Zero-Correlation Linear Cryptanalysis of Reduced-Round LBlock

Hadi Soleimany

Aalto University School of Science,
Department of Information and Computer Science
hadi.soleimany@aalto.fi

Abstract. Zero-correlation linear attack is a new method for cryptanalysis of block ciphers. In this paper we adapt Matrix method [13] to find zero-correlation approximations. Then we present several zero-correlation linear approximations for 14 rounds of Lblock. Finally, we describe a cryptanalysis for 22 rounds of the reduced Lblock. While the previous attacks on Lblock used chosen plaintexts, the new attack needs distinct known plaintexts which is a more realistic model. Also the time complexity is 2^8 times faster than the previous attack.

Keywords: zero-correlation linear cryptanalysis, Lblock, Matrix method

1 Introduction

Differential and linear cryptanalysis are the two most prominent cryptanalysis methods against block ciphers. Several improvements of these methods have been made and applied against block ciphers. Truncated differential cryptanalysis, higher order differential cryptanalysis [15], boomerang [19] and amplified boomerang cryptanalysis [11], multiple differential [5] and impossible differential cryptanalysis [1] have been proposed inspired by differential cryptanalysis. Also some extensions of linear cryptanalysis have been introduced which usually exploit several linear approximations with high correlation simultaneously. Kaliski and Robshaw used multiple linear approximations with the same key mask [10]. The concept of linear hull presented by Nyberg which uses several linear approximation with the same input and output masks [18]. Multiple linear approximations cryptanalysis and multidimensional linear cryptanalysis are proposed in [4] and [9] respectively. Recently, a novel extension of linear cryptanalysis is proposed which uses zero-correlation linear approximations [6]. It can be seen as the counterpart of impossible differential cryptanalysis. The original proposed had the disadvantage to require almost the full codebook of data. Bogdanov et.al. proposed a framework which uses several independent zero-correlation linear approximations to reduce data complexity [7]. The assumption of independence for all linear approximations is an important challenge. Based on the multidimensional linear attack, recently a new distinguisher is proposed to eliminate independence assumption [8]. The distinguisher is supposed to use distinct known plaintexts.

In this paper the multidimensional zero-correlation linear method is applied to attack 22 rounds of Lblock [20]. Lblock is a lightweight block cipher with semi-Feistel structure. The security of the cipher has been evaluated in [12, 16, 20]. Impossible differential cryptanalysis has achieved the best results so far and has been applied up to 22 rounds of Lblock. The attack uses 2^{58} chosen plaintexts and the time complexity is $2^{79.28}$ which is almost equivalent to the exhaustive search. The authors used the Matrix method to establish the impossible differential characteristic. In this paper, we show how to use Matrix method [13, 14] to find 8×8 different classes of zero-correlation linear approximations for 14 rounds which each one includes $2^8 - 1$ different zero-correlation approximations. Based on $2^8 - 1$ zero-correlation approximations we present an attack on 22 rounds of the reduced Lblock. We also use the weaknesses in the permutation layer of Lblock to decrease the time complexity. The attack uses distinct known plaintexts. As depicted in Table 1, there is a trade-off between the time complexity and the data complexity of the attack.

Table 1. Summary of the attacks on Lblock: CP = Chosen Plaintexts, DKP = Distinct Known Plaintexts

Attack	Rounds	Data	Time	Memory (Bytes)	Source
Integral Attack (CP)	20	$2^{63.7}$	$2^{63.7}$	Not Specified	[20]
Impossible Differential (CP)	20	2^{63}	$2^{72.7}$	2^{60}	[20]
Impossible Differential (CP)	21	$2^{62.5}$	$2^{73.7}$	2^{64}	[16]
Impossible Differential (CP)	21	2^{63}	$2^{69.5}$	2^{68}	[12]
Impossible Differential (CP)	22	2^{58}	$2^{79.28}$	2^{68}	[12]
Zero Correlation (DKP)	22	2^{64}	$2^{70.54}$	2^{64}	This paper
Zero Correlation (DKP)	22	$2^{62.1}$	$2^{71.27}$	2^{64}	This paper
Zero Correlation (DKP)	22	2^{60}	2^{79}	2^{64}	This paper

The paper is structured as follows: In Section 2, we briefly describe Lblock. In Section 3 we review the previous work on zero-correlation linear cryptanalysis. In Section 4 we show how to use Matrix method as an automatic tool to find zero-correlation approximations and obtain several zero-correlation linear distinguisher for 14 rounds of the Lblock. Section 5 describes an attack on 22 rounds of Lblock. We conclude the paper in Section 6.

