An algorithm for bounding non-minimum weight differentials in 2-round LSX-ciphers

Vitaly Kiryukhin

JSC «InfoTeCS» LLC «SFB Lab», Moscow, Russia
Vitaly.Kiryukhin@infotecs.ru

Abstract

This article describes some approaches to bounding non-minimum weight differentials (EDP) and linear hulls (ELP) in 2-round LSX-cipher. We propose a dynamic programming algorithm to solve this problem. For 2-round Kuznyechik the nontrivial upper bounds on all differentials (linear hulls) with 18 and 19 active Sboxes was obtained. These estimates are also holds for other differentials (linear hulls) with a larger number of active Sboxes. We obtain a similar result for 2-round Khazad. As a consequence, the exact value of the maximum expected differential (linear) probability (MEDP/MELP) was computed for this cipher.

Keywords: Kuznyechik, Khazad, SPN, LSX, differential cryptanalysis, linear cryptanalysis, MEDP, MELP

1 Introduction

Differential [2] and linear [3] cryptanalysis are the two most known statistical attacks applicable to block ciphers. In this paper we will focus on the first method. The analogous results for linear cryptanalysis will be obtained in a similar way, due to the existing well-known duality [4].

There are several approaches to estimating the security of ciphers against differential attacks. Many papers are devoted to the differential characteristics. The maximal probability of such characteristics (EDCP) decreases when the number of active Sboxes within $R$ rounds increases. The upper bound on such probability can be analytically obtained for many LSX-ciphers (AES [11], Khazad [12], Kuznyechik [1], etc.). In particular, these results are presented in [11, 17].

However, many researchers note that differential cryptanalysis exploits differentials and not characteristics (see for example [16, 14, 5]). The probability (EDP) of a differential $(\Delta x, \Delta y)$ corresponds to the sum of the probabilities of all characteristics with input difference $\Delta x$ and output difference
\( \Delta y \) \[8\]. So from this point of view security of a cipher against differential attacks is based on the maximum expected differential probability (MEDP) over \( R \geq 2 \) rounds.

**Related works.** For 2-round LSX-ciphers, some approaches to computing upper bounds on the MEDP are known \[13, 14, 15\].

An algorithm for computing the exact MEDP of 2-round AES was proposed in \[5\]. Article \[10\] describes upper bounds on the MEDP for so-called «nested» LSX-ciphers (e.g. 4-round AES).

In \[16\] was shown that for some 2-round LSX-ciphers the MEDP is achieved by differentials involving a number of active Sboxes which exceeds the branch number of the linear layer (non-minimum weight differentials).

Some results about differential properties of 2-round Kuznyechik was obtained in \[18\]. The cited paper contains an algorithm for constructing the best minimum weight differentials and a proof that all other differentials have a lower EDP. Thanks to these two results, the exact value of the 2-round MEDP was computed.

**Our contribution.** We propose a dynamic programming algorithm designed for bounding non-minimum-weight differentials in 2-round LSX-ciphers. It uses only the difference distribution table and the differential branch number of the linear layer. The algorithm minimizes the number of high probability differential trails and does not try to minimize the total number of trails. Because of this reason, the algorithm is not effective for ciphers with small block size (for example, 32-bit 2-round AES).

We applied the developed algorithm to the 2-round Kuznyechik (Section 4 and Appendix B): the probability of any 2-round differential (linear hull) with \( n + 3 = 19 \) active Sboxes is bounded by \( 2^{-88.34} \) (\( 2^{-79.63} \) correspondingly). These bounds also holds for any differential (linear hull) with \( a \geq n+3 \) active Sboxes. Similar results were obtained for 2-round Khazad (Appendix C), and as a result, the exact values of \( \text{MEDP} = 2^{-45} + 2^{-60} \) and \( \text{MELP} = 2^{-37.80} \) are also proved.

The set of estimates obtained by us can be used in further researches to calculate the bounds on the MEDP (MELP) for more rounds. We plan to use our new results together with a modified KMT2-DC (KMT2) algorithm \[6, 7\]. The approach \[7\] allows to incorporate other upper bounds when those bounds are superior to the values determined directly by the original algorithm \[6\]. In this way, we hope to prove the greater security of Kuznyechik to differential and linear cryptanalysis.
2 Notations and definitions

An LSX cipher $E$ consists of sequence of rounds. Each of them contains three operations: $X$ – modulo 2 addition of an input block with an iterative key, $S$ – parallel application of a fixed bijective substitution $s$, $L$ – linear transformation which may be represented as multiplication by the binary matrix.

To simplify the text and notation we consider only byte-oriented LSX-ciphers.

Let us denote:
$n$ – block size in bytes,
$\oplus$ – bitwise XOR operation,
$v[i]$ – $i$-th element of vector or sequence $v$, $1 \leq i \leq l$, where $l$ is the number of elements of $v$,
$\text{Supp}(v) = \{i : v[i] \neq 0\}$ – the support of a vector $v$,
$\text{wt}(v) = \# \{i : v[i] \neq 0\}$ – the weight of a vector $v$,
$F_q$ – finite field of $q$ elements,
$F_q^*$ – set of all nonzero elements of the field $F_q$,
$F_q^l$ – set of $l$-element vector over $F_q$.

Depending on the context, we will interpret a value $z \in \mathbb{Z}_{2^l-1}$ as element of $F_{2^l}$ or $F_q^l$ or as an integer.

**Definition 1.** Let $s : F_2^8 \rightarrow F_2^8$, let $a, b \in F_2^8$ be fixed, and let $x$ be a random variable having uniform distribution on $F_2^8$. The differential probability of $(a, b)$ is defined as

$$\text{DP}(a, b) = \Pr_x (s(x) \oplus s(x \oplus a) = b).$$

**Definition 2.** Let $E$ be a cipher with key-size $\kappa$ and block-size $l$. Let $x$ be a random variable having uniform distribution on $F_2^l$. Then the expected (over keys $K$) differential probability of $(\Delta x, \Delta y)$ is defined as

$$\text{EDP}(\Delta x, \Delta y) = 2^{-\kappa} \sum_{K \in F_2^\kappa} \Pr_x (E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y),$$

where $E_K$ is a cipher with key $K$.

**Definition 3.** The maximum expected differential probability is

$$\text{MEDP} = \max_{\Delta x \neq 0, \Delta y} \text{EDP}(\Delta x, \Delta y).$$
Definition 4. Let \( s \) be a function \( F_2^8 \to F_2^8 \). The differential distribution table (DDT) is a \( 2^8 \times 2^8 \) matrix of transition probabilities such that
\[
DDT[a][b] = \frac{\# \{ x \in F_2^8, \ s(x) \oplus s(x \oplus a) = b \} }{2^8} = DP(a, b), \ a, b \in F_2^8,
\]
and \( p_{\text{max}} = \max_{a \neq b} DDT[a][b] \).

Definition 5. Let \( \mathcal{L} \)-transformation (from \( F_2^n \) to \( F_2^n \)) be \( F_2^8 \)-linear. We associate with \( \mathcal{L} \) the code \( \mathcal{C}_\mathcal{L} \) of length \( 2n \) over \( F_2^8 \) defined by
\[
\mathcal{C}_\mathcal{L} = \{ (c, \mathcal{L}(c)) \mid c \in F_2^n \}.
\]
The differential branch number \( B_\mathcal{L} \) of the linear transformation \( \mathcal{L} \) is the minimum distance of the code \( \mathcal{C}_\mathcal{L} \)
\[
B_\mathcal{L} = \min_{c \neq 0} \text{wt} (c, \mathcal{L}(c)).
\]
Further, to simplify the text, we assume that \( \mathcal{C}_\mathcal{L} \) is an MDS code and \( B = B_\mathcal{L} = n + 1 \).

2-round LSX-cipher may be represented as a sequence of operations
\[
y = K_3 \oplus S(K_2 \oplus LS(K_1 \oplus x)),
\]
where \( x, y \in F_2^n \) are the plaintext and the ciphertext, \( K_1, K_2, K_3 \in F_2^n \) are round keys derived from the masterkey \( K \). The linear transformation on the last round was omitted without loss of generality.

A differential trail \( \Omega = (\Delta x, \Delta_1, \Delta_2, \Delta y) \) in 2-round LSX is a collection of four differences, where \( \Delta x = x \oplus x' \), \( \Delta_1 \) is the difference after the first nonlinear transformation, \( \Delta_2 = \mathcal{L}(\Delta_1) \), \( \Delta y = y \oplus y' \), \( x \) and \( x' \) are plaintext blocks, \( y \) and \( y' \) are the corresponding ciphertext blocks.

