# The change in linear and differential characteristics of substitution multiplied by transposition

A.V. Menyachikhin
and88@list.ru

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

1. genetic algorithms,
2. hill climbing methods,
3. methods of gradient descent,
4. spectral-linear and spectral-differential methods.

The main problem of using heuristic methods is the high level of their time complexity. The $\delta_g$-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 $x, y \in V_n$, $g \in S(V_n)$.

**Definition**

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

$$\delta_g = \max_{\alpha,\beta \in V_n^\times} \left|\delta^g_{\alpha,\beta}\right|$$

where $\delta^g_{\alpha,\beta} = 2^{1-n} \cdot \left|\{x \in V_n \,|x \circ \alpha = g(x) \circ \beta\}\right| - 1.$

S-boxes with small value of $\delta_g$-parameter offer better resistance against linear attacks.

**Definition**

The *linear approximation table* (LAT) of s-box $g$ is a $2^n \times 2^n$ matrix $T_1$ such that $T_1(\alpha,\beta) = \delta^g_{\alpha,\beta}$.

For $g \in S(V_n)$ and for element $\delta \in \left\{ \frac{i}{2^{n-2}} \,\middle|\, i = 0, 1, \ldots, 2^{n-2} \right\}$ we define the set

$$L(g,\delta) = \left\{ (\alpha,\beta) \in V_n^\times \times V_n^\times \,\middle|\, \left|\delta^g_{\alpha,\beta}\right| = \delta \right\}.$$

**Definition**

The *linear spectrum* of s-box $g$ is defined as

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

**Definition**

The *differential uniformity* of s-box $g$ is defined as

$$p_g = \max_{\alpha,\beta \in V_n^{\times}} p^g_{\alpha,\beta},$$

$$\text{where } p^g_{\alpha,\beta} = 2^{-n} \cdot \left|\{x \in V_n \,|g(x \oplus \alpha) \oplus g(x) = \beta\}\right|.$$

S-boxes using in cryptographic primitives must have a low $p_g$-parameter value to provide high resistance to differential cryptanalysis.

**Definition**

The *difference distribution table* (DDT) of s-box $g$ is a $2^n \times 2^n$ matrix $T_2$ such that $T_2(\alpha, \beta) = p^g_{\alpha,\beta}$.

For $g \in S(V_n)$ and for element $p \in \left\{ \frac{i}{2^{n-1}} \,\middle|\, i = 0, 1, ..., 2^{n-1} \right\}$ we define the set

$$D(g, p) = \left\{(\alpha, \beta) \in V_n^{\times} \times V_n^{\times} \,\middle|\, p^g_{\alpha,\beta} = p \right\}.$$

**Definition**

The *differential spectrum* of s-box $g$ is defined as

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

The next section deals with the change in linear and differential characteristics of substitution multiplied by transposition. This issue has been studied in [1]. The authors showed that for $h = (x, y)\, g$ such that $g, h : V_n \to V_n$ we get:

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

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

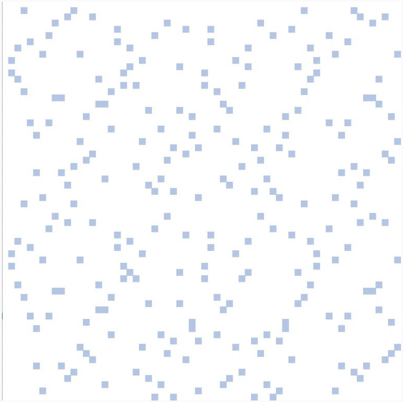
In [2] the similar properties of boolean functions are used to optimize the hill climbing methods.


1. Yu Y., Wang M., Li Y. Constructing differentially 4 uniform permutation from know ones. Chinese Journal of Electronics, 2013, 22:2, 495–499
2. Millan W., Clark A. and Dawson E. Boolean functions design using hill climbing methods. Lecture Notes in Computer Science, 1999, 1587, 1–11.

For example, let us consider two s-boxes $g, h \in S(V_6)$ such that $h = (x, y)\, g$. The following tables illustrate the change in linear approximation table and difference distribution table of s-box $h$.

**The change in LAT of substitution multiplied by transposition**

**The change in DDT of substitution multiplied by transposition**



## Algorithm 1

Input. Substitution $g \in S(V_n)$; the elements $x, y \in V_n$; the LAT $T_1(g)$; the linear spectrum $D(g)$.

Step 1. For each element $i = 0, ..., n-1$ do the following items:

- calculate elements $\alpha = x \oplus y$ and $\beta = g(x) \oplus g(y)$;
- if $\alpha \circ i > 0$ then add $i$ to the list $I_1$;
- if $\beta \circ i > 0$ 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 items:

- calculate $\left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| = \left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| - 1$;
- calculate value $\delta^g_{\alpha,\beta} = \delta^g_{\alpha,\beta} + (-1)^{\alpha\circ x \oplus \beta \circ g(x) \oplus 1} \cdot 2^{2-n}$;
- calculate $\left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| = \left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| + 1$.

Step 3. 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 linear spectrum $L(h)$.

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

**Proposition**

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

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

Let us denote by $t_1$ the complexity of an algorithm 1.

**Proposition**

*As $n \to \infty$ we obviously have*

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

**Remark**

*The algorithm 1 is nearly $n$ times faster than the classical algorithm of calculating the linearity.*

## Algorithm 2

Input. $g \in S(V_n)$; $x, y \in V_n$; $T_2(g)$; $D(g)$.