2 A Brief Description of Lblock

Throughout this paper we use the following notations:

Notation

- P: Plaintext
- C: Ciphertext

– Sk_i : 32-bit round key
– K_i : 80-bit in the register K in round i
– S_i : 4×4 S-box
– $\lll i$: i-bit left cyclic shift
– $[i]_2$: Binary form of an integer i
– $X(i)$: i-th nibble of X where the right most one is 0
– $X[i]$: i-th bit of X where the least significant bit (lsb) is 0
– $X[i-j]$: concatenation of $i, i-1, \cdots, j$-th bit of X where $i \geq j$
– $X(i-j)$: concatenation of $i, i-1, \cdots, j$-th nibble of X where $i \geq j$
– $|$: concatenation of two binary strings
– $L_i|R_i$: the output of the i-round of Lblock

Lblock

Lblock is a block cipher which is designed for constrained environment applications. Lblock is a variant Feistel block cipher with 32 rounds. It supports 80 secret key bits and the block size is $b = 64$ bits. One round of Lblock and the round function are depicted in Figure 1.

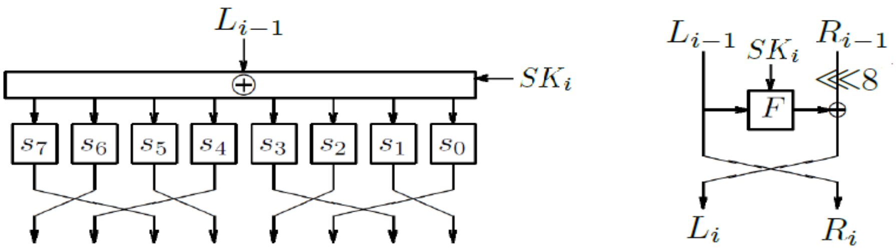

Fig. 1. One round of Lblock

Encryption Algorithm Let $P = L_0|R_0$ be a 64-bit plaintext. Then encryption process is as follows:

– For $i = 1, 2, \cdots, 31$, do
  - $R_i = L_{i-1}$
  - $L_i = F(L_{i-1}, SK_i) \oplus (R_{i1} \lll 8)$
– $L_{32} = L_{31}$, $R_{32} = F(L_{31}, SK_{32}) \oplus (R_{31} \lll 8)$
– $C = L_{32}|R_{32}$.

In our attacks on reduced-round Lblock we also consider the last round to be without swapping the halves as in the original Lblock.

Key schedule The 80-bit master key K is stored in a key register. In the i-th step, the leftmost 32 bits of current content of register K are extracted as the round key SK_i. Then the key register is updated as follows:

1. $K \lll 29$,
2. $K[79-76] = S_9[79-76]$,
3. $K[75-72] = S_8[75-72]$,
4. $K[50-46] = K[50-46] \oplus [i]_2$,
   where s_9 and s_8 are two 4-bit S-boxes.

Definitions of the S-boxes used in Lblock can be found in Appendix A. For more details we refer to [20].

3 Zero-Correlation Linear Approximation

Consider a function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and let the input of the function be $x \in \mathbb{F}_2^n$. A linear approximation with an input mask u and an output mask v is the following function:

$$x \mapsto u \cdot x \oplus v \cdot f(x).$$

The linear approximation has probability

$$p(u;v) = Pr(u \cdot x \oplus v \cdot f(x) = 0)$$

and its correlation is defined as follows:

$$c_f(u;v) = 2p(u;v) - 1.$$

In linear cryptanalysis we are interested in the linear approximation with correlation far from zero. The number of known plaintexts needed in the linear cryptanalysis is inversely proportional to the squared correlation. Zero-correlation linear cryptanalysis uses linear approximations such that the correlation is equal to zero for all keys. If the number of zero-correlation approximations is 2^m, then by [8] the number of required distinct plaintexts is about $2^{n+2-m/2}$. The key recovery can be done with the same method utilized by Matsui's Algorithm 2 [17].

