The ISC Int'l Journal of Information Security
ISeCure
January 2016, Volume 8, Number 1 (pp. 73–84)
http://www.isecure-journal.org

A New Method for Accelerating Impossible Differential Cryptanalysis and its Application on LBlock☆

Akram Khalesi [1], Hossein Bahramgiri [1,2,*], and Davod Mansuri [2]
[1] *Department of Information and Communication Technology, Malek-e-Ashtar University of Technology, Tehran, Iran*
[2] *Institute of Research on Information and Communication Security (IRICS), Malek-e-Ashtar University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 22 September 2015
First Revised: 21 November 2015
Last Revised: 15 December 2015
Accepted: 22 December 2015
Published Online: 4 January 2016

Keywords:
Differential Cryptanalysis, Impossible Differential Cryptanalysis; LBlock.

ABSTRACT

Impossible differential cryptanalysis, an extension of the differential cryptanalysis, is one of the most efficient attacks against block ciphers. This cryptanalysis method has been applied to most of the block ciphers, and has shown significant results. Using structures, key schedule considerations, early abort, and pre-computation are some common methods to reduce complexities of this attack. In this paper, we present a new method for decreasing the time complexity of impossible differential cryptanalysis through breaking down the target key space into subspaces, and extending the results on subspaces to the main target key space. The main advantage of this method is that there is no need to consider the effects of changes in the values of independent key bits on each other. Using the 14-round impossible differential characteristic observed by Boura *et al.* at ASIACRYPT 2014, we implement this method on 23-round LBlock and demonstrate that it can reduce the time complexity of the previous attacks to $2^{71.8}$ 23-round encryptions using 2^{59} chosen plaintexts and 2^{73} blocks of memory.

1 Introduction

1.1 Motivation, Contribution and Organization

Cryptanalysis methods that rely on eliminating the round's function key's effect by working on the differences, form a considerable part of attacks against block ciphers. Differential cryptanalysis [1] proposed by Biham *et al.* is the first attack of this category. Impossible differential, higher order differential, truncated differential, and boomerang cryptanalyses are other attacks of this type.

Impossible differential cryptanalysis was introduced independently by Knudsen on DEAL block cipher [7] and Biham *et al.* on Skipjack block cipher [2]. It is one of the conventional cryptanalyses methods for block ciphers showing remarkable results. Distinguisher of impossible differential attack, impossible differential characteristic, is n-round of target algorithm with specific input and output differences holding with probability zero, implying that a pair with such input difference cannot lead to the specified output difference after n rounds. After extending the characteristic by adding some rounds to the plaintext and ciphertext sides of the distinguisher, we try to find the key values leading chosen pairs of plaintexts to the attack's

☆ This article is an extended version of an ISCISC'2014 paper.
* Corresponding author.

Email addresses: a_khalesi@mut.ac.ir (A. Khalesi), bahramgiri@mut.ac.ir (H. Bahramgiri), davodmansuri@mut.ac.ir (D. Mansuri)

ISSN: 2008-2045

distinguisher. Since these key values lead to the distinguisher holding with probability zero; they cannot be the correct key and should be eliminated from the target key space.

LBlock is a lightweight block cipher of generalized Feistel type represented in ACNS2011 [3]. The algorithm consists of 32 rounds with 64-bit block length, and 80-bit key length. In [3], the designers evaluated LBlock's security against different attacks including impossible differential cryptanalysis. Using a 14-round impossible differential characteristic They presented the attack on 20 rounds of the algorithm. Using another 14-round characteristic, impossible differential cryptanalysis on 21 rounds of the algorithm is presented in [4]. This improvement was achieved by applying mentioned methods, such as key schedule consideration. The authors in [5] represented another impossible differential cryptanalysis of LBlock. They have separated the plaintext and ciphertext sides added to the distinguisher and searched each part independently. Using this method, they could reduce the complexity on 21 rounds of algorithm. Also, they have extended the attack to 22 rounds. Our new method can be considered as an extension of their work. By investigating the key schedule of the cipher in [9], the authors have extended impossible differential cryptanalysis to 23 rounds of the algorithm. In [10], the authors presented another impossible differential cryptanalysis on 23 rounds of the algorithm by using an approach that can help reducing the number of pairs used in the attack.