Definition 6 ([16]). The expected 2-round trail \( \Omega \) probability is defined as
\[
\text{EDCP}(\Omega) = 2^{-\kappa} \sum_{K \in F_2^n} \text{Pr}_x \left( \Delta_1 = x_1 \oplus x'_1 \text{ and } \Delta_2 = x_2 \oplus x'_2 \text{ and } \Delta y = y \oplus y' \right),
\]
where \( x \) is a random variable with the uniform distribution, \( x' = \Delta x \oplus x \), \( x_1, x'_1 \) are states after the first \( S \)-transformation, \( x_2, x'_2 \) are states before the second \( S \)-transformation, \( \kappa \) is a size of the masterkey \( K \).

We further assume that all round keys are independent and uniformly distributed (so-called Markov assumption [8]). Under this assumption we have
\[
\text{EDCP}(\Delta x, \Delta_1, \Delta_2, \Delta y) = \left( \prod_{j=1}^{n} \text{DP}(\Delta x[j], \Delta_1[j]) \right) \left( \prod_{j=1}^{n} \text{DP}(\Delta_2[j], \Delta y[j]) \right).
\]
Note that if \( \text{EDCP} (\Delta x, \Delta_1, \Delta_2, \Delta y) \neq 0 \), then \( \text{Supp} (\Delta x) = \text{Supp} (\Delta_1) \), \( \text{Supp} (\Delta_2) = \text{Supp} (\Delta y) \) and \( (\Delta_1, \Delta_2) \) is a codeword of the code \( C_L \). Therefore

\[
\text{EDP} (\Delta x, \Delta y) = \sum_{(\Delta x, \Delta y) \in C_L, \\text{Supp}(\Delta x) = \text{Supp}(\Delta_1), \\text{Supp}(\Delta_2) = \text{Supp}(\Delta y)} \prod_{j \in \text{Supp}(\Delta x)} \text{DP} (\Delta x[j], \Delta_1[j]) \prod_{j \in \text{Supp}(\Delta y)} \text{DP} (\Delta_2[j], \Delta y[j]).
\]

The equality between the above formula for \( \text{EDP} (\Delta x, \Delta y) \) and the definition [2] was proved in [8].

We define the weight (number of nonzero bytes) of the differential \( (\Delta x, \Delta y) \) or the differential trail \( (\Delta x, \Delta_1, \Delta_2, \Delta y) \) as \( \text{wt}(\Delta x) + \text{wt}(\Delta y) \). Denote

\[
\text{MEDP}_w = \max_{\Delta x \neq 0, \Delta y, \text{wt}(\Delta x) + \text{wt}(\Delta y) = w} \text{EDP} (\Delta x, \Delta y),
\]

\[
\text{MEDP}_w^+ = \max_{\Delta x \neq 0, \Delta y, \text{wt}(\Delta x) + \text{wt}(\Delta y) \geq w} \text{EDP} (\Delta x, \Delta y), \ B \leq w \leq 2 \cdot n.
\]

Note that all mentioned definitions \( \text{EDP}, \text{EDCP}, \text{MEDP} \) are related to 2-round case unless otherwise stated.

Our main goal is to compute the nontrivial upper bound on \( \text{MEDP}_B^+, \text{MEDP}_{B+1}^+, \text{MEDP}_{B+2}^+ \) etc.

3 Upper bound on non-minimum weight differentials

The strategy of our approach is as follows. Each differential trail \( \Omega = (\Delta x, \Delta_1, \Delta_2, \Delta y) \) in 2-round differential \( (\Delta x, \Delta y) \) uniquely corresponds to codeword \( (\Delta_1, \Delta_2) \) in \( C_L \). All possible trails (codewords) in the differential have the form \( \text{Supp}(\Delta x) = \text{Supp}(\Delta_1), \text{Supp}(\Delta_2) = \text{Supp}(\Delta y) \). Derive constraints («maximum cost») for the entire set of such codewords. Divide the set into several subsets. Compute contribution to the constraints («cost») and the corresponding upper bound («value») for each possible subset. Select subsets so that the upper bound («total value») is maximum and the selection satisfies all constraints («total cost» does not exceed «maximum cost»). Thus, we obtain the upper bound on the differential.

3.1 Auxiliary lemmas

Lemma 1 (The rearrangement inequality [9]). Let \( l \in \mathbb{N} \), and suppose \( c_1, c_2, \ldots, c_l \) and \( d_1, d_2, \ldots, d_l \) are sequences of nonnegative values. Let
\(\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_l\) and \(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_l\) be the sequences obtained by sorting original sequences in nonincreasing order. Then

\[ \sum_{i=1}^{l} c_i d_i \leq \sum_{i=1}^{l} \tilde{c}_i \tilde{d}_i. \]

**Lemma 2.** Let \(l \in \mathbb{N}\), and suppose \(c_1, c_2, \ldots, c_l,\) and \(\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_l,\) and \(d_1, d_2, \ldots, d_l\) are sequences of nonnegative values. Each of them sorted in nonincreasing order. Suppose there exists \(l', 1 \leq l' \leq l,\) such that

1) \(\tilde{c}_i \geq c_i,\) for \(1 \leq i \leq l'\)
2) \(\tilde{c}_i \leq c_i,\) for \(l' + 1 \leq i \leq l\)
3) \(\sum_{i=1}^{l} c_i \leq \sum_{i=1}^{l'} \tilde{c}_i\)
Then \(\sum_{i=1}^{l} c_i d_i \leq \sum_{i=1}^{l} \tilde{c}_i d_i.\)

**Proof.** The proof of the lemma is given in particular in [6]. \(\square\)

If statements 1-3 holds for some sequences \(\tilde{c}\) and \(c,\) then we will say that \(\tilde{c}\) is greater than \(c\) under the conditions of Lemma 2. Let \(D\) be a \(h \times v\) matrix such that

\[ D[i][j] \in \{p_1, p_2, \ldots, p_t, p_{\text{max}}\}, 1 \leq i \leq h, 1 \leq j \leq v, t \in \mathbb{N}, 0 \leq p_1 < p_2 < \ldots < p_t < p_{\text{max}} \leq 1, p_k, p_{\text{max}} \in \mathbb{R}, 1 \leq k \leq t.\]

Denote

\[ \nu_k(D) = \# \{(i, j) : D[i][j] = p_k, 1 \leq i \leq h, 1 \leq j \leq v, 1 \leq k \leq t, \} \]

\[ \nu_{\text{max}}(D) = \# \{(i, j) : D[i][j] = p_{\text{max}}, 1 \leq i \leq h, 1 \leq j \leq v, \}. \tag{1} \]

Denote by \(\omega_l(D)\) the number of rows containing exactly \(l\) elements \(p_{\text{max}}\)

\[ \omega_l(D) = \# \{i : \# \{j : D[i][j] = p_{\text{max}}, 1 \leq j \leq v\} = l, 1 \leq i \leq h\}, \]

\[ \sum_{l=1}^{v} \omega_l(D) \cdot l = \nu_{\text{max}}(D), \quad l_{\text{max}}(D) = \max_{\omega_l(D) \neq 0} \left( l \right). \tag{2} \]

Let \(\tilde{D}\) be the reordered matrix \(D\) (see Fig. 1). The reordering procedure consists of three following steps:
1) sort each row of \(\tilde{D}\) in nonincreasing order;
2) sort each column of \(\tilde{D}\) in nonincreasing order;
3) reorder each unequal to \(p_{\text{max}}\) element:

\[ \forall i, j, i', j' : \tilde{D}[i][j] = p_{\text{max}}\) or \(\tilde{D}[i'][j'] = p_{\text{max}}\) or

\[ \left( \tilde{D}[i][j] \geq \tilde{D}[i'][j'], i' > i \text{ or } i' = i, j' > j \right), \]

\[ 1 \leq i, i' \leq h, 1 \leq j, j' \leq v. \]
Lemma 3. Let $D$ and $\tilde{D}$ be defined as above, then

\[
\nu_k(D) = \nu_k(\tilde{D}), \quad \nu_{\max}(D) = \nu_{\max}(\tilde{D}), \quad 1 \leq k \leq t,
\]

\[
\omega_l(D) = \omega_l(\tilde{D}), \quad l_{\max}(D) = l_{\max}(\tilde{D}), \quad \forall l \in \mathbb{N},
\]

\[
\sum_{i=1}^{h} \prod_{j=1}^{v} D[i][j] \leq \sum_{i=1}^{h} \prod_{j=1}^{v} \tilde{D}[i][j].
\]

Figure 1: Example of matrix $\tilde{D}$ after the reordering procedure.