To describe this process in more detail, let us describe a cipher E as a cascade $E = E_f \circ E_z \circ E_b$. Assume there exists m independent linear approximations for E_z such that all $\ell = 2^m - 1$ nonzero linear combinations of them have correlation zero. For each key candidate, the adversary encrypts the plaintexts for the beginning rounds E_b and decrypts the corresponding ciphertexts for the final rounds E_f.

For each of $i \in \mathbb{F}_2^m$ he allocates a counter T_i and computes the number of times which the corresponding data value is equal to i. Then the adversary computes the statistic T value

$$T = \sum_{i=0}^{2^m-1} \frac{(T_i - N2^{-m})^2}{N2^{-m}(1-2^{-m})}. \tag{1}$$

The value T for right key guess follows a χ^2-distribution with mean $\mu_0 = \ell\frac{2^n-N}{2^n-1}$ and variance $\sigma_0^2 = 2\ell(\frac{2^n-N}{2^n-1})^2$ while for the wrong key the distribution is a χ^2-distribution with mean $\mu_1 = \ell$ and variance $\sigma_1^2 = 2\ell$.

Let show error probability type I as α and error probability type II as β. If we consider the decision threshold $t = \mu_0 + \sigma_0 z_{1-\alpha} = \mu_1 + \sigma_1 z_{1-\beta}$ then the amount of distinct known plaintexts is as follows:

$$N = \frac{2^n(z_{1-\alpha} + z_{1-\beta})}{\sqrt{\ell/2} - z_{1-\beta}} \tag{2}$$

where $z_\gamma = \Phi^{-1}(p)$ for $0 < p < 1$ where Φ is the cumulative function of the standard normal distribution. For more details we refer to [8].

4 Matrix method

Several tools have been proposed for finding statistical distinguisher. Such tools help us to analyze algorithms systematically. A cryptanalytic tool for finding impossible differential characteristics in block ciphers with bijective function was introduced in [13, 14]. It is called Matrix method and uses "miss-in-the-middle" approach to find impossible differential characteristic. Miss-in-the-middle technique proposes to construct the impossible differential characteristic by two (truncated) differential paths with probability one and which lead to a contradiction in the middle. Matrix method is a tool for finding these paths. In this section we show this technique is also useful for finding zero-correlation linear approximation. We can follow the linear approximation patterns of input and output masks in the intermediate rounds and inquire whether no linear characteristics with non-zero-correlation exists. So the Matrix method is also useful to automate the process of finding the longest zero-correlation linear approximations.

4.1 Matrix Method for Finding Linear Approximation with Correlation Zero

The state is partitioned into n words (usually with the same length). In the linear pattern, the linear mask of each word can have five types:

1. zero mask which is denoted by 0,
2. fixed non-zero mask which is denoted by 1^*,
3. non-fixed non-zero mask which is denoted by 1,
4. the exclusive-or of a fixed non-zero mask and a non-fixed mask which is denoted by 2^*,
5. an unknown mask which is denoted by 2 or larger values (with or without $*$).

After that we describe the encryption round as a matrix $M^{a\times a}$. The matrix shows how a linear mask of each output word is affected by the linear mask of an input word. Let show the input and the output of the round by two bit strings A and B respectively. If $B(j)$ is not affected by a linear mask of $A(i)$ the value (i, j) set to 0. If a linear mask of $A(i)$ affects $B(j)$ directly the value (i, j) set to 1. Finally if $B(j)$ is affected by a linear mask of $A(i)$ after the round function the value (i, j) set to 1_F. For decryption of the round, another matrix is defined similarly. To define the matrices we can use lemmas in Appendix B which are introduced in [6] (see also [3]).

We can define the arithmetics operations by considering the definition of five possible types for a linear mask of word. Two types of word are summed under the following rules: $0 + x = x$, $1 + 1 = 2$, $1^* + 1^* = 1^*$, $1^* + 1 = 2^*$ and for other cases we can just sums the actual values (by holding $*$). In the same way, the rules for multiplication between the state vector and round matrix entry can be defined: $1 \cdot 1_F = 1$, $1^* \cdot 1_F = 1$,$2^* \cdot 1_F = 2$,$2 \cdot 1_F = 2$, $x \cdot 1 = x$,$x \cdot 0 = 0$ and $y \cdot 1_F = y$ for states $x \geq 0$ and $y \geq 3$ (with or without $*$).