In conventional impossible differential attacks, the effects of changes in most of the target key bits are considered while the attackers are studying the possible values for some specific key bits, although lots of them are independent; and these considerations come to attacks with higher time complexities. The main idea of the new method is preventing this overload by separating the target key space into subspaces. We classify the target key bits into the groups and determine the values of key bits in each group independently. Then, we extend the achieved results to the whole target key space. This idea was first applied on 22-round LBlock in [21]; however, there was no efficient algorithm for extending the achieved results to the main target key space. In this paper, we generalize the method and propose an efficient algorithm for combining the results stored in distinct tables.

Using this improved method in parallel with previous ones, an improved attack on 23-round LBlock is presented in this paper. The results of this attack are compared with the previous works in Table 1. The outstanding advantages of this method are its generality that can be applied on different block ciphers (especially those with weak diffusion layers including lightweight algorithms) and also applicability in parallel with other techniques proposed for improving impossible differential cryptanalysis. The rest of this paper is organized as follows: In Section 2, we introduce the impossible differential cryptanalysis and describe the new method. Prevalent notations for LBlock and brief description of it are presented in Section 3. In Section 4, we clarify the new method by applying it on LBlock. Finally, we conclude the paper in Section 5.

Table 1. Comparison of attacks on LBlock

No. Rounds	Cryptanalysis method	Memory complexity	Data complexity	Time complexity	Ref.
18	Integral	2^{16}	2^{62}	2^{12}	[17]
20	Integral	2^{35}	$2^{63.6}$	$2^{39.6}$	[17]
20	Impossible Differential	2^{60}	2^{63}	$2^{72.7}$	[3]
21	Impossible Differential	2^{68}	$2^{62.5}$	$2^{73.7}$	[4]
21	Impossible Differential	2^{68}	2^{63}	$2^{69.5}$	[5]
22	Impossible Differential	$2^{72.67}$	2^{58}	$2^{79.28}$	[5]
22	Zero-Correlation Linear	2^{64}	2^{62}	$2^{71.27}$	[14]
22	Integral	2^{63}	2^{61}	2^{70}	[16]
22	Related Key Impossible Differential	not mentioned	2^{47}	2^{70}	[6]
23	Related Key Impossible Differential	$2^{61.4}$	$2^{61.4}$	$2^{78.3}$	[8]
23	Impossible Differential	not mentioned	2^{57}	$2^{77.4}$	[9]
23	Impossible Differential	$2^{74.6}$	2^{59}	$2^{75.36}$	[10, 12, 13]
23	Impossible Differential	2^{73}	2^{59}	$2^{71.8}$	This Paper
24	Key Difference Invariant Bias (Related-Key)	2^{61}	$2^{62.95}$	$2^{70.67}$	[19]
Full	Higher Order Key Partitioning	Negligible	2	$2^{78.338}$	[20]
Full	Biclique	Negligible	$2^{78.4}$	2^{52}	[18]

2 Impossible Differential Cryptanalysis and the New Method

Consider the typical round function in block ciphers of Feistel type, consisted of add round key by XOR op-

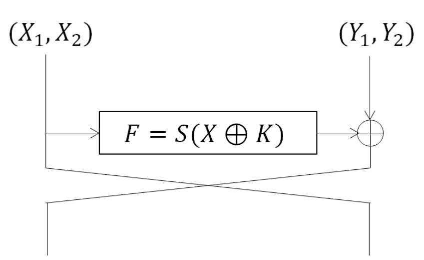

Figure 1. typical round function in algorithms of Feistel ciphers

eration, and applying S-Boxes, as depicted in Figure 1. The terms used in this paper are as follows:

Definition 1 For a pair (X_1, Y_1) and (X_2, Y_2), *Cancellation* occurs when $X_1 \neq X_2$ and $Y_1 \neq Y_2$ and $[F(X_1, K) \oplus F(X_2, K)] = [S(X_1 \oplus K) \oplus S(X_2 \oplus K)] = (Y_1 \oplus Y_2)$.

Cancellations on specific locations in the target algorithm are the *conditions* of the impossible differential attack. Key values satisfying the attack's conditions result the attack's characteristic from the chosen pairs. These values for the key are those we are looking for during the attack procedure.

