-2}\}, \quad |L(g)| = 2^{n-2} + 1.$$

Definition 5. Two s-boxes g and h are called *affine equivalent* if there exist two invertible $n \times n$ matrices A, B over the field $\mathbb{F}_2$, and elements $a, b \in V_n$ such that $h(x) = B(g(A(x) \oplus a)) \oplus b$.

3. Main results

This section deals with the change in linear and differential characteristics of substitution multiplied by transposition. This issue has been

studied in [10]. The authors had showed that for $h = (x, y)g$ such that $g, h: V_n \rightarrow V_n$ we get:

$$\delta_g - 2^{2-n} \leq \delta_h \leq \delta_g + 2^{2-n},$$

$$p_g - 2^{2-n} \leq p_h \leq p_g + 2^{2-n}.$$

In [7] the similar properties of Boolean functions are used to optimize the hill climbing methods.

In this section we describe two new algorithms for calculating linearity and differential uniformity of substitution $h \in S(V_n)$ such that $h = (x, y)g$. We also formally prove the correctness of new algorithms and study their time complexity.

3.1. Efficient computation of elements in the linear approximation table

The first subsection deals with the relationship between elements of linear approximation tables for the substitutions $g, h \in S(V_n)$ such that $h = (x, y)g$.

Algorithm 1.

- Input. Substitution $g \in S(V_n)$; the elements $x, y \in V_n$;
the LAT $T_1(g)$; the linear spectrum $L(g)$.
- Step 1. For each element $i = 0, \dots, 2^n - 1$ do the following operations:
- calculate elements $\alpha = x \oplus y$ and $\beta = g(x) \oplus g(y)$,
 - if $\alpha \circ i \neq 0$ then add i to the list I_1 ,
 - if $\beta \circ i \neq 0$ then add i to the list I_2 .
- Step 2. For each ordered pair $(\alpha, \beta) \in I_1 \times I_2$ do the following operations:
- decrement the value $|L(g, |\delta_{\alpha, \beta}^g|)|$ by 1,
 - change the value $\delta_{\alpha, \beta}^g$ by $(-1)^{\alpha \circ x \oplus \beta \circ g(x) \oplus 1} \cdot 2^{2-n}$,
 - increment the value $|L(g, |\delta_{\alpha, \beta}^g|)|$ by 1.
- Step 3. The algorithm stops after calculating $h = (x, y)g$ and $L(h) = L(g)$.

Output. Substitution $h \in S(V_n)$ such that $h = (x, y)g$;
the linear spectrum $L(h)$.

The correctness of the algorithm 1 is presented in the first proposition.

Proposition 1. For substitutions $g, h \in S(V_n)$ such that $h = (x, y)g$ we have

$$\delta_{\alpha, \beta}^h - \delta_{\alpha, \beta}^g = \begin{cases} 0, & \text{if either } (x \oplus y) \circ \alpha = 0 \text{ or} \\ & (g(x) \oplus g(y)) \circ \beta = 0, \\ (-1)^{\alpha \circ x \oplus \beta \circ g(x) \oplus 1} \cdot 2^{2-n} & \text{in the opposite case.} \end{cases}$$

Proof. Consider the following sequence of equalities

$$\begin{aligned} \delta_{\alpha, \beta}^h - \delta_{\alpha, \beta}^g &= (|\{z \mid z \circ \alpha = h(z) \circ \beta\}| - |\{z \mid z \circ \alpha = g(z) \circ \beta\}|) \cdot 2^{1-n} \\ &= \left(\sum_{z \in V_n} (1 - z \circ \alpha \oplus h(z) \circ \beta) - \sum_{z \in V_n} (1 - z \circ \alpha \oplus g(z) \circ \beta) \right) \cdot 2^{1-n} \\ &= (x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(y) \circ \beta + y \circ \alpha \oplus g(y) \circ \beta - y \circ \alpha \oplus g(x) \circ \beta) \cdot 2^{1-n}. \end{aligned}$$

It is easy to show that if $x \circ \alpha = y \circ \alpha$ or $g(x) \circ \beta = g(y) \circ \beta$, then $\delta_{\alpha, \beta}^h - \delta_{\alpha, \beta}^g = 0$.