For a given state, we use the matrix iteratively to obtain the new state over multiple rounds. To find the longest zero-correlation linear approximation we compute the new states in both forward and backward directions just before the values of all words become only 2 and $x > 2$ (with or without). Finally, we scan intermediate values and check the incoherence events.

For example for the encryption matrix for Feistel structure is $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 1_F \end{smallmatrix}\right)$. if we assume the initial mask type as $(1^*, 0)$ the mask type of the third round can be obtained as follows:

$$\big(1^*, 0\big) \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1_F \end{pmatrix} = \big(1^* \cdot 0 + 0 \cdot 1,\, 1^* \cdot 1 + 0 \cdot 1_F\big) = \big(0 + 0,\, 1^{+}0\big) = \big(0,\, 1^*\big)$$

$$\big(0,\, 1^*\big) \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1_F \end{pmatrix} = \big(0 \cdot 0 + 1^* \cdot 1,\, 0 \cdot 1 + 1^* \cdot 1_F\big) = \big(0 + 1^*,\, 0 + 1\big) = \big(1^*,\, 1\big)$$

$$\big(1^*, 1\big) \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1_F \end{pmatrix} = \big(1^* \cdot 0 + 1 \cdot 1,\, 1^* \cdot 1 + 1 \cdot 1_F\big) = \big(0 + 1\ ,\, 1^* + 1\big) = \big(1\ ,\, 2^*\big)$$

4.2 Zero-Correlation Linear Approximation for 14-rounds of Lblock

We applied Matrix method for Lblock. The Encryption and decryption matrices can be found in Appendix C. The longest zero-correlation linear approximation was obtained for 14 rounds of Lblock. If the input mask would be exactly one non zero nibble in L_r and the output mask after 14 rounds would be one non zero nibble in R_{r+14}, then the linear approximation has correlation zero. For example $(000a0000|00000000) \rightarrow (00000000|0b000000)$ has correlation exactly zero which the values a and b are non zero. The states of the rounds which can be found by using the encryption are depicted in Table 2. The contradiction occurs in $R_7(5)$. We note there exists 8×8 different classes of zero-correlation linear approximations for 14 rounds which each one includes $2^8 - 1$ different zero-correlation approximations. We will use this observation to reduce the data complexity as described in 3.

Table 2. Zero-correlation linear approximation for 14-round Lblock

Round	Γ_{L_r}	Γ_{R_r}
0	0001*0000	00000000
1	00000000	001*0000
2	01*000000	00010000
3	01000000	01*010000
4	01000001*	01010010
5	01001001	01010111*
6	010111*01	01111112
7	11111201	121212*13
7	11221132*	12011111
8	01111112	01110011*
9	110011*01	00101001
10	10100100	000011*00
11	0011*0000	00000100
12	00010000	0001*0000
13	01*000000	00000000
14	00000000	01*000000

5 Zero-Correlation Linear Cryptanalysis of 22 Reduced-Round Lblock

In this section, we propose a zero-correlation linear attack on 22-round LBlock. The attack utilizes the 14-round zero-correlation linear approximations described in Table 2 from round 5 to 18. After collecting sufficient plaintext-ciphertext pairs, we guess corresponding subkeys for the first four rounds and the last four rounds and estimate the correlation of approximations as described in Algorithm 1.

Based on the error probabilities α and β, the number of pairs N in Algorithm 1 and the decision threshold t are determined. The time complexity of the Algorithm 1 is $N \cdot 2^{28} \cdot 2^{28}$ where N is the number of plaintexts used in the cryptanalysis. So the time complexity is much more than exhaustive search. To overcome this restriction we note $L_4(4)$ and $R_{18}(6)$ are not affected by all bits in rounds $1-4$ and $19-22$. So Algorithm 1 is not optimal and it repeats the same procedure for different pairs. We show that it is possible to remove repetitions and reduce time complexity significantly.