Definition 2 For each cancellation, corresponding to one F-function, we form a group made up of the F-function's subkey, and also the key bits of the other rounds which affect the input pair value or the desired differential output value of that F-function (X_1,X_2 and $Y_1 \oplus Y_2$ in definition 1). The number of key groups in the attack equals the number of cancellations.

Definition 3 *Involved-key-bits* is the set of key bits presented in groups.

Definition 4 *Information-key-bits* is a subset of involved-key-bits of the smallest size that can specify the involved-key-bits value uniquely.

Due to the redundancy of the key schedule, it is probable to have relations between different rounds' subkeys in the set of involved-key-bits. As a result, size of involved-key-bits is less than or equal to the size of information-key-bits. There is a well-known property for invertible S-boxes that we apply it in our attack implementation:

Property 1 For F-function $F(X, K) = S(X \oplus K)$, where S is a given invertible S-Box, for an input pair (X_1, K) and (X_2, K); if we know the value of (X_1, X_2), we can determine the values of K leading to a specified differential value for $[S(X_1 \oplus K) \oplus S(X_1 \oplus K)] = \Delta Y$.

2.1 Impossible Differential Cryptanalysis

The characteristic in impossible differential attack consists of n rounds of the target algorithm with $\Delta\alpha$

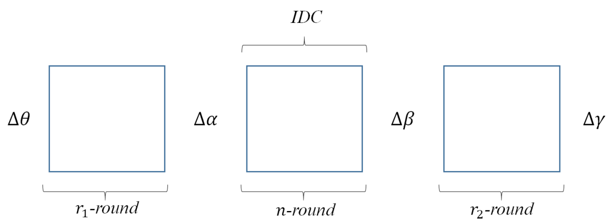

Figure 2. impossible differential cryptanalysis

and $\Delta\beta$ as the differential input and output of this n-round characteristic, respectively (Figure 2). $\Delta\alpha$ (resp. $\Delta\beta$) contains m_1 (resp. m_1) non-zero difference sub-blocks (mostly $m_1 = m_2 = 1$). This characteristic states that a pair of texts with $\Delta\alpha$ as the differential input to the round i cannot lead to $\Delta\beta$ as the differential output of the round $i + n - 1$.

Adding r_1 rounds to the plaintext side and r_2 rounds to the ciphertext side of the characteristic, impossible differential attack can be implemented on $n + r_1 + r_2$ rounds of the algorithm, as Figure 2. After this extension, differential input (resp. output) of the $n+r_1+r_2$-round version of the algorithm equals $\Delta\theta$ (resp. $\Delta\gamma$) containing $l_1 + m_1$ (resp. $l_2 + m_2$) non-zero difference sub-blocks. The number of conditions in this attack is $l_1 + l_2$. In this way, the attack is based on finding key values that cancel l_1 (resp. l_2) number of these non-zero difference sub-blocks on specific locations from the plaintext side (resp. ciphertext side), leading to the attack's characteristic. Since the impossible differential characteristic comes to a contradiction, none of the key values leading to this characteristic can be the correct key.

For finding the incorrect key values, we take some pairs with the specified differential forms in the plaintext and ciphertext sides; $\Delta\theta$ and $\Delta\gamma$, respectively. Then, we test the possible key values on them to find those which lead to the impossible differential characteristic, i.e. incorrect keys. The number of pairs needed for recovering the correct key depends on the number of cancellations that determine the number of condition-bits (the bits that there are conditions on them). For $l_1 + l_2$ cancellations at t-bit F-functions, the number of condition-bits equals $(l_1 + l_2)t$. If the number of information-key-bits in the attack is s, each pair can discard $2^s(2^{-(l_1+l_2)t})$ key values on average. So, if the attacker wants to eliminate all the incorrect keys, the number of required pairs, q, is determined by the following inequality:

$$2^s(1 - 2^{-(l_1+l_2)t})^q \leq 1$$

Time and memory complexities of impossible differential cryptanalysis are dependent on the attack procedure. In [11] a lower bound for time complexity of impossible differential cryptanalysis is provided. The authors prove that an impossible differential attack, which gives 2^{ε} values for the L-bit involved keys, contains at least $(L-\varepsilon+ln(2))\times 2^{L}$ memory accesses.