Step 1. For each element $\alpha = 1, ..., 2^n - 1$ such that $\alpha \neq x \oplus y$ do the following items:
calculate elements
$\beta_0 = g(x) \oplus g(x \oplus \alpha)$ and $\beta_2 = g(y) \oplus g(y \oplus \alpha)$;
Let us consider 2 cases.
Case 1: assume that $\beta_0 = \beta_2$; then

- calculate element $\beta_1 = g(y) \oplus g(x \oplus \alpha)$;
- for each element $i = 0, 1$ calculate values:

$$\left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| - 1,$$

$$\left|D\left(g, p^g_{\alpha,\beta_i} + 4 \cdot (-1)^{i+1}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i} + 4 \cdot (-1)^{i+1}\right)\right| + 1.$$

Case 2: suppose that $\beta_0 \neq \beta_2$; then

- calculate elements
$\beta_1 = g(y) \oplus g(x \oplus \alpha)$ and $\beta_3 = g(x) \oplus g(y \oplus \alpha)$;
- for each element $i = 0, ..., 3$ calculate values:

$$\left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| - 1,$$

$$\left|D\left(g, p^g_{\alpha,\beta_i} + 2 \cdot (-1)^{i+1}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i} + 2 \cdot (-1)^{i+1}\right)\right| + 1.$$

Step 2. The algorithm stops after calculating $h = (x, y)\, g$, $D(h) = D(g)$.

Output. $h \in S(V_n)$; $D(h)$.

Let us denote the indicator function

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

The correctness of the algorithm 2 is shown in the following proposition.

**Proposition**

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

$$p^h_{\alpha,\beta} - p^g_{\alpha,\beta} = \begin{cases} 0, \textit{if } \alpha = x \oplus y \\ \left(I_\beta(\beta_1) + I_\beta(\beta_3) - I_\beta(\beta_0) - I_\beta(\beta_2)\right) \cdot 2^{1-n}, \textit{otherwise} \end{cases},$$

*where* $\beta_1 = g(x \oplus \alpha) \oplus g(y)$, $\beta_3 = g(y \oplus \alpha) \oplus g(x)$, $\beta_0 = g(x \oplus \alpha) \oplus g(x)$, $\beta_2 = g(y \oplus \alpha) \oplus g(y)$.

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

**Proposition**

*As $n \to \infty$ we obviously have $t_2 = O(2^n)$.*

**Remark**

*The algorithm 2 is nearly $2^n$ times faster than the classical algorithm of computing the differential uniformity.*

The results of this paper can be applied to optimize some heuristic methods of constructing s-boxes. It is well-known that the most heuristic methods are based on swap operations. 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.

Spectral-linear and spectral-differential methods were first introduced in 2016 [1]. These methods are based on using linear $L(g_i)$ and differential $D(g_i)$ spectra to improve iteratively given S-box with respect to all properties.

Methods and device for their realization were patented in Russian Federation in 2017 [2].


1. Menyachihin A.V. Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters. Mathematical aspects of cryptography, 2017, 8:2, 97–116.
2. Menyachikhin A.V. Method for generating S-boxes using the values of linear and differential spectra and device for its realization. RU Patent №2633132, 2017, Bull. №29.

Let $t_{sl}$ be the computational complexity of algorithm 2 described in [1].

**Proposition**

*As $n \to \infty$ we have the following $t_{sl} = O\left(2^{7n}\right)$.*

Let $t_{sd}$ be the computational complexity of algorithm 1 described in [1].

**Proposition**

*As $n \to \infty$ we obviously have $t_{sd} = O\left(n \cdot 2^{6n}\right)$.*

Suppose $t_{new}$ is the average execution time of the modified algorithm, $t_{old}$ is the average execution time of its original version. For $n$ $(n = 5, ..., 8)$ table 3 includes the value $\frac{t_{old}}{t_{new}}$ of spectral-linear and spectral-differential methods. In particular from table 1 we obtain the following:

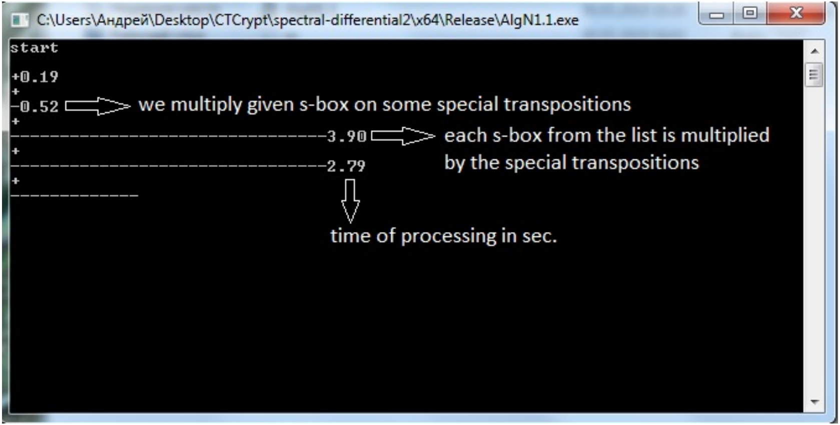
1. if $n = 7$ then modified algorithm is nearly 4 times faster than its original version (see Algorithm 2 in [1]);
2. if $n = 8$ then modified algorithm is nearly 28 times faster than the old one (see Algorithm 1 in [2]).

Table 3.

| $n$ | 5 | 6 | 7 | 8 |
|---|---|---|---|---|
| Spectral-linear method | 2 | 3 | 4 | 5 |
| Spectral-differential method | 5 | 8 | 16 | 28 |

# Cryptographic applications

Here, we demonstrate the operation of the device based on modified spectral-differential method with size of list $|I| = 32$ for some given s-box $g_0 \in S(V_8)$.



# Thanks for your attention

Feel free to contact us at and88@list.ru