The nibble $L_4(4)$ is affected by 32 bits of plaintext $L_0|R_0$, 20 bits of $L_1|R_1$, 12 bits of $L_2|R_2$ and 8 bits of $L_3|R_3$. Also $L_{18}(6)$ is affected by 32 bits of ciphertext $L_{22}|R_{22}$, 20 bits of $L_{21}|R_{21}$, 12 bits of $L_{20}|R_{20}$ and 8 bits of $L_{19}|R_{19}$. We call these bits "active" and other ones "neutral". The idea is to ignore neutral bits and instead of encrypting and decrypting all plaintex-ciphertext pairs, do it only once and count the number of pairs, which have the same value in active bits. In each step, for each subkey candidate, we encrypt (decrypt) active bits in round

Algorithm 1 Attack procedure

for all 2^{28} subkey nibbles $SK_1(4,2,1), SK_2(6,0), SK_3(7), SK_4(5)$ in rounds $1-4$ **do**
  for all 2^{28} subkey nibbles $SK_{19}(1), SK_{20}(2), SK_{21}(5,0), SK_{22}(7,6,0)$ in rounds $19-22$ **do**
    for all 2^8 possible values $i = 0, \cdots, 2^8-1$ **do**
      allocate the counter T_i and set them zero
    end for
    for all N plaintext-ciphertext pairs **do**
      encrypt plaintext to obtain the nibble $L_4(4)$;
      decrypt ciphertext to obtain the nibble $R_{18}(6)$;
      for correponding $i = (L_4(4)|R_{18}(6))$ increase the counter T_i by one.
    end for
    compute the statistic value $T = N \cdot 2^8 \sum_{i=0}^{2^8-1}(\frac{T_i}{N} - \frac{1}{2^8})$
    If $T < t$, then the guess key is a possible candidate.
  end for
end for
Do exhaustive search for all keys which corresponds to the guess subkey bits

r over one round and count the number of pairs which give the same value in active bits in round $r+1$ $(r-1)$.

The attack procedure is as follows:

1. Collect N plaintexts with corresponding ciphertexs.
2. Allocate a 8-bit counter $N_0[x_0, x_{22}]$ for each of 2^{64} possible values of $(x_0|x_{22})$ which $x_0 = L_0(5,4,2,1,0)|R_0(6,4,1)$ and $x_{22} = L_{22}(7,6,4,2,0)|R_{22}(7,5,2)$ and set them zero. Calculate the number of pairs of plaintext-ciphertext with given values x_0 and x_{22} and save it in $N_0[x_0, x_{22}]$. In this step, around 2^{64} plaintext-ciphertext pairs are divided into 2^{64} different state. The expected pairs for each state is around one. So the assumption N_0 as a 8-bit counter is sufficient.
3. Guess the 3 nibbles $SK_1(4,2,1)$. Allocate a counter $N_1[x_1, x_{22}]$ for each of 2^{52} possible values of $(x_1|x_{22})$ which $x_1 = L_1(6,3,0)|R_1(5,0)$ and set them zero. For all 2^{32} possible values of x_0, encrypt x_0 one round to obtain x_1 and update the value $N_1[x_1, x_{22}] = N_1[x_1, x_{22}] + N_0[x_0, x_{22}]$ for all 2^{32} values of x_{22}
4. Guess 2 nibbles $SK_2(6,0)$. Allocate a counter $N_2[x_2, x_{22}]$ for each of 2^{44} possible values of $(x_2|x_{22})$ which $x_2 = L_2(7,2)|R_2(3)$ and set them zero. For all 2^{20} possible values of x_1, encrypt x_1 one round to obtain x_2 and update the value $N_2[x_2, x_{22}] = N_2[x_2, x_{22}] + N_1[x_1, x_{22}]$ for all 2^{32} values of x_{22}.
5. Guess the nibble $SK_3(7)$. Allocate a counter $N_3[x_3, x_{22}]$ for each of 2^{40} possible values of $(x_3|x_{22})$ which $x_3 = L_3(5)|R_3(2)$ and set them zero. For all 2^{12} possible values of x_2, encrypt x_2 one round to obtain x_3 and update the value $N_3[x_3, x_{22}] = N_3[x_3, x_{22}] + N_2[x_2, x_{22}]$ for all 2^{32} values of x_{22}.
6. Guess the nibble $SK_4(5)$. Allocate a counter $N_4[x_4, x_{22}]$ for each of 2^{36} possible values of $(x_4|x_{22})$ which $x_4 = L_4(4)$ and set them zero. For all 2^8

possible values of x_3, encrypt x_3 one round to obtain x_4 and update the value $N_4[x_4, x_{22}] = N_3[x_4, x_{22}] + N_3[x_3, x_{22}]$ for all 2^{32} values of x_{22}.