In conventional impossible differential cryptanalysis, there are some common methods to reduce the attack's complexity; such as using structures, key schedule considerations, early abort, and pre-computation. Using structures leads attackers to choose texts of specific form in which there are constant values in some positions of the texts' block and the other positions can take all possible values. By this method, the chosen pairs from one structure have the desired differential form. Key schedule consideration is one of the most effective methods to reduce the complexity of attacks. The key schedule generates rounds' subkeys from one seed, main key, usually in an iterative form; and therefore, it contains some redundancy. In ideal key schedules, this redundancy is not exploitable; but for most of the algorithms, the attackers can find some applicable redundancy to reduce size of the target key space. In early abort technique, attackers discard the pairs or key values that cannot comply with some conditions as soon as possible. Here, pre-computation means performing the repetitive computations that are independent from the achieved results of the attack's steps and classifying them in tables before the attack starts. Then, attackers can refer to these tables and find the result of the intended computation when they need, instead of spending time on computing in the online phase of the attack.

2.2 The New Method

In impossible differential attack, the attackers are searching for the values of the information-key-bits satisfying the attack's conditions to discard them from the target key space. Not necessarily all the information-key-bits have effect on all the conditions. However, in conventional impossible differential cryptanalysis methods, the effects of changes in the values of all the information-key-bits are considered while attackers are studying a specific cancellation. Here, we present a new method which is based on reducing computational complexity by refusing to study the effects of independent key bits' changes on the attack's conditions.

In our attack scenario, for each one of l_1+l_2 cancellations, we determine the effective key bits and put them in distinct groups. Then, for each group, we form a table consisted of flags corresponding to all the possible values of the determined key bits. For each chosen pair, we find the key values satisfying each cancellation and set the achieved values' flags in the corresponding table. The combinations of key values that their flags are set from the tables lead the chosen pair to the attack's distinguisher and should be eliminated from the target key space.

According to property 1, we just need to know the input pair value of (X_1, X_2) and the desired output difference of the F-function $F = S(X \oplus K)$ for each cancellation to determine the key values of the function; and just the changes in the key bits that affect the input pair value or the desired differential output value of the F-function are needed to be considered, not the changes in all the involved-key-bits. Therefore, there are two types of key bits in each group:

Type-1: key bits of the F-function on which we have cancellation,

Type-2: key bits which affect the input pair value or the desired differential output value of this function

(Groups of key bits corresponding to cancellations on the first and last rounds do not contain key bits of type-2.)

For each group of key bits that contains key bits of type-2, we guess this type of key bits to determine the input pair value and the desired differential output value of the F-function. Then, we can determine the values of type-1 key bits in each group.

Impossible differential cryptanalysis by using the new method can be done within the following steps:

(1) Choosing a proper n-round impossible differential characteristic,
(2) Extending number of rounds to $n+r_1+r_2$,
(3) Determining the non-zero difference sub-blocks of the plaintext and ciphertext sides,
(4) Specifying number of cancellations l_1+l_2 and their locations,
(5) Forming l_1+l_2 key groups corresponding to l_1+l_2 cancellations and determining each group's key bits,
(6) Forming l_1+l_2 tables corresponding to l_1+l_2 key groups consisted of flags corresponding to the possible values of the determined key bits for each group,
(7) Choosing pairs with the specified difference in the plaintext and ciphertext sides, and repeating the two following steps for each pair,
(8) Determining key bits values of each groups and setting their corresponding values' flags in the corresponding table,
(9) Eliminating the combinations of key bits values of tables that their flags are set from the target key space.

We clarify this method in Section 4 by applying it on 23-round LBlock. To the best of our knowledge, impossible differential attack by this method comes to the best results in terms of data and time complexities, considering the cryptanalysis methods in single-key model excluding biclique attack.