Suppose now that

$$\begin{cases} x \circ \alpha \neq y \circ \alpha, \\ g(x) \circ \beta \neq g(y) \circ \beta, \end{cases} \text{ i. e. } \begin{cases} y \circ \alpha = (x \circ \alpha) \oplus 1, \\ g(y) \circ \beta = (g(x) \circ \beta) \oplus 1. \end{cases}$$

Note that in this case we have

$$\begin{aligned} &\left(\underbrace{x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(x) \circ \beta \oplus 1}_{(-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1}} \right. \\ &\quad \left. + \underbrace{x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(x) \circ \beta \oplus 1}_{(-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1}} \right) \cdot 2^{1-n} \\ &= (-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1} \cdot 2^{2-n}. \end{aligned}$$

This completes the proof. □

Let us denote by t_1 the complexity of Algorithm 1.

Proposition 2. *If $n \rightarrow \infty$ then*

$$t_1 = O(2^{2n}).$$

Proof. It is easy to see that $|I_1| = |I_2| = 2^{n-1}$. The complexity of the algorithm is the product of the number $|I_1| \cdot |I_2| = 2^{2n-2}$ of iterations of step 2 and the constant number of operations on the step 2 of the algorithm. $\square$

Remark 1. The algorithm 1 is approximately n times faster than the classical algorithm of calculating the linearity.

Example 1. Consider the following substitutions $g, h \in S(V_3)$ such that $h = (1, 2)g$:

$$g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 1 & 0 & 3 & 5 & 6 & 7 \end{pmatrix}, \quad h = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 4 & 0 & 3 & 5 & 6 & 7 \end{pmatrix}.$$

The results obtained by algorithm 1 are presented in Table 1 (all changing elements of the matrix are in bold).

In particular, using proposition 1 we obtain

1) Since $(1 \oplus 2) \circ 3 = 0$, then for any $\beta \in V_3$ it follows that

$$\delta_{3,\beta}^h - \delta_{3,\beta}^g = 0.$$

2) $\delta_{1,1}^h - \delta_{1,1}^g = (-1)^{(1\circ 1) \oplus (1\circ 4) \oplus 1} \cdot 2^{2-3} = 1/2$.

3) $\delta_{2,3}^h - \delta_{2,3}^g = (-1)^{(2\circ 1) \oplus (3\circ 4) \oplus 1} \cdot 2^{2-3} = -1/2$.

Table 1

Substitution g								Substitution h							
2 4 1 0 3 5 6 7								2 1 4 0 3 5 6 7							
LAT of g								LAT of h							
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	$\frac{1}{2}$	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0
0	0	0	0	0	0	-1	0	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	0
0	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$-\frac{1}{2}$
0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
0	$\frac{1}{2}$	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	$-\frac{1}{2}$
0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	$\frac{1}{2}$
δ	0	$\frac{1}{2}$		1				δ	0	$\frac{1}{2}$		1			
$ L(g, \delta) $	24	24		1				$ L(g, \delta) $	21	28		0			

3.2. Efficient computation of the difference distribution table elements

The second subsection deals with the relationship between elements of difference distribution tables for the substitutions $g, h \in S(V_n)$ such that $h = (x, y)g$.

Algorithm 2.

- Input. Substitution $g \in S(V_n)$; the elements $x, y \in V_n$;
the DDT $T_2(g)$; the differential spectrum $D(g)$.
- Step 1. For each element $\alpha = 1, \dots, 2^n - 1$ such that $\alpha \neq x \oplus y$ do the following operations:
calculate elements $\beta_0 = g(x) \oplus g(x \oplus \alpha)$, $\beta_2 = g(y) \oplus g(y \oplus \alpha)$ and:
if $\beta_0 = \beta_2$, then
- calculate element $\beta_1 = g(y) \oplus g(x \oplus \alpha)$;
 - for each element $i = 0, 1$ change the values:
decrease the value $|D(g, p_{\alpha, \beta_i}^g)|$ by 1,
increase the value $|D(g, p_{\alpha, \beta_i}^g + 4 \cdot (-1)^{i+1})|$ by 1;
- else if $\beta_0 \neq \beta_2$, then
- calculate elements $\beta_1 = g(y) \oplus g(x \oplus \alpha)$,
 $\beta_3 = g(x) \oplus g(y \oplus \alpha)$ and:
 - for each element $i = 0, 1, 2, 3$ change the values:
decrease the value $|D(g, p_{\alpha, \beta_i}^g)|$ by 1,
increase the value $|D(g, p_{\alpha, \beta_i}^g + 2 \cdot (-1)^{i+1})|$ by 1.
- Step 2. The algorithm stops after calculating $h = (x, y)g$ and $D(h) = D(g)$.
- Output. Substitution $h \in S(V_n)$ such that $h = (x, y)g$;
the differential spectrum $D(h)$.