Proof. Let $\tilde{D} = D$ before reordering. We show that at each of three step the value

\[
\sum_{i=1}^{h} \prod_{j=1}^{v} \tilde{D}[i][j]
\]

does not decrease. We also show that the final form of $\tilde{D}$ is given uniquely (up to permutation of identical elements).

The first step of the reordering procedure does not change the value (3) due to commutativity of multiplication.

By the rearrangement inequality it follows that the second step does not decrease (3).

Note that after these two steps

\[
\prod_{j=1}^{v} \tilde{D}[i][j] \geq \prod_{j=1}^{v} \tilde{D}[i+1][j], \quad \forall i = 1, h - 1.
\]

The set of $\omega_0, \ldots, \omega_{l_{\max}}$ is also the same as before. Therefore, the positions of all elements $p_{\max}$ are known (the gray area in figure 1).
Algorithm 1: Step 3 of the reordering procedure

For step 3, we will use a procedure similar to the well-known selection sort. Let’s write a row-by-row coordinates of all elements

\[(1,1), (1,2), \ldots, (1,v), (2,1), (2,2), \ldots, (2,v), \ldots, (h,1), (h,2), \ldots, (h,v).\] (5)

Remove from (5) all elements \((i,j)\) such that \(\~D[i][j] = p_{\text{max}}\). We obtain the sequence of indexes

\[\text{Indexes} = (i_1,j_1), (i_2,j_2), \ldots, (i_N,j_N), N \leq h \cdot v.\]

Reorder the table elements according to the pseudocode.

We have got the table as in figure 1. Let us show further that value (3) has never decreased in the reordering process.

Let’s consider line 5 of the pseudocode (Algorithm 1). If \(i_{\text{pos}} = i_{\text{pos}_{\text{max}}}\) then (3) remains the same due to commutativity of multiplication. If \(i_{\text{pos}} < i_{\text{pos}_{\text{max}}}\) then (3) is not decreased due to (4). But after the exchange of elements, inequality (4) may not be true any more.

Line 6 has a technical role and does not affect the final appearance of \(\~D\). This sort does not decrease (3) because of the rearrangement inequality. Inequality (4) becomes true after this sorting. Also note that sorting does not change the previously reordered elements.

The Lemma is proved. \(\Box\)

**Lemma 4.** Let \(D\) and \(\~D\) be given as in Lemma 3. Suppose \(c_1, c_2, \ldots, c_h\) is a sequence of nonnegative values. Let \(\~c_1, \~c_2, \ldots, \~c_h\) be obtained by sorting the above sequence in nonincreasing order. Then

\[\sum_{i=1}^{h} c_i \prod_{j=1}^{v} D[i][j] \leq \sum_{i=1}^{h} \~c_i \prod_{j=1}^{v} \~D[i][j].\]

**Proof.** Directly follows from Lemmas 1 and 3. \(\Box\)
3.2 Representation of trails in the differential

Consider an arbitrary differential \((\Delta x, \Delta y)\), \(\text{wt}(\Delta x) + \text{wt}(\Delta y) = \mathcal{B} + 1\). The differential consists only of trails \((\Delta x, \Delta_1, \Delta_2, \Delta y)\) such that \(\text{Supp}(\Delta x) = \text{Supp}(\Delta_1) = \{k_1, k_2, \ldots, k_t\}, \text{Supp}(\Delta y) = \text{Supp}(\Delta_2) = \{m_1, m_2, \ldots, m_r\}, t + r = \mathcal{B} + 1 = n + 2\).

It is easy to show that the number of differential trails does not exceed \(T \leq (2^8 - 1)^2\). Otherwise, there is a pair of codewords \((\Delta_1, \Delta_2)\) and \((\Delta'_1, \Delta'_2)\) such that \(\text{wt}((\Delta_1, \Delta_2) \oplus (\Delta'_1, \Delta'_2)) < \mathcal{B}\).

Let’s imagine a set of differential trails in the form of a table. Such a table, called Trails, has a size of \(T \times (n + 2)\). Each row is non-zero bytes of the corresponding codeword

\[
\text{Trails}[i] = \Delta_1[k_1], \ldots, \Delta_1[k_t], \Delta_2[m_1], \ldots, \Delta_2[m_r], 1 \leq i \leq T,
\]

\[
\text{EDP}(\Delta x, \Delta y) = \sum_{i=1}^{T} \prod_{j=1}^{t} \text{DP}(\Delta x[k_j], \text{Trails}[i][j]) \cdot \prod_{j=t+1}^{t+r} \text{DP}(\text{Trails}[i][j], \Delta y[m_{j-t}]).
\]

(6)

For definiteness let’s sort the table by the byte value in the first column (see Fig.2).

Let an arbitrary byte of \(\Delta x\) with an index \(k_j, 1 \leq j \leq t\) be fixed. Consider \(j\)-th column of Trails. Bytes with the same value \(x\) will have the same probability \(\text{DP}(\Delta x[k_j], x)\). Similarly for \(\Delta y\). Let us denote the corresponding table by \(\text{DP}^*\) (Trails), where

\[
\text{DP}^*\left(\text{Trails}[i][j]\right) = \text{DP}(\Delta x[k_j], \text{Trails}[i][j]), 1 \leq i \leq T, 1 \leq j \leq t,
\]

\[
\text{DP}^*\left(\text{Trails}[i][j]\right) = \text{DP}(\text{Trails}[i][j], \Delta y[m_{j-t}]), 1 \leq i \leq T, t < j \leq t + r.
\]

(7)

We will divide table columns into 3 groups (subtables). The group \(C\) contains exactly 1 column. In the group Trails\(_I\) there are \(u\) columns. The third group has \(v\) columns, \(1 + u + v = n + 2\).

\[
\text{Trails} = C || \text{Trails}_I || \text{Trails}_II,
\]

\[
\text{DP}^*(\text{Trails}) = \text{DP}^*(\text{Trails}_I) || \text{DP}^*(\text{Trails}_I) || \text{DP}^*(\text{Trails}_II),
\]

(8)

where || is concatenation. We also denote

\[
\text{Block}_j = \{\text{Trails}_I[i] || \text{Trails}_II[i] : C[i] = j, 1 \leq i \leq T\}, j \in \mathbb{F}_{2^8}^*.
\]

(9)
3.3 DDT simplification

Let all elements in each row (column) of the DDT be sorted in nonincreasing order. The row and the column with zero indexes are ignored. Let us denote the such table $DDT_{row}$ ($DDT_{col}$ correspondingly)

$$
DDT_{row}[x][1] \geq DDT_{row}[x][2] \geq \ldots \geq DDT_{row}[x][2^8-1], \quad x \in \mathbb{F}_{2^8}^*,
$$
$$
DDT_{col}[1][y] \geq DDT_{col}[2][y] \geq \ldots \geq DDT_{col}[2^8-1][y], \quad y \in \mathbb{F}_{2^8}^*.
$$

We define sequences $m_x, m_y$ and $m$ as

$$
m_x[i] = \max_{a \in \mathbb{F}_{2^8}^*} DDT_{row}[a][i], \quad m_y[i] = \max_{a \in \mathbb{F}_{2^8}^*} DDT_{col}[i][a], \quad i \in \mathbb{F}_{2^8}^*, \quad (10)
$$
$$
m[i] = \max(m_x[i], m_y[i]), \quad 1 \leq i \leq 2^8 - 1.
$$

The sequence $m$ is «greater» than any sorted nontrivial row/column of the DDT. Let $r$ be any nontrivial sorted row/column of the DDT. Then, $m[i] \geq r[i], 1 \leq i \leq 2^8-1$. Denote $\nu_{\text{max}}(m) = \#\{i : m[i] = p_{\text{max}}, 1 \leq i \leq 2^8-1\}$.

Note, that $\sum_{i=1}^{2^8-1} m[i] \geq 1$.