3 Brief Description of LBlock

The used notation in this paper is summarized here:

A: a bit string

$A|B$: concatenation of strings A and B

$A <<< B$: left rotation of A by j bits

K_i^r: the i^{th} nibble of the r^{th} round key

R^{r-1}: the right half of the r^{th} round's input

L^{r-1}: the left half of the r^{th} round's input

R_i^{r-1}: the i^{th} nibble of R^{r-1}

L_i^{r-1}: the i^{th} nibble of L^{r-1}

ΔR^{r-1}: the difference of two R^{r-1}

ΔL^{r-1}: the difference of two L^{r-1}

ΔR_i^{r-1}: the difference of two R_i^{r-1}

ΔL_i^{r-1}: the difference of two L_i^{r-1}

S_i^r : output of the i^{th} S-Box in the r^{th} round

ΔS_i^r: the difference of two S_i^r

(Note that the nibbles are indexed beginning with zero subscript and ending with seven, and the zero-indexed nibble is the rightmost one)

LBlock is a lightweight block cipher with 64-bit block length and 80-bit key length [3]. Its structure is a generalization of a Feistel structure with 32 rounds. It can be implemented efficiently both in hardware and software. The iterative round function F consists of three basic functions: (1) add round key by XOR, (2) confusion function by S-Boxes, and (3) diffusion function by permutation and rotation. LBlock's algorithm uses 4-bit S-Boxes $S_i, i = 0, 1, \ldots, 9$; where $S_i, i = 0, 1, \ldots, 7$ are used in the round functions and S-Boxes S_8 and S_9 are used in the key schedule. Its round function is depicted in Figure 3.

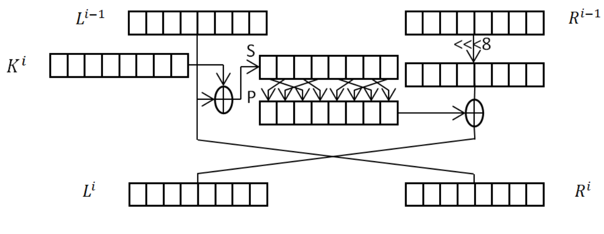

Figure 3. Round function of LBlock.

4 Improved Impossible Differential Cryptanalysis on LBlock

S-Boxes used in an algorithm are so effective in the security against cryptanalyses. After studying differential distribution tables of the S-Boxes of LBlock, $S_i, i = 0, 1, \ldots, 7$ we found a property common among them:

Property 2 For the SBoxes used in the LBlock's algorithm, the probability of generating a specific output difference from an input difference is $2^{(-1.4)}$; and for each possible input pair and output difference, there are on average $2^{1.4}$ key value.

4.1 Impossible Differential characteristic

We use the 14-round impossible differential characteristic as our attack's distinguisher [10]. This characteristic is as follows (see Figure 5 in the Appendix):

$(0000, 0000, 0000, a000) \longrightarrow (0000, 0b00, 0000, 0000)$

Where $a, b \in \{0, 1\}^4 \backslash \{0\}^4$ are two non-zero nibbles. We add five rounds to the plaintext side and four rounds to the ciphertext side of the characteristic and propose our attack on 23-round LBlock (Figure 4).

4.2 Separating Target Key Space into Subspaces

There are eighteen cancellations during the attack; and therefore, we separate target key bits into eighteen groups. The groups $g_i, i = 1, 2, 3, \ldots, 18$ and their corresponding conditions are demonstrated in Table 2. We find key values of each group, committing the corresponding cancellation, and discard the combinations of these keys values from the target key space.

4.3 Pre-computation

For $S_i, i = 0, 1, \ldots, 7$ we form the tables $A_i, i = 0, 1, \ldots, 7$ indexed by $(x_1, x_2, \Delta y)$. For all possible values of $(x_1, x_2, \Delta y)$, we find key values which result specific values for $\Delta y = [S_i(x_1 \oplus k) \oplus S_i(x_2 \oplus k)], i = 0, 1, \ldots, 7$; and put them in the corresponding rows of $A_i, i = 0, 1, \ldots, 7$. Also, for 18 groups of key bits, we form tables $U_i, i = 1, 2, 3, \ldots, 18$ consisted of flags corresponding to the possible values for the key bits in each group.