Let us introduce the indicator function

$$I_\beta(x) = \begin{cases} 1, & \text{if } \beta = x, \\ 0, & \text{if } \beta \neq x, \end{cases} \quad \text{where } \beta, x \in V_n.$$

The correctness of the algorithm 2 follows from the following proposition.

Proposition 3. For substitutions $g, h \in S(V_n)$ such that $h = (x, y)g$ we

have

$$p_{\alpha,\beta}^h - p_{\alpha,\beta}^g = \begin{cases} 0, & \text{if } \alpha = x \oplus y, \\ (I_\beta(x_1) + I_\beta(x_2) - I_\beta(x_3) - I_\beta(x_4)) \cdot 2^{1-n} & \text{otherwise,} \end{cases}$$

where $x_1 = g(x \oplus \alpha) \oplus g(y)$, $x_2 = g(y \oplus \alpha) \oplus g(x)$, $x_3 = g(x \oplus \alpha) \oplus g(x)$, $x_4 = g(y \oplus \alpha) \oplus g(y)$.

Proof. Let us consider the following transformations:

$$\begin{aligned} p_{\alpha,\beta}^h - p_{\alpha,\beta}^g &= (|\{z \mid h(z \oplus \alpha) \oplus h(z) = \beta\}| - |\{z \mid g(z \oplus \alpha) \oplus g(z) = \beta\}|) \cdot 2^{-n} \\ &= \left(\sum_{z \in V_n} I_\beta(h(z \oplus \alpha) \oplus h(z)) - \sum_{z \in V_n} I_\beta(g(z \oplus \alpha) \oplus g(z)) \right) \cdot 2^{-n} \\ &= 2^{1-n} \cdot \sum_{z \in \{x,y\}} (I_\beta(h(z \oplus \alpha) \oplus h(z)) - I_\beta(g(z \oplus \alpha) \oplus g(z))). \end{aligned}$$

It is easy to check that if $\alpha = x \oplus y$, then $p_{\alpha,\beta}^h - p_{\alpha,\beta}^g = 0$.

Otherwise, if $\alpha \neq x \oplus y$, then

$$\begin{aligned} & \left(I_\beta(g(x \oplus \alpha) \oplus g(y)) + I_\beta(g(y \oplus \alpha) \oplus g(x)) \right. \\ & \quad \left. - I_\beta(g(x \oplus \alpha) \oplus g(x)) - I_\beta(g(y \oplus \alpha) \oplus g(y)) \right) \cdot 2^{1-n} \\ & = (I_\beta(x_1) + I_\beta(x_2) - I_\beta(x_3) - I_\beta(x_4)) \cdot 2^{1-n}. \end{aligned}$$

This concludes the proof. □

Let us denote by t_2 the complexity of an algorithm 2.

Proposition 4. *If $n \rightarrow \infty$, then*

$$t_2 = O(2^n).$$

Proof. The complexity of algorithm is the product of the number 2^n of iterations of step 1 and constant number of operations on steps of the algorithm. □

Remark 2. The algorithm 2 is approximately 2^n times faster than the classical algorithm of computing the differential uniformity.

Example 2. Consider the substitutions $g, h \in S(V_3)$ such that $h = (0, 1)g$:

$$g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 6 & 1 & 7 & 2 & 4 & 3 & 0 \end{pmatrix}, \quad h = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 5 & 1 & 7 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

The results obtained by algorithm 2 are presented in Table 2 (all changing elements of the matrix are in bold).

In particular, taking into account proposition 3 we can conclude that:

- 1) Since $\alpha = (0 \oplus 1) = 1$, then for any $\beta \in V_3$ we have

$$p_{1,\beta}^h - p_{1,\beta}^g = 0,$$

$$2) p_{4,4}^h - p_{4,4}^g = (I_4(4) + I_4(1) - I_4(7) - I_4(2)) \cdot 2^{-2} = 1/4,$$

$$3) p_{7,5}^h - p_{7,5}^g = (I_5(6) + I_5(6) - I_5(5) - I_5(5)) \cdot 2^{-2} = -2/4.$$