We also define the sequences $\rho, \rho_x, \rho_y$ as follows. Let $\rho_x$ ($\rho_y$) be one of the nontrivial sorted row (column) of the DDT. The sequence $\rho_x$ ($\rho_y$) must be greater than any other sorted row (column) of the DDT under the conditions of Lemma 2.

$$
\sum_{i=1}^{2^8-1} \rho_x[i] = \sum_{i=1}^{2^8-1} \rho_y[i] = 1. \quad \text{If } \rho_x \text{ is greater than } \rho_y \text{ under the conditions of Lemma 2, then } \rho = \rho_x \text{ otherwise } \rho = \rho_y.
$$

3.4 Constraints

We formulate a Lemma giving us some restrictions on the set of code-words.
Lemma 5. Let table Trails\(_\text{II}\) and sequence \(m\) be given as above. The table \(\text{DP}^* (\text{Trails}_{\text{II}})\) is defined by analogy with (7). Let us denote \(\omega_l (\text{DP}^* (\text{Trails}_{\text{II}}))\) the number of rows containing exactly \(l\) elements \(p_{\text{max}}\):

\[
\omega_l (\text{DP}^* (\text{Trails}_{\text{II}})) = \#\{i : \#\{j : \text{DP}^* (\text{Trails}_{\text{II}}[i][j]) = p_{\text{max}}, 1 \leq j \leq v\} = l, \ 1 \leq i \leq T\}.
\]

Then

\[
\omega_2 \leq \left(\frac{v}{2}\right) \cdot \left(\nu_{\text{max}}(m)\right)^2,
\]

and finally

\[
\sum_{l=2}^{v} \omega_l \cdot \left(\frac{l}{2}\right) \leq \left(\frac{v}{2}\right) \cdot \left(\nu_{\text{max}}(m)\right)^2.
\]

Proof. Let’s consider two arbitrary columns of Trails\(_\text{II}\). These columns do not contain any identical byte pairs. The total number of different byte pairs does not exceed \(T \leq \left(2^8 - 1\right)^2\). In each column not more than \(\nu_{\text{max}}(m)\) values are mapped in \(p_{\text{max}}\). Hence, not more than \((\nu_{\text{max}}(m))^2\) byte pairs are mapped in \((p_{\text{max}}, p_{\text{max}})\). The number of ways to select 2 columns is \(\binom{v}{2}\).

Thus we have (12).

Suppose there is a row containing 3 elements \(p_{\text{max}}\). Then \(\binom{3}{2} = 3\) pairs of columns are generated, each of which contains a pair \((p_{\text{max}}, p_{\text{max}})\). Similarly for rows with \(l\) elements \(p_{\text{max}}\). Each of them «takes» \(\binom{l}{2}\) pairs. Thereby we obtain (13).

3.5 Bounds on \(\text{DP}^*(\text{Block})\)

Suppose that we are given an arbitrary \(\text{Block} \in \{\text{Block}_j, j \in F_{2^8}^*\}\). The block dimensions are \(h \cdot (n + 1)\), \(h \leq 2^8 - 1\). We will give an upper bound on Block’s contribution to the differential \(\sum_{i=1}^{h} \prod_{j=1}^{n+1} \text{DP}^* (\text{Block}[i][j])\). We will use Lemmas 2, 3, 4.

Consider \(v = 0\) and \(u = n + 1\). Then we have

\[
\sum_{i=1}^{h} \prod_{j=1}^{u} \text{DP}^* (\text{Block}[i][j]) \leq \max \left(\max_{x \in F_{2^8}} \sum_{i=1}^{2^8-1} (\text{DDT}[x][i])^u, \max_{y \in F_{2^8}} \sum_{i=1}^{2^8-1} (\text{DDT}[i][y])^u\right).
\]

The inequality (14) is so-called «FSE 2003 bound» on MEDP [14]. Lemma 2 allows us to select a row (column) that maximizes expression (14). Then
we can rewrite inequality (14)

\[
\sum_{i=1}^{h} \prod_{j=1}^{u} \text{DP}^* \left( \text{Block}[i][j] \right) \leq \sum_{i=1}^{2^8-1} (\rho[i])^u.
\]  

(15)

Let \( v > 0 \). We will divide Block into two parts:

\[
\text{Block} = \text{Block}_I \| \text{Block}_II,
\]

\[
\sum_{i=1}^{h} \prod_{j=1}^{n+1} \text{DP}^* \left( \text{Block}[i][j] \right) = \sum_{i=1}^{h} \prod_{j=1}^{u} \text{DP}^* \left( \text{Block}_I[i][j] \right) \prod_{j=1}^{v} \text{DP}^* \left( \text{Block}_II[i][j] \right),
\]

where \( \text{Block}_I \) contains \( u \) columns, and \( \text{Block}_II \) contains \( v \) columns, \( u + v = n + 1 \). We will evaluate the contribution of \( \text{Block}_I \) by using the sequence

\[
(\rho[1])^u, (\rho[2])^u, \ldots, (\rho[2^8 - 1])^u.
\]

(17)

We will also get a bound on the contribution of \( \text{Block}_II \) by using Lemma 3. Suppose that each column of \( \text{DP}^* \left( \text{Block}_II \right) \) contains elements from the sequence \( m \). Assume also that we know

\[
\omega_l \left( \text{DP}^* \left( \text{Block}_II \right) \right) = \#\left\{ i : \#\{ j : \text{DP}^* \left( \text{Block}_II[i][j] \right) = p_{\text{max}}, 1 \leq j \leq v \} = l, 1 \leq i \leq h \right\},
\]

\[
0 \leq l \leq v,
\]

\[
\sum_{l=1}^{v} \omega_l \cdot l \leq \nu_{\text{max}}(m) \cdot v.
\]

(18)

In other words, \( \omega_l \) is the number of rows containing exactly \( l \) elements \( p_{\text{max}} \).

Let \( \text{Block}_II \) be a table obtained by the reordering procedure from Lemma 3. Then we get

\[
\sum_{i=1}^{h} \prod_{j=1}^{v} \text{DP}^* \left( \text{Block}_II[i][j] \right) \leq \sum_{i=1}^{h} \prod_{j=1}^{v} \text{DP}^* \left( \text{\~Block}_II[i][j] \right)
\]

Thanks to Lemma 4, we finally obtain

\[
\sum_{i=1}^{h} \prod_{j=1}^{n+1} \text{DP}^* \left( \text{Block}[i][j] \right) \leq \sum_{i=1}^{h} \prod_{j=1}^{v} \text{DP}^* \left( \text{\~Block}_II[i][j] \right).
\]

(19)

Thus, if we know the distribution \( \omega_l \), \( 0 \leq l \leq v \), then we can calculate the upper bound on \( \sum_{i=1}^{h} \prod_{j=1}^{n+1} \text{DP}^* \left( \text{Block}[i][j] \right) \).
3.6 Optimization problem

Let’s will form all possible sets

\[ s_i = \{(l, \omega_l), 0 \leq l \leq v\}, \ 1 \leq i \leq N. \]  
(20)

For each set \( \sum_{l=1}^{v} \omega_l \cdot l = \nu_{\text{max}}(m) \cdot v \) is true. In fact, we construct all possible partitions of the number \( \nu_{\text{max}}(m) \cdot v \). The maximum term in the partition does not exceed \( v \).

For each set \( s_i \), calculate the estimate \( \pi_i \) using (19) and \( \zeta_i \) for constraints (13):

\[ \zeta_i = \sum_{l=2}^{v} \omega_l \cdot \binom{l}{2}. \]

We can choose such \( u \) and \( v \), which would minimize the final estimation. For most practical cases we use \( u = 1 \) and \( v = n \). We get a set of pairs

\[ (\pi_1, \zeta_1), (\pi_2, \zeta_2), \ldots, (\pi_N, \zeta_N'). \]  
(21)

Pairs with the same \( \zeta_i \) value can be removed. The pair with the largest \( \pi_i \) must be left. Hence \( N' \leq \left( \binom{v}{2} \right) \cdot \left( \nu_{\text{max}}(m) \right)^2 \).

We can estimate the first column of \( \text{DP}^* \) (Trails\( \Pi \)) using the sequence \( \rho_x \) (or \( \rho_y \)). Due to the fact that \( \text{wt}(\Delta x) \geq 1 \) and \( \text{wt}(\Delta y) \geq 1 \), we can choose \( \rho_x \) or \( \rho_y \). We will choose so as to minimize the final value. For certainty, we assume that \( \rho_x \) has been chosen.

Denote \( I = i_1, i_2, \ldots, i_{2^8-1}, 1 \leq i_j \leq N', 1 \leq j \leq 2^8 - 1 \). Then

\[ \text{MEDP}_{B+1} \leq \overline{\text{MEDP}}_{B+1} \leq \max_I \sum_{j=1}^{2^8-1} \rho_x[j] \cdot \pi_{i_j} \text{ and } \sum_{i \in I} \zeta_i \leq \left( \binom{v}{2} \right) \cdot \left( \nu_{\text{max}}(m) \right)^2. \]  
(22)

The optimal \( I \) is chosen by us using dynamic programming (see non-optimized version of the pseudocode in Appendix A, Algorithm 2).