7. Guess the 3 nibbles $SK_{22}(7, 6, 0)$. Allocate a counter $N_5[x_4, x_{21}]$ for each of 2^{24} possible values of $(x_4|x_{21})$ which $x_{21} = L_{21}(4, 2)|R_{21}(5, 3, 0)$ and set them zero. For all 2^{32} possible values of x_{22}, decrypt x_{22} one round to obtain x_{21} and update the value $N_5[x_4, x_{21}] = N_5[x_4, x_{21}] + N_4[x_4, x_{22}]$ for all 2^4 values of x_4.
8. Guess 2 nibbles $SK_{21}(5, 0)$. Allocate a counter $N_6[x_4, x_{20}]$ for each of 2^{16} possible values of $(x_4|x_{20})$ which $x_{20} = L_{20}(3)|R_{20}(2, 0)$ and set them zero. For all 2^{20} possible values of x_{21}, decrypt x_{21} one round to obtain x_{20} and update the value $N_6[x_4, x_{20}] = N_6[x_4, x_{20}] + N_5[x_4, x_{21}]$ for all 2^4 values of x_4.
9. Guess the nibble $SK_{20}(2)$. Allocate a counter $N_7[x_4, x_{19}]$ for each of 2^{12} possible values of $(x_4|x_{19})$ which $x_{19} = L_{19}(0)|R_{19}(1)$ and set them zero. For all 2^{12} possible values of x_{20} decrypt x_{20} one round to obtain x_{19} and update the value $N_7[x_4, x_{19}] = N_7[x_4, x_{19}] + N_6[x_4, x_{20}]$ for all 2^4 values of x_4.
10. Guess the nibble $SK_{19}(1)$. Allocate a counter $N_8[x_4, x_{18}]$ for each of 2^8 possible values of $(x_4|x_{18})$ which $x_{18} = R_{18}(6)$ and set them zero. For all 2^8 possible values of x_{19}, decrypt x_{19} one round to obtain x_{18} and update the value $N_8[x_4, x_{18}] = N_8[x_4, x_{18}] + N_7[x_4, x_{19}]$ for all 2^4 values of x_4.
11. Compute the statistic value $T = N \cdot 2^8 \sum_{x_4=0}^{2^4-1} \sum_{x_{18}=0}^{2^4-1} (\frac{N_8[x_4, x_{18}]}{N} - \frac{1}{2^8})$. If $T < t$, then the guess key is a possible candidate.
12. Do exhaustive search for all keys which corresponds to the guess subkey bits.

Attack complexity

The memory complexity of the attack is dominated by step 2 which needs 2^{64} bytes.

Time complexity of step 1 and 2 is equal to the number of needed plaintext-ciphertext pairs N.

Step 3 requires $2^{12} \times 2^{32} \times 2^{32} = 2^{76}$ memory accesses, because we should guess 12 bits for SK_1, and for 2^{32} values encrypt x_0 one round and then update N_1 for 2^{32} times.

Step 4 requires $2^{12} \times 2^8 \times 2^{20} \times 2^{32} = 2^{72}$ memory accesses, because for all of guessed 2^{12} keys in previous step, we should guess 8 bits for SK_2, and for 2^{20} values encrypt x_1 one round and then update N_2 for 2^{32} times.

Step 5 requires $2^{20} \times 2^4 \times 2^{12} \times 2^{32} = 2^{68}$ memory accesses, because for all of guessed 2^{20} keys in previous steps, we should guess 4 bits for SK_3 and for 2^{12} values encrypt x_2 one round and then update N_3 for 2^{32} times.

Step 6 requires $2^{24} \times 2^4 \times 2^8 \times 2^{32} = 2^{68}$ memory accesses, because for all of guessed 2^{24} keys in previous steps, we should guess 4 bits for SK_4 and for 2^8 values encrypt x_3 one round and then update N_4 for 2^{32} times.

Step 7 requires $2^{28} \times 2^{12} \times 2^{32} \times 2^4 = 2^{72}$ memory accesses, because for all of guessed 2^{28} keys in previous steps, we should guess 12 bits for SK_{22} and for 2^{32} values decrypt x_{22} one round and then update N_5 for 2^4 times.