Table 2

Substitution g								Substitution h							
5 6 1 7 2 4 3 0								6 5 1 7 2 4 3 0							
DDT of g								DDT of h							
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0	0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0
0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
0	0	$\frac{2}{4}$	0	0	0	0	$\frac{2}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
0	0	$\frac{2}{4}$	0	0	0	0	$\frac{2}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$
0	0	0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0	0	0	$\frac{2}{4}$	0	$\frac{2}{4}$	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	$\frac{2}{4}$	$\frac{2}{4}$	0
p	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1			p	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1		
$ D(g, p) $	36	0	12	0	1			$ D(g, p) $	25	16	6	0	0		

4. Cryptographic applications

The results of this paper may be applied to optimize some heuristic methods of constructing s-boxes. It is well-known that many heuristic methods are based on swap operations (for example, see [1–6, 9]). We can optimize some of this methods using algorithms of this paper. Let's show this for the spectral-linear and spectral-differential methods of generating s-boxes (see [4, 5]).

Let t_{sl} be the computational complexity of the algorithm 2 described in [4]. To optimize this algorithm we need to use the results of this paper at the step 2 as follows (see the Figure 1 below). First we calculate the LAT and the DDT for each item of the list I using classical algorithms with complexities $O(2^{2n}n)$ and $O(2^{2n})$ respectively. Then we multiply each element of list I on transpositions: $g'_{i,j} = (x, y)g_i$. Next we calculate the parameters $\delta_{g'_{i,j}}$, $|L(g'_{i,j}, \delta_{g'_{i,j}})|$, $p_{g'_{i,j}}$ for each element $g'_{i,j}$, $i = \overline{1, |I|}$, $j = \overline{1, C_{2^n}^2}$, from the intermediate list using algorithms 1 and 2 of this paper with complexity $O(2^{2n})$. Finally, we insert the element $g'_{i,j}$ into the list I' if the parameters of $g'_{i,j}$ are better than g_i .

(g_i) denotes g_i and its LAT and DDT as the result of classical algorithms with complexities $O(2^{2n}n)$ and $O(2^{2n})$ respectively;

$[g'_{i,j}]$ denotes $g'_{i,j}$ and its parameters $\delta_{g'_{i,j}}$, $|L(g'_{i,j}, \delta_{g'_{i,j}})|$, $p_{g'_{i,j}}$ as the result of algorithms 1 and 2 of this paper with complexity $O(2^{2n})$

Figure 1: structure of the modified step 2 of the algorithm 2 taken from [4]

As a result we have the following proposition.

Proposition 5. *If $n \rightarrow \infty$, then*

$$t_{sl} = O(2^{7n}).$$

Proof. In the paper [4] it was proved that $t_{sl} = O(n \cdot 2^{7n})$. The complexity of algorithm is the product of the complexity $c_1 \cdot 2^{4n} \cdot n$, $c_1 = \text{const}$, of step 2 and the maximum number $c_2 \cdot 2^{3n}$, $c_2 = \text{const}$, of iterations of the step 2.

Using algorithms 1 and 2 of our paper we obtain the following complexity bounds of step 2: $c_1 \cdot 2^{4n}$. This completes the proof of the proposition. $\square$

Let t_{sd} be the computational complexity of algorithm 1 described in [4]. Reasoning similarly as for the above case, we easily derive the following proposition.

Proposition 6. *If $n \rightarrow \infty$, then*

$$t_{sd} = O(n \cdot 2^{6n}).$$

Remark 3. Modified algorithms use the same amount computer memory.

Suppose t_{new} is the average execution time of the modified algorithm, t_{old} is the average execution time of its original version. For $n = 5, \dots, 8$ the table 3 contains the values $\frac{t_{old}}{t_{new}}$ for spectral-linear and spectral-differential methods. In particular, table 3 shows that:

- 1) if $n = 7$ then modified algorithm is approximately 4 times faster than its original version (see Algorithm 2 in [4]);
- 2) if $n = 8$ then modified algorithm is approximately 28 times faster than the old one (see Algorithm 1 in [4]).

Table 3

n	5	6	7	8
Spectral-linear method (see Algorithm 2 in [4])	2	3	4	5
Spectral-differential method (see Algorithm 1 in [4])	5	8	16	28

Modified spectral-linear and spectral-differential methods have been applied to a large number of random permutations $g \in S(V_6)$. As a result we have a lot of new affine nonequivalent permutations $g' \in S(V_6)$ having almost optimal cryptographic parameters (see table 4 in the appendix).

5. Conclusions

The results of our paper may be summarized as follows.

The δ_h -parameter (the linear spectrum $L(h)$, the LAT $T_1(h)$) of an s-box $h \in S(V_n)$ such that $h = (x, y)g$ may be computed by means of the algorithm 1 described in this paper with the time complexity $O(2^{2n})$.