There is a trivial estimate on \( \text{MEDP}_{B+2} \leq \sum_{i=1}^{2^8-1} \rho[i] \cdot \text{MEDP}_{B+1} = \overline{\text{MEDP}}_{B+1} \). Similar can be done for \( \text{MEDP}_{B+3} \) etc. Thus, we proved that \( \text{MEDP}_{B+1}^+ \leq \overline{\text{MEDP}}_{B+1} \).

3.7 Another constraints

We can compute the estimate on \( \text{MEDP}_{B+1}^+ \) more precisely.

Consider the table \( \text{DP}^* \) (Trails\( \Pi \)). The number of rows that contains many elements \( p_{\text{max}} \) is quite small.

Recall that \( \text{wt}(\text{Trails}_{\Pi}[i] \oplus \text{Trails}_{\Pi}[j]) \geq v - 1, i \neq j \). Otherwise, there is a codeword \( c \in C_L \), \( \text{wt}(c) < B \). Thus, any two rows of Trails\( \Pi \) have exactly one equal byte, or these rows do not have any matches.
In each column of Trails$_{II}$, no more than $\nu_{\max}(m)$ bytes are mapped in $p_{\max}$. Trails$_{II}$ has $v$ columns. Denote $W = \nu_{\max}(m) \cdot v$.

Suppose that some row of DP$^*$ (Trails$_{II}$) contains $w_1$ elements $p_{\max}$.

Let’s say $w_1$ bytes of $W$ were involved. Let the other row contain $w_2$ elements $p_{\max}$. These two rows can intersect at most one byte. Therefore, at least $w_2 - 1$ bytes are selected from $W$. The third row can intersect with the first and the second rows. Hence we subtract $w_3 - 2$ from $W$. Continue until $W \geq 0$.

Let us have a series $w_1, w_2, w_3, \ldots, w_T$ sorted in nonincreasing order, where $T$ is the number of rows in Trails$_{II}$. Then

$$
\left( W - \sum_{i=1}^{l} (w_i - (i - 1)) \right) \geq 0
$$

must be true for all $l \leq T$.

Let’s form all series $\psi = w_1, w_2, \ldots, w_l$ for which the inequality (23) is true. Denote the set of such series by $\Psi$. We will use a relatively small value of $l$ (about 5, 6).

We can modify the algorithm from Subsection 3.6 as follows. For each set $s_i$ from (20), we form a series $\psi = w_1, w_2, \ldots, w_l$. We obtain a sequence similar to (21): $(\pi_1, \zeta_1, \psi_1), (\pi_2, \zeta_2, \psi_2), \ldots, (\pi_N, \zeta_N, \psi_N)$.

Hence, another constraint is added to the optimization problem (22):

$$
\text{sort}_l \left( \psi_{i_1}||\psi_{i_2}||\ldots||\psi_{i_{2^8-1}} \right) \in \Psi, \ 1 \leq i_j \leq N, 1 \leq j \leq 2^8 - 1,
$$

where $\text{sort}_l$ is $l$ largest elements of the sequence. Note that we do not need to store the entire sequence $\psi_{i_1}||\psi_{i_2}||\ldots||\psi_{i_{2^8-1}}$ in memory. We only need the first $l$ values. Using the limitations described in this subsection requires a lot of computing resources. Therefore, this modification is not used in the calculation of bound on MEDP$^+_{B+2}$.

3.8 Computing MEDP$^+_{B+2}$ and other

Let us have $(\Delta x, \Delta y)$ such that $\text{wt}(\Delta x) + \text{wt}(\Delta y) = B + 2 = n + 3$. Then Lemma 5 can be reformulated by analogy as follows.

Lemma 6. Let the conditions of Lemma 5 be hold, but weight of the differential be equal to $n + 3$. Then

$$
\sum_{l=3}^{v} \omega_l \cdot \left( \begin{array}{c} l \\ 3 \end{array} \right) \leq \left( \begin{array}{c} v \\ 3 \end{array} \right) \cdot (\nu_{\max}(m))^3.
$$

(24)
The algorithm is similar to Subsection 3.6, but the optimization problem is solved in two steps. As in Subsection 3.6:

- form all possible sets
  \[ s_i = \{ (l, \omega_l), 0 \leq l \leq v \}, 1 \leq i \leq N, \sum_{l=1}^{v} \omega_l \cdot l = v \cdot \nu_{\max}(m) \cdot v; \]
- for each set \( s_i \), calculate the estimate \( \pi_i \) by (19); \( \xi_i = \sum_{l=2}^{v} \omega_l \cdot \binom{l}{2} \);
  \[ \eta_i = \sum_{l=3}^{v} \omega_l \cdot \binom{l}{3}. \]

We obtain the sequence \( \pi_1, \xi_1, \eta_1 \), \( \pi_2, \xi_2, \eta_2 \), \ldots, \( \pi_N, \xi_N, \eta_N \).

Let’s solve first optimization problem for all values \( \eta' \leq \binom{3}{3} \cdot (\nu_{\max}(m))^3 \). Denote \( I = i_1, i_2, \ldots, i_{2^8-1}, i_j \in \mathbb{N}, 1 \leq j \leq 2^8 - 1. \)

\[ \pi' = \max_i \sum_{j=1}^{2^8-1} \rho_x[j] \cdot \pi'_{ij}, \quad \text{under condition} \quad \sum_{i \in I} \xi_i \leq \binom{v}{2} \cdot (\nu_{\max}(m))^2 \quad \text{and} \quad \sum_{i \in I} \eta_i = \eta'. \]

We can get all the values \( \eta' \) by solving the optimization problem once. Thus, the sequence \( \pi'_1, \eta'_1, \pi'_2, \eta'_2, \ldots, \pi'_N, \eta'_N \) will be obtained, \( N' \leq \binom{3}{3} \cdot (\nu_{\max}(m))^3 \).

We will solve the second optimization problem

\[ \text{MEDP}^+_{B+2} \leq \text{MEDP}^+_{B+2} = \max_I \sum_{j=1}^{2^8-1} \rho_x[j] \cdot \pi'_ij \quad \text{and} \quad \sum_{i \in I} \eta'_i \leq \binom{v}{3} \cdot (\nu_{\max}(m))^3. \]

The pseudocode in Appendix A contains a non-optimized version of the algorithm. Application of the described approach is computationally infeasible for \( \text{MEDP}^+_{B+3} \) in most cases. Furthermore, the potential estimation shift is very small (see summary table I).

4 New bounds on MEDP for 2-round Kuznyechik

Kuznyechik block cipher [1] consists of a sequence of 9 rounds and a post-whitening key addition. The block size is 128 bits \( (n = 16 \text{ bytes}) \), the key has a size of 256 bits. The cipher Sbox has no explicit analytical form [19], such as in AES. The rows and columns of the DDT have different unbalanced distributions. The sequence \( m_y \) is «greater» than \( m_x \). L-transformation is defined as a LFSR over \( F_{2^8} \), the differential branch number \( B = n + 1 \).

In [18] was proved that each 2-round best differential contains only one differential trail

\[ \text{MEDP} = \text{MEDP}_B = \max_{\Omega \neq 0} \text{EDCP}(\Omega) = \left( \frac{8}{256} \right)^{13} \left( \frac{6}{256} \right)^4 = 2^{-86.66\ldots}. \]

Using the proposed algorithms we showed that

\[ \text{MEDP}^+_{B+1} \leq 2^{-87.54\ldots}, \quad \text{MEDP}^+_{B+2} \leq 2^{-88.34\ldots}. \]
The calculation \( \text{MEDP}^{+}_{B+1} \) and \( \text{MEDP}^{+}_{B+2} \) used the fact that \( \text{wt}(\Delta x) \geq 2 \). We can use \( \rho_x \) instead of \( \rho \) (the rows of DDT instead the columns) in at least two coordinates. Obtained bound on \( \text{MEDP}^{+}_{B+3} \) will be not less than \( 2^{-88.42} \).

Table 1 shows all computed values. The numbers are rounded to the second decimal place. The second data column presents the bounds we obtained using «FSE 2003 bounds» [12]. The last column (*) shows the limitation on the capabilities of the presented algorithm. For information about the linear method, see Appendix B.