Step 8 requires $2^{40} \times 2^8 \times 2^{20} \times 2^4 = 2^{72}$ memory accesses, because for all of guessed 2^{40} keys in previous steps, we should guess 8 bits for SK_{21} and for 2^{20} values decrypt x_{21} one round and then update N_6 for 2^4 times.

Step 9 requires $2^{48} \times 2^4 \times 2^{12} \times 2^4 = 2^{68}$ memory accesses, because for all of guessed 2^{48} keys in previous steps, we should guess 4 bits for SK_{20} and for 2^{12} values decrypt x_{20} one round and then update N_7 for 2^4 times.

Step 10 requires $2^{52} \times 2^4 \times 2^8 \times 2^4 = 2^{68}$ memory accesses, because for all of guessed 2^{52} keys in previous steps, we should guess 2^4 for SK_{19} and for 2^8 values decrypt x_{19} one round and then update N_8 for 2^4 times.

Step 11 requires $2^{56} \times 2^8 = 2^{64}$ memory accesses, because for all of guessed 2^{56} keys in previous steps, we should read 2^8 values of $N_8[x_4, x_{18}]$.

So step 12 requires $2^{80} \cdot \beta$ full encryption because we expect a wrong subkey survives with probability β.

The time complexity is dominated by step 3 and step 12. The time complexity of round 3 is 2^{76} memory accesses. If we consider one memory accesses as a half round, the time complexity of step 3 is $2^{76} \times \frac{1}{2} \times \frac{1}{22} = 2^{70.54}$ of 22-round Lblock. Based on the error probability type I α and error probability type II β, the number of plaintexts-ciphetexts pairs needed, time complexity of step 12 and success probability are determined.

There is a trade-off between the time complexity and the data complexity of the attack, as depicted in Table 1. To reduce the time complexity as much as possible, we assume to have access the full codebook. In this case, the error probabilities and time complexity of step 12 is negligible compared to the complexity of step 3. To have a lowest data complexity, we can set $\alpha = 2^{-2.7}$ and $\beta = 2^{-1}$. In this case data complexity decreases to $N = 2^{60}$ in cost of increasing time complexity. The time complexity is dominated by step 12 which needs $2^{80} \cdot 2^{-1} = 2^{79}$ 22-round Lblock encryption. The more realistic assumption is the state between these cases. For example, if we set $\alpha = 2^{-2.7}$ and $\beta = 2^{-10}$ then $z_{1-\alpha} = 1$ and $z_{1-\beta} = 3.09$. Equation (2) determines the data complexity $N = 2^{62.1}$. The time complexity is dominated by step 3 and 12 $2^{70.5} + 2^{70} = 2^{71.27}$. The success probability is $1 - \alpha = 0.84$.

6 Conclusion

In this paper we showed how to use Matrix method to establish zero-correlation linear approximations automatically. We used this method to obtain several zero-correlation linear approximation for the 14 rounds of Lblock. We also hope described method will be useful for further research in other block ciphers. Based on the 14 rounds distinguisher we present an attack on the 22 rounds of Lblock. While the previous attack used chosen plaintexts, our attack model is distinct known palaintexts which is a more realistic model. The overall complexity is improved by a factor of 2^8.

Acknowledgements

The author would like to thank Kaisa Nyber and Céline Blondeau for very helpful comments and suggestions.