The p_h -parameter (the differential spectrum $D(h)$, the DDT $T_2(h)$) of an s-box $h \in S(V_n)$ such that $h = (x, y)g$ may be computed by means of the algorithm 1 described in this paper with the time complexity $O(2^n)$.

We have optimized some heuristic methods of generating s-boxes. The optimized methods may be used for generating s-boxes $g \in S(V_n)$, $n \leq 16$.

References

- [1] Isa H., Jamil N., Z'aba M., "Hybrid heuristic methods in constructing cryptographically strong S-boxes", *Int. J. Cryptology Research*, **6**:1 (2016), 1–15.
- [2] Ivanov G., Nikolov N., Nikova S., "Cryptographically strong S-boxes generated by modified immune algorithm", *Lect. Notes Comput. Sci.*, **9540**, 2015, 31–42.
- [3] Ivanov G., Nikolov N., Nikova S., "Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties", *Cryptogr. and Communic.*, **8**:2 (2016), 247–276.
- [4] Menyachihin A.V., "Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters", *Matematicheskie Voprosy Kriptografii*, **8**:2 (2017), 97–116.
- [5] Menyachihin A.V., "The method for generating s-boxes by using elements of linear and differential spectra and device implementing it", Patent RU 2633132, 29, 2017.
- [6] Kazymyrov O.V., Kazymyrova V.N., Oliynykov R.V., "A method for generation of high-nonlinear S-boxes based on gradient descent", *Mathematicheskie Voprosy Kriptografii*, **5**:2 (2014), 71–78.
- [7] Millan W., Clark A., Dawson E., "Boolean functions design using hill climbing methods", *Lect. Notes Comput. Sci.*, **1587**, 1999, 1–11.
- [8] Millan W., "How to improve the nonlinearity of bijective S-boxes", Australian Conf. Inf. Security and Privacy, *Lect. Notes Comput. Sci.*, **1438**, 1998, 181–192.
- [9] Tesar P., "A new method for generating high non-linearity S-boxes", *Radioengineering*, **19**:1 (2010), 23–26.
- [10] Yu Y., Wang M., Li Y., "Constructing differentially 4-uniform permutation from known ones", *Chinese J. Electronics*, **22**:2 (2013), 495–499.