<table>
<thead>
<tr>
<th>( (p_{\text{max}})^B )</th>
<th>( \text{FSE2003 MEDP}_B \leq )</th>
<th>( \text{MEDP}_B = )</th>
<th>( \text{MEDP}^{+}_{B+1} \leq )</th>
<th>( \text{MEDP}^{+}_{B+2} \leq )</th>
<th>( (*) \text{MEDP}^{+}_{B+3} \leq )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-85)</td>
<td>(-83.97)</td>
<td>(-86.66)</td>
<td>(-87.54)</td>
<td>(-88.34)</td>
<td>(-88.42)</td>
</tr>
<tr>
<td>( (p_{\text{lin}}\text{.max})^B )</td>
<td>( \text{FSE2003 MELP}_B \leq )</td>
<td>( \text{MELP}_B = )</td>
<td>( \text{MELP}^{+}_{B+1} \leq )</td>
<td>( \text{MELP}^{+}_{B+2} \leq )</td>
<td>( (*) \text{MELP}^{+}_{B+3} \leq )</td>
</tr>
<tr>
<td>(-74.54)</td>
<td>(-73.54)</td>
<td>(-76.73)</td>
<td>(-77.15)</td>
<td>(-79.63)</td>
<td>(-80.50)</td>
</tr>
</tbody>
</table>

Table 1: Summary table of results for Kuznyechik (log_2 scale).

5 Conclusion

We propose a dynamic programming algorithm for bounding non-minimum weight differentials (linear hulls) in 2-round LSX-ciphers. Thanks to the presented algorithm, we derive some new bounds on differentials and linear hulls for 2-round Kuznyechik (Table 1). Similar results were obtained for 2-round Khazad (Table 2), and as a result, the exact values of MEDP = \( 2^{-45} + 2^{-60} \) and MELP = \( 2^{-37.80} \) are also proved.

The source codes of the presented algorithms can be found at:

https://gitlab.com/v.kir/diff2rLSX

For any LSX-cipher with independent round keys, the \( R \)-round MEDP (MELP) is the upper bound for \( (R+1) \)-round MEDP (MELP). The presented results are a step towards obtaining new nontrivial bounds on \( R \)-round MEDP (MELP), i.e. new proofs of Kuznyechik strength against differential and linear cryptanalysis.

References


A Pseudocode of algorithms

Require: $(\pi_1, \zeta_1), (\pi_2, \zeta_2), \ldots, (\pi_{N'}, \zeta_{N'})$, and $\rho_x$, and $s = \binom{v}{2} \cdot \left(\nu_{\max}(m)\right)^2$

Ensure: $\text{MEDP}_{B+1}$

1: $\tilde{\rho}_x := \text{nondecreasing\_sort}(\rho_x) // 0, \ldots, 0, \frac{2}{256}, \ldots, p_{\max}$
2: $\rho_x := \text{nonzero\_elements}(\tilde{\rho}_x) // \frac{2}{256}, \ldots, p_{\max}$
3: state$[s] := [0, \ldots, 0]$ // indexing from 0
4: for $j := 1$ to len($\tilde{\rho}_x$) do
5: new\_state$[s] := [0, \ldots, 0]$ // indexing from 0
6: pr$_x := \tilde{\rho}_x[j]$
7: for $c := 0$ to $s$ do
8: for $i := 1$ to $N'$ do
9: pr := pr$_x \cdot \pi_i + \text{state}[c]$
10: pairs := $\zeta_i + c$
11: if pairs $\leq s$ then
12: if new\_state[pairs] $< pr$ then
13: new\_state[pairs] := pr
14: end if
15: end if
16: end for
17: end for
18: state := new\_state
19: end for
20: return $\text{max}(\text{state})$

Algorithm 2: Computing $\text{MEDP}_{B+1}$

The pseudocode above (Algorithm 2) contains a non-optimized version of the algorithm. The complexity of the algorithm is

$$O \left( \text{len}(\tilde{\rho}_x) \cdot N' \cdot \binom{v}{2} \cdot \left(\nu_{\max}(m)\right)^2 \right),$$

where $\text{len}(\tilde{\rho}_x)$ is a number of nonzero elements in $\rho_x$.

If $v = 16$, $\nu_{\max}(m) = 2$, $\text{len}(\tilde{\rho}_x) \leq 2^7$ (Kuznyechik), then the approximate number of operations is $2^{25}$ (less than a minute on a common PC). The number of distinct pairs $N' = 7665$. 

18
Require: \( (\pi_1, \zeta_1, \eta_1), (\pi_2, \zeta_2, \eta_2), \ldots, (\pi_N, \zeta_N, \eta_N) \), and \( \rho_x \), and
\( s_{\text{pairs}} = \binom{N}{2} \cdot \left( \nu_{\text{max}}(m) \right)^2 \), \( s_{\text{triplets}} = \binom{N}{3} \cdot \left( \nu_{\text{max}}(m) \right)^3 \)

Ensure: MEDP\(_{B+2}\)

1: \( \tilde{\rho}_x := \text{nondecreasing}\_\text{sort}(\rho_x) \) // 0, \ldots, 0, \frac{2}{256}, \ldots, p_{\text{max}} 
2: \( \tilde{\rho}_x := \text{nonzero}\_\text{elements}(\tilde{\rho}_x) \) // \frac{2}{256}, \ldots, p_{\text{max}} 
3: \( \text{state}[s_{\text{pairs}}][s_{\text{triplets}}] := [0, \ldots, 0] \) // indexing from 0,0 
4: \( \text{for } j := 1 \text{ to } \text{len}(\tilde{\rho}_x) \text{ do} \)
5: \( \text{new}\_\text{state}[s_{\text{pairs}}][s_{\text{triplets}}] := [0, \ldots, 0] \) // indexing from 0,0 
6: \( \text{for } c_{\text{pairs}} := 0 \text{ to } s_{\text{pairs}} \text{ do} \)
7: \( \text{for } c_{\text{triplets}} := 0 \text{ to } s_{\text{triplets}} \text{ do} \)
8: \( \text{for } i := 1 \text{ to } N \text{ do} \)
9: \( \text{pr} := \text{pr}_x \cdot \pi_i + \text{state}[c_{\text{pairs}}][c_{\text{triplets}}] \)
10: \( \text{pairs} := \zeta_i + c_{\text{pairs}} \)
11: \( \text{triplets} := \eta_i + c_{\text{triplets}} \)
12: \( \text{if } \text{pairs} \leq s_{\text{pairs}} \text{ and } \text{triplets} \leq s_{\text{triplets}} \text{ then} \)
13: \( \text{if } \text{new}\_\text{state}[\text{pairs}][\text{triplets}] < \text{pr} \text{ then} \)
14: \( \text{new}\_\text{state}[\text{pairs}][\text{triplets}] := \text{pr} \)
15: \( \text{end if} \)
16: \( \text{end if} \)
17: \( \text{end for} \)
18: \( \text{end for} \)
19: \( \text{end for} \)
20: \( \text{state} := \text{new}\_\text{state} \)
21: \( \text{end for} \)
22: \( (\pi'_1, \eta'_1), \ldots, (\pi'_N, \eta'_N) := (\text{state}[s_{\text{pairs}}][0], 0), \ldots, (\text{state}[s_{\text{pairs}}][s_{\text{triplets}}], s_{\text{triplets}}) \)
23: \( \text{return call Algorithm 2 } ((\pi'_1, \eta'_1), (\pi'_2, \eta'_2), \ldots, (\pi'_N, \eta'_N), \rho_x, s = s_{\text{triplets}}) \)

Algorithm 3: Computing MEDP\(_{B+2}\)

The complexity of Algorithm 3 is estimated as trivial as Algorithm 2. If \( v = 16, \nu_{\text{max}}(m) = 2, \text{len}(\tilde{\rho}_x) \leq 2^7 \), then \( N = 7665 \) and the approximate number of operations is \( 2^{41} \) (about an hour on common PC).

B Application to Linear Cryptanalysis

There is a certain duality between differential and linear cryptanalysis [4]. It allows us to apply the algorithms described above to calculate linear characteristics.

We make the appropriate substitutions.

Differential probability (DP, EDP, EDCP, MEDP) is replaced by linear probability (LP, ELP, ELCP, MELP correspondingly). DDT is replaced by Linear Approximation Table (LAT). Input/output differences \( \Delta x \) and \( \Delta y \)
are replaced by input/output masks $\mu_x$ and $\mu_y$ correspondingly.

$$LP(\mu_x,\mu_y) = \left(2 \Pr(\mu_x \cdot x = \mu_y \cdot f(x)) - 1\right)^2, \mu_x,\mu_y \in \mathbb{F}_2^l, f : \mathbb{F}_2^l \rightarrow \mathbb{F}_2^l, $$

where $\cdot$ is the inner product over $\mathbb{F}_2$, and $x \in \mathbb{F}_2^l$ is a uniformly distributed random variable.