References

1. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Diffrentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 1223. Springer, Heidelberg (1999)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 221. Springer, Heidelberg (1991)
3. Biham, E.: On Matsui's Linear Cryptanalysis. In: Santis, A,D. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 341-355. Springer, Heidelberg (1994)
4. Biryukov, A., De Canni'ere, C., Quisquater, M.: On multiple linear approximations. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 122. Springer, Heidelberg (2004)
5. Blondeau, C., Gérard, B.: Multiple Differential Cryptanalysis: Theory and Practice. In Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 35-54. Springer, Heidelberg (2011)
6. Bogdanov, A., Rijmen, V.: Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers. Accepted to Designs, Codes and Cryptography, in press, Springer-Verlag, 2012.
7. Bogdanov, A., Wang, M.: Zero Correlation Linear Cryptanalysis with Reduced Data Complexity. In Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 29-48, Springer, Heidelberg (2012)
8. Bogdanov, A., Leander, G., Nyberg, K., Wang, M.: Integral and Multidimensional Linear Distinguishers with Correlation Zero. ASIACRYPT 2012. LNCS, Wang, X., Sako, K. (eds.), Springer, Heidelberg (2012)
9. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional Extension of Matsuis Algorithm 2. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 209227. Springer, Heidelberg (2009)
10. Kaliski, B., Robshaw, M.: Linear cryptanalysis using multiple approximations. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 2639. Springer, Heidelberg (1994)
11. Kelsey, J., Kohno, T., Schneier, B.: Amplied Boomerang Attacks Against Reduced-Round MARS and Serpent. In Schneier, B. (ed) FSE 2001. LNCS, vol. 1978, pp. 7593, Springer, Heidelberg (2002)
12. Karakoç, F., Demirci, H., Harmanci, A.: Impossible Differential Cryptanalysis of Reduced-Round LBlock. In: Askoxylakis, I., Pöhls, HC., Posegga, J. (eds.) WISTP 2012. LNCS, vol. 7322, pp. 179188. Springer, Heidelberg (2012)
13. Kim, J., Hong, S., Lim, J.: Impossible differential cryptanalysis using matrix method. Discrete Mathematics 310(5), 9881002 (2010)
14. Kim, J., Hong, S., Sung, J., Lee, J., Lee, S.: Impossible Differential Cryptanalysis for Block Cipher Structures. In Johansson,T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 82-96. Springer, Heidelberg (2003)
15. Knudsen, L. R.: Truncated and higher order differential. In Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196211. Springer, Heidelberg (1995)

16. Liu, Y., Gu, D., Liu, Z., Li, W.: Impossible Differential Attacks on Reduced-Round LBlock. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 97-108. Springer, Heidelberg (2012)
17. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386397. Springer, Heidelberg (1994)
18. Nyberg, K.: Linear Approximation of Block Ciphers. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 439444. Springer, Heidelberg (1995)
19. Wagner, D.: The boomerang Attack. In: Knudsen, L. (ed.) FSE 1999. LNCS, vol. 1636, pp.156-170. Springer, Heidelberg (1999)
20. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327344. Springer, Heidelberg (2011)

A Lblock S-boxes

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_0	14	9	15	0	13	4	10	11	1	2	8	3	7	6	12	5
S_1	4	11	14	9	15	13	0	10	7	12	5	6	2	8	1	3
S_2	1	14	7	12	15	13	0	6	11	5	9	3	2	4	8	10
S_3	7	6	8	11	0	15	3	14	9	10	12	13	5	2	4	1
S_4	14	5	15	0	7	2	12	13	1	8	4	9	11	10	6	3
S_5	2	13	11	12	15	14	0	9	7	10	6	3	1	8	4	5
S_6	11	9	4	14	0	15	10	13	6	12	5	7	3	8	1	2
S_7	13	10	15	0	14	4	9	11	2	1	8	3	7	5	12	6
S_8	8	7	14	5	15	13	0	6	11	12	9	10	2	4	1	3
S_9	11	5	15	0	7	2	9	13	4	8	1	12	14	10	3	6

B Lemmas

Lemma 1. ***XOR operation:*** *Let $f(x_1, x_2) = x_1 \oplus x_2$ then the correlation of linear approximation $u_1 \cdot x_1 + u_2 \cdot x_2 = v \cdot f(x_1, x_2)$ is non-zero if and only if $u_1 = u_2 = v$.*

Lemma 2. ***Branching operation:*** *Let $f(x) = (x, x)$ then the correlation of linear approximation $u_1 \cdot x + u_2 \cdot x = v \cdot f(x)$ is non-zero if and only if $u = v_1 + v_2$.*

Lemma 3. ***Bijective Function:*** *Let $f(x)$ be a bijective function then the correlation of linear approximation $u \cdot x = v \cdot f(x)$ is non-zero if and only if $u = v = 0$ or $u \neq 0$ and $v \neq 0$.*

C Lblock Matrices

$$M_{Encryption} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F
\end{pmatrix}$$

$$M_{Decryption} = \begin{pmatrix}
0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1_F & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1_F & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$