6. Appendix

Table 4

$g' \in S(V_6)$		$p_{g'}$	$ D(g', p_{g'}) $	$\delta_{g'}$	$ L(g', \delta_{g'}) $
8 18 f 9 22 6 37 2e 1f 2f 1 2c c 32 2d 7 e 30 3c 3 0 b 2b 38 17 3f a 1d 2a 16 11 1c	25 27 3a 34 13 35 36 d 4 5 29 21 1a 31 14 3d 12 15 28 1b 26 3b 19 2 3e 20 1c 23 24 10 39 33	$\frac{4}{64}$	240	$\frac{10}{32}$	44
36 3a 24 3d 2d c 34 31 2f 30 10 21 3e 2e 13 1 19 5 2c 32 35 11 38 25 2 17 15 f 39 3f 0 20	26 1d 33 7 22 28 4 12 14 16 27 23 3c e a 1b 6 1f 8 1a b 3 3b 9 1c 18 1e 29 2b 2a 37 d	$\frac{4}{64}$	264	$\frac{10}{32}$	58
1b 16 37 1c 8 1c f 34 12 0 23 18 2d e 1a 2c 35 17 5 21 1f 38 3e 3a 19 2f 25 7 2b 3b b	a 3f 11 14 39 29 3d 3 30 2 22 26 2a 1d 13 20 10 28 32 27 15 33 9 c 6 1 3c d 4 24 36 2e	$\frac{4}{64}$	264	$\frac{10}{32}$	58
34 1f 2f 2f 35 14 32 2e 11 8 2d 29 22 1b 1 15 6 1e 30 5 9 10 3a 25 d 3c c 1c 3d e 12 39	33 20 3f a 1a 7 17 2c b 26 2 0 2b f 28 4 24 3e 36 31 19 38 3 16 2a 1d 3b 18 27 13 23 37	$\frac{4}{64}$	282	$\frac{10}{32}$	55
36 15 23 34 1f a 35 3c 8 9 18 19 2e 39 5 1c 10 38 25 e 1 3f 33 4 3a 24 2 b 1b 31 3b 2f	30 16 d 22 7 3d 0 2c f 20 1e 28 6 1d 12 11 32 13 2d 3e 29 1a 37 2b 27 17 3 26 c 14 21 2a	$\frac{4}{64}$	285	$\frac{10}{32}$	63
14 29 12 22 c 1c 2b 3f 3d 27 1f 33 36 10 34 2f 3a 25 32 1a d 35 3e 23 2c 39 18 13 e 21 b 26	f 19 2e 2a 6 3b 20 1b 15 0 17 38 5 11 1d 3c 28 3 24 4 1 7 8 2 9 16 a 30 37 1e 2d 31	$\frac{4}{64}$	303	$\frac{12}{32}$	1
2f 11 39 c 2c 6 1e 15 e 20 1a 2d 37 32 17 18 35 19 12 3f 27 2b 2e a 1d 2a 3c 31 36 10 3 1	1c 14 38 23 4 b 33 f 1f 3e 8 22 7 21 25 d 3d 34 3b 26 29 30 16 28 13 2 0 3a 9 24 5 1b	$\frac{4}{64}$	312	$\frac{12}{32}$	6
25 1f 8 11 1c 20 0 15 9 23 18 2a 12 1d 7 38 2f 2 35 21 b 34 36 1a 6 31 d c 1e 26 14 39	17 3c 4 2e 30 e 37 f 32 3b 3d 22 13 16 33 27 1 28 19 2b 3 2c 5 24 3f 10 2d 29 3a 1b a 3e	$\frac{4}{64}$	297	$\frac{12}{32}$	7
19 7 2d 29 1b 38 27 3f 11 31 0 2a 10 3b 22 23 1d 8 20 3 3a 13 2f 1e 26 12 28 1c 2b 3e 32	21 e 33 36 1 24 2e 25 4 30 35 2c 14 3c 16 6 18 c 1a 39 f a b 9 15 d 1f 5 34 3d 2 17	$\frac{4}{64}$	309	$\frac{12}{32}$	6
4 27 8 22 19 2f 3c 1d 7 3a c 1e 28 3f 30 2d 6 18 5 32 3b 10 36 1a 13 a 16 1b 9 24 1 2b	3 b 37 38 15 21 1c d 1f 34 3e 23 20 39 26 25 f 3d 31 17 2 2a 29 33 0 11 35 e 12 14 2e 2c	$\frac{4}{64}$	291	$\frac{12}{32}$	3
d 33 3d 22 2d 1d 3b 7 5 26 15 2 16 25 4 1f 29 21 1a 13 9 6 38 32 e 27 2c 2f c 14 39 18	19 24 1c 30 20 1b 1 35 36 10 11 1e a 2e 3a 3f 2b 3 37 34 23 17 28 f 31 12 2a 0 3c b 3e 8	$\frac{4}{64}$	309	$\frac{14}{32}$	1
10 0 18 3f 3d 26 6 1e 11 17 25 33 3a 32 d 1a 12 16 2f e b 3 8 a 23 2e 31 2d 2c 34 21 35	2 28 3e 37 1 7 1c 2b 20 5 29 3b 27 c 36 1f 19 1d 4 2a 39 14 13 30 24 38 9 1b 22 f 3c 15	$\frac{4}{64}$	315	$\frac{14}{32}$	1
30 1a 3c 16 f d 29 6 2c e 17 21 5 a 11 36 3f 19 3d 7 3b 18 31 2a 3a 2b b 15 37 0 28 22	32 2f 2d 3e 33 34 2 1 1d 39 1c 2e 14 9 10 1b 1f 26 1e 20 3 38 35 4 23 24 27 13 c 12 25 8	$\frac{4}{64}$	315	$\frac{14}{32}$	1
16 25 13 a 28 8 3b 4 29 21 37 39 12 1c 3a 17 2f 31 c e 6 0 9 23 2d 24 3c 2a 3d 2c 18	1b 27 19 20 7 1e 11 b 34 d 30 14 1 3f 1a 22 26 5 32 35 2 1d 33 15 3e f 10 2b 1f 3 36 2e	$\frac{4}{64}$	288	$\frac{14}{32}$	1
32 35 28 3c 5 27 7 4 2c 14 36 6 25 1c 1f 3 3b 38 f 26 a 24 2e 23 1 34 e 2d 10 33 20 30	21 1d d 1b 8 2 13 16 22 15 b 2a c 3e 19 0 18 9 3a 1a 1c 39 11 3d 31 2b 17 37 12 29 2f	$\frac{4}{64}$	297	$\frac{14}{32}$	1