Differential branch number is replaced by linear branch number. If a linear transformation generates an MDS code both values are equal to $n + 1$.

The value $p_{\text{max}} = \max_{a \neq 0, b} \text{DDT} [a][b]$ is replaced by

$$p_{\text{lin, max}} = \max_{a \neq 0, b} \text{LAT} [a][b] = LP(a, b), a, b \in \mathbb{F}_2^8.$$

By analogy with the differential trail a linear characteristic $\Omega = (\mu_x, \mu_1, \mu_2, \mu_y)$ for 2 rounds is introduced. $ELP(\Omega)$ is equal to

$$ELP(\Omega) = \left( \prod_{j=1}^{n} LP(\mu_x[j], \mu_1[j]) \right) \left( \prod_{j=1}^{n} LP(\mu_2[j], \mu_y[j]) \right),$$

where $\mu_2 = L^T \cdot \mu_1$, $L$ is a binary matrix such that $y = L(x) = L \cdot x$ and $L^T$ is a transposed matrix.

The linear code $C_L$ is replaced by the code $C_{L^T}$.

The linear hull (similar to differential) is the set of all linear characteristics having input mask $\mu_x$ and output mask $\mu_y$.

The expected probability of the 2-round linear hull $(\mu_x, \mu_y)$ is equal to:

$$ELP (\mu_x, \mu_y) = \sum_{(\mu_1, \mu_2) \in \mathbb{F}_2^{2 \times n}} \left( \prod_{j=1}^{n} LP(\mu_x[j], \mu_1[j]) \right) \left( \prod_{j=1}^{n} LP(\mu_2[j], \mu_y[j]) \right)$$

$$\text{and}$$

$$MELP = \max_{\mu_x \neq 0, \mu_y} ELP(\mu_x, \mu_y).$$

(25)

In order to go to linear cryptanalysis, one needs to replace all formulas in Section 3 according to the above analogies.

For 2-round Kuznyechik the only best linear hull containing 37 linear characteristics $\Omega_1, \Omega_2, \ldots, \Omega_{37}$ is found [18].

$$MELP = MELP_\mathcal{B} = \sum_{i=1}^{37} = ELCP(\Omega_i) = 2^{-76.73 \ldots}.$$  

We show that

$$MELP_{\mathcal{B}+1}^+ \leq 2^{-77.15 \ldots}, \quad MELP_{\mathcal{B}+2}^+ \leq 2^{-79.63 \ldots}.$$  

A bound on $MELP_{\mathcal{B}+3}^+$ will be not less than $2^{-80.50 \ldots}$.  

20
C Khazad

Khazad [12] is a 64-bit \((n = 8\text{ byte})\) block cipher using a 128-bit key. It is an 8-round SP network. The plaintext is initially XORed with the whitening key and then undergoes 8 identical rounds.

S-transformation and L-transformation are involutions, \(S = S^{-1}, L = L^{-1}\).

The sequences \(m_x\) and \(m_y\) are equal (see definition 10).

Due to this involution structure, we can consider only half of the subsets of codewords. Let’s assume that for some 2-round differential \((\Delta x, \Delta y)\) we know the value of \(\text{EDP}(\Delta x, \Delta y)\). Then we know the value of \(\text{EDP}(\Delta y, \Delta x) = \text{EDP}(\Delta x, \Delta y)\).

We have shown that each best differential contains two differential trails \(\Omega_1\) and \(\Omega_2\).

\[
\text{EDCP}(\Omega_1) = p_{\text{max}}^B = \left(\frac{8}{256}\right)^9 = 2^{-45}, \quad \text{EDCP}(\Omega_2) = 2^{-60}.
\]

Eight best differentials \((\Delta x, \Delta y)\) and eight differentials \((\Delta y, \Delta x)\) were found. For each of them \(\text{MEDP}_B = \text{EDP}(\Delta x, \Delta y) = \text{EDP}(\Delta y, \Delta x) = \text{EDCP}(\Omega_1) + \text{EDCP}(\Omega_2)\).

We proved that \(\text{MEDP}_{B+1}^+ \leq 2^{-44.99}\) and with improvements described in Subsection 3.7 \(\text{MEDP}_{B+1}^+ \leq 2^{-45.02}\). Using algorithm from Subsection 3.8, we get \(\text{MEDP}_{B+2}^+ \leq 2^{-45.09}\). Thus

\[
\text{MEDP} = \text{MEDP}_B = 2^{-45} + 2^{-60}.
\]

We also found 16 best linear hulls: eight in the form \((\mu_x, \mu_y)\) and eight in the form \((\mu_y, \mu_x)\). Each of them contains 108 linear characteristics \(\Omega_1, \Omega_2, \Omega_3, ..., \Omega_{108}\).

\[
\text{ELCP}(\Omega_1) = 2^{-37.80...} < p_{\text{lin,max}}^B = 2^{-36}, \quad \text{ELCP}(\Omega_2) = 2^{-67.70...}.
\]

\[
\text{MELP}_B = \sum_{i=1}^{108} \text{ELCP}(\Omega_i) = 2^{-37.80...}.
\]

\[
\text{MELP}_{B+1}^+ \leq 2^{-37.83...}, \quad \text{MELP}_{B+2}^+ \leq 2^{-37.92...}.
\]

Because of this, we get

\[
\text{MELP} = \text{MELP}_B = 2^{-37.80...}.
\]

The obtaining of \(\text{MEDP}_{B+3}^+\) and \(\text{MELP}_{B+3}^+\) is computationally infeasible task for us. Furthermore, the result of the algorithm will be not less than \(2^{-45.11...}\) and \(2^{-37.94...}\) respectively.
Table 2: Table of results (log₂ scale).

The best differentials

We show only 8 of the 16 differentials \((\Delta x, \Delta y)\). The remaining differentials \((\Delta y, \Delta x)\) can be easy obtained by swapping \(\Delta x\) and \(\Delta y\).

<table>
<thead>
<tr>
<th>(\Delta x)</th>
<th>(1208f00000000000f)</th>
<th>(\log_2 \text{EDCP}(\Omega_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Omega_1)</td>
<td>1248f00000000000f</td>
<td>0000b548fbeb4800</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>c8070a000000002300</td>
<td>0000130753a60700</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>0000bf0818910800</td>
<td>000008bf91180008</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>081200f0000000f00</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>481200f0000000f00</td>
<td>000048b5ebfb0048</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>07c8000a00002300</td>
<td>00000713a6530007</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>000008bf91180008</td>
<td></td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>f0001208000000f00</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>f0001248000000f00</td>
<td>b54800004800fbeb</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>0a00c80700230000</td>
<td>13070000070053a6</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>0000bf0818910800</td>
<td>bf08000008001891</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>00f008120f000000</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>00f048120f000000</td>
<td>48b500000048ebfb</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>000a07c823000000</td>
<td>07130000007a653</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>000008bf91180008</td>
<td>08bf000000089118</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>0000f0000f00000f00</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>0f00000000f048120</td>
<td>0048ebfb48b50000</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>23000000000a07c8</td>
<td>007a65307130000</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>000008bf91180008</td>
<td>0008911808bf0000</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>0000f0000f00000f00</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>000f0000f00001248</td>
<td>4800fbeb5480000</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>0230000000a00c807</td>
<td>070053a613070000</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>000f0000f00001248</td>
<td>08001891bf080000</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>0000f0000f00001208f</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>000f0000f0001208f</td>
<td>4800fbeb5480000</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>0023000000a00c807</td>
<td>070053a613070000</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>000f0000f0001208f</td>
<td>08001891bf080000</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>00000000000f1208f000</td>
<td>(\log_2 \text{EDCP}(\Omega_i))</td>
</tr>
<tr>
<td>(\Omega_1)</td>
<td>00000000000f1208f000</td>
<td>c8070a0000002300</td>
</tr>
<tr>
<td>(\Omega_2)</td>
<td>00000000000f1248f000</td>
<td>53a607000001307</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>00000000000f1208f000</td>
<td>189108000000bf08</td>
</tr>
</tbody>
</table>

Table 3: The best 2-round Khazad differentials
The best linear hulls

As in the previous subsection, we show only 8 of the 16 linear hulls.

<table>
<thead>
<tr>
<th>$\Omega_i$</th>
<th>$\mu_x$</th>
<th>$\mu_y$</th>
<th>$\log_2 \text{ELCP}(\Omega_i)$</th>
<th>$\Omega_i$</th>
<th>$\mu_x$</th>
<th>$\mu_y$</th>
<th>$\log_2 \text{ELCP}(\Omega_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6f07e80000000500</td>
<td>0006f0eb400e153</td>
<td>-37.80</td>
<td>22</td>
<td>e964e0000004000</td>
<td>0000e6f00b453e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>2</td>
<td>8e06f0705000000</td>
<td>0000e6f0eb400e153</td>
<td>-67.71</td>
<td>23</td>
<td>b1476b0000071000</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>3</td>
<td>6f0e00e0153b400</td>
<td>0006f0eb400e153</td>
<td>-70.37</td>
<td>24</td>
<td>2de5e000000c400</td>
<td>0000c2dfe0083c6</td>
<td>-75.71</td>
</tr>
<tr>
<td>4</td>
<td>008e076f00050000</td>
<td>0000e6f0eb400e153</td>
<td>-75.71</td>
<td>25</td>
<td>05d4a3000004300</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>5</td>
<td>000050006f07e800</td>
<td>0000e6f0eb400e153</td>
<td>-75.71</td>
<td>26</td>
<td>46528300000e6f80</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>6</td>
<td>0b4e015300006f0e</td>
<td>0000e6f0eb400e153</td>
<td>-75.71</td>
<td>27</td>
<td>da4f06e00007600</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
</tbody>
</table>

Table 4: The best 2-round Khazad linear hulls

<table>
<thead>
<tr>
<th>$\Omega_i$</th>
<th>$\mu_x$</th>
<th>$\mu_y$</th>
<th>$\log_2 \text{ELCP}(\Omega_i)$</th>
<th>$\Omega_i$</th>
<th>$\mu_x$</th>
<th>$\mu_y$</th>
<th>$\log_2 \text{ELCP}(\Omega_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6f078e0000000500</td>
<td>00006f0eb400e153</td>
<td>-37.80</td>
<td>22</td>
<td>e964e0000004000</td>
<td>0000e6f00b453e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>2</td>
<td>0006f0eb400e153</td>
<td>00006f0eb400e153</td>
<td>-67.71</td>
<td>23</td>
<td>b1476b0000071000</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>3</td>
<td>8e06f0705000000</td>
<td>00006f0eb400e153</td>
<td>-70.37</td>
<td>24</td>
<td>2de5e000000c400</td>
<td>0000c2dfe0083c6</td>
<td>-75.71</td>
</tr>
<tr>
<td>4</td>
<td>008e076f00050000</td>
<td>00006f0eb400e153</td>
<td>-75.71</td>
<td>25</td>
<td>05d4a3000004300</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>5</td>
<td>000050006f07e800</td>
<td>00006f0eb400e153</td>
<td>-75.71</td>
<td>26</td>
<td>46528300000e6f80</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
<tr>
<td>6</td>
<td>0b4e015300006f0e</td>
<td>00006f0eb400e153</td>
<td>-75.71</td>
<td>27</td>
<td>da4f06e00007600</td>
<td>0000b1fd3000e4e1</td>
<td>-75.71</td>
</tr>
</tbody>
</table>

Table 5: One of the best 2-round Khazad linear hull, $\mu_x = 6f078e0000000500$, $\mu_y = 00006f0eb400e153$ (part 1)
<table>
<thead>
<tr>
<th>$\Omega_i$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\log_2$ ELCP($\Omega_i$)</th>
<th>$\Omega_i$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\log_2$ ELCP($\Omega_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>e5bf2e400000002500000a55600a768</td>
<td>-78.68</td>
<td>76</td>
<td>a701260000007000000a7d60000818</td>
<td>-85.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>ff2c1300000004d000000f8e80000c0</td>
<td>-78.76</td>
<td>77</td>
<td>24037300000400000000e9</td>
<td>-85.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>70144800000063000000d7020000834b</td>
<td>-78.90</td>
<td>78</td>
<td>29424900000050000000d6</td>
<td>-85.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>665c50000000c000000869200006337</td>
<td>-79.02</td>
<td>79</td>
<td>930a09000000d610000039c</td>
<td>-85.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>f02e5200000065b00000f97c00a2d4</td>
<td>-79.20</td>
<td>80</td>
<td>9f3f90000000c10000005b80000762</td>
<td>-85.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>7e523400000077000007e4f600043ca</td>
<td>-79.32</td>
<td>81</td>
<td>de56a10000000e</td>
<td>-85.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>a2e0600000005000000a8a70000459</td>
<td>-79.54</td>
<td>82</td>
<td>b708e5000000d400000045b</td>
<td>-86.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>b877d100000094800000b6640d0676</td>
<td>-79.71</td>
<td>83</td>
<td>946601000000e6</td>
<td>-86.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>61f11ca000000a0</td>
<td>-79.85</td>
<td>84</td>
<td>97af66000000d61c</td>
<td>-86.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>3a0840000000b000000903a0e0171c</td>
<td>-80.03</td>
<td>85</td>
<td>4905c2000000d200000086b</td>
<td>-86.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>7607c00000006300000711b9000a4f</td>
<td>-80.15</td>
<td>86</td>
<td>8f3f85000000c0</td>
<td>-86.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>2a8e14000000260000006e5b0000d446</td>
<td>-80.25</td>
<td>87</td>
<td>be542a000000e6</td>
<td>-86.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>6b72a00000009000000e6a000004676</td>
<td>-80.34</td>
<td>88</td>
<td>b2a300000000a6</td>
<td>-86.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>75c9600000001b0000075140000c34</td>
<td>-80.37</td>
<td>89</td>
<td>9c2a5c000000e6</td>
<td>-86.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>b335f00000006af00000b55a00e4df</td>
<td>-80.83</td>
<td>90</td>
<td>897760000000c4</td>
<td>-87.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>e85f140000008f000000c67e0000d480</td>
<td>-81.02</td>
<td>91</td>
<td>a67b12000000e6</td>
<td>-87.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>1f05820000002b000000b4f20</td>
<td>-81.34</td>
<td>92</td>
<td>841320000000e6</td>
<td>-87.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>f9e4d40000004d000000f7b50000f2a</td>
<td>-81.40</td>
<td>93</td>
<td>c97850000000e4</td>
<td>-88.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>66fa300000003b000000e86a0000ec9</td>
<td>-81.85</td>
<td>94</td>
<td>a5e64000000e410000004a7</td>
<td>-88.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>6c9c70000000e8000000c6520001f66</td>
<td>-82.05</td>
<td>95</td>
<td>5c85d2000000e0</td>
<td>-89.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>7e5f60000000e6000000d6b30001d8a</td>
<td>-82.19</td>
<td>96</td>
<td>fe5820000000c4</td>
<td>-89.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>814e2000000038000000815c0000e210</td>
<td>-82.37</td>
<td>97</td>
<td>4e6780000000e6</td>
<td>-89.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>217e20000000ea000000c21600005b8</td>
<td>-82.57</td>
<td>98</td>
<td>523a87000000e8</td>
<td>-90.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>6e4e7000000039000000e42d00003c0</td>
<td>-82.82</td>
<td>99</td>
<td>682a70000000e6</td>
<td>-90.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>eaf90300000031000000e6a10000956</td>
<td>-82.83</td>
<td>100</td>
<td>e08760000000f50000004a</td>
<td>-90.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>db192e0000004700000078a500107d4</td>
<td>-82.90</td>
<td>101</td>
<td>317fd100000097000001d4</td>
<td>-91.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>7b2a140000002000000074360000d5b</td>
<td>-83.66</td>
<td>102</td>
<td>739510000000e0</td>
<td>-91.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>c027170000001c000000c4f60000d700</td>
<td>-83.76</td>
<td>103</td>
<td>62f24000000e0</td>
<td>-92.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>eb8d70000000a0000000b3f60000e69</td>
<td>-83.85</td>
<td>104</td>
<td>0c5914000000e5</td>
<td>-92.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>15d5100000007e000000159200005bc</td>
<td>-84.03</td>
<td>105</td>
<td>173c79000000e4</td>
<td>-92.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>ccb8000000079000000c3050000c7e</td>
<td>-84.57</td>
<td>106</td>
<td>dc8c80000000e4</td>
<td>-93.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>2a83740000006000000283000005539</td>
<td>-84.68</td>
<td>107</td>
<td>0f0240000000e4</td>
<td>-94.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>55c0a00000000b0000005550000446b3</td>
<td>-85.02</td>
<td>108</td>
<td>3ad197000000e0</td>
<td>-97.66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: One of the best 2-round Khazad linear hull, $\mu_x = 6f078e0000000500$, $\mu_y = 00006f0eb400e153$ (part 2)