4.4 Key Recovery

The key recovery procedure is organized as the following steps:

Step 1 (Choosing plaintexts). Take 2^n structures of the form $(b_1|a_1|b_2|a_2)|(b_3|b_4|b_5|a_3)|(b_6|b_7|b_8|b_9)|(b_{10}|b_{11}|b_{12}|a_4))$ where $a_i, i = 1, 2, 3, 4$, are 4-bit fixed constants, and each $b_i, i = 1, 2, 3, \ldots, 12$ takes all 2^4 possible values for a nibble. Almost all the chosen pairs of each structure have differential form of $((*|0|*|0)|(*|*|*0)|(*|*|*|*)|(*|*|*0))$, where "$*$" presents a non-zero 4-bit value. Each struc-

Table 2. Conditions of the Cancellations and Corresponding Groups of Key Bits.

NO.	Condition	Corresponding group of key bits
1	$S_1(L_1^0 \oplus K_1^1) \oplus S_1(L'^0_1 \oplus K_1^1) = \Delta R_6^0$	$g_1 = \{K_1^1\}$
2	$S_3(L_3^0 \oplus K_3^1) \oplus S_3(L'^0_3 \oplus K_3^1) = \Delta R_7^0$	$g_2 = \{K_3^1\}$
3	$S_2(L_2^0 \oplus K_2^1) \oplus S_2(L'^0_2 \oplus K_2^1) = \Delta R_1^0$	$g_3 = \{K_2^1\}$
4	$S_5(L_5^0 \oplus K_5^1) \oplus S_5(L'^0_5 \oplus K_5^1) = \Delta R_2^0$	$g_4 = \{K_5^1\}$
5	$S_5(L_5^1 \oplus K_5^2) \oplus S_5(L'^1_5 \oplus K_5^2) = \Delta R_2^1$	$g_5 = \{K_7^1, K_5^2\}$
6	$S_7(L_7^1 \oplus K_7^2) \oplus S_7(L'^1_7 \oplus K_7^2) = \Delta R_3^1$	$g_6 = \{K_6^1, K_7^2\}$
7	$S_6(L_6^1 \oplus K_6^2) \oplus S_6(L'^1_6 \oplus K_7^2) = \Delta R_5^1$	$g_7 = \{K_4^1, K_6^2\}$
8	$S_1(L_1^2 \oplus K_1^3) \oplus S_1(L'^2_1 \oplus K_1^3) = \Delta R_6^2$	$g_8 = \{K_2^1, K_3^2, K_1^3\}$
9	$S_3(L_3^2 \oplus K_3^3) \oplus S_3(L'^1_3 \oplus K_3^3) = \Delta R_7^2$	$g_9 = \{K_0^1, K_2^2, K_3^3\}$
10	$S_7(L_7^3 \oplus K_7^4) \oplus S_7(L'^3_7 \oplus K_7^4) = \Delta R_3^3$	$g_{10} = \{K_7^1, K_5^1, K_4^2, K_6^3, K_7^4\}$
11	$S_3(L_3^4 \oplus K_3^5) \oplus S_3(L'^4_3 \oplus K_3^5) = \Delta R_7^4$	$g_{11} = \{K_1^1, K_2^1, K_3^1, K_7^1, K_1^2, K_3^2, K_0^3, K_2^4, K_3^5\}$
12	$S_0(L_0^{22} \oplus K_0^{23}) \oplus S_0(L'^{22}_0 \oplus K_0^{23}) = \Delta L_2^{23}$	$g_{12} = \{K_0^{23}\}$
13	$S_5(L_5^{22} \oplus K_5^{23}) \oplus S_5(L'^{22}_5 \oplus K_5^{23}) = \Delta L_4^{23}$	$g_{13} = \{K_5^{23}\}$
14	$S_6(L_6^{22} \oplus K_6^{23}) \oplus S_6(L'^{22}_6 \oplus K_6^{23}) = \Delta L_7^{23}$	$g_{14} = \{K_6^{23}\}$
15	$S_1(L_1^{21} \oplus K_1^{22}) \oplus S_1(L'^{21}_1 \oplus K_1^{22}) = \Delta L_0^{22}$	$g_{15} = \{K_2^{23}, K_1^{22}\}$
16	$S_4(L_4^{21} \oplus K_4^{22}) \oplus S_4(L'^{21}_4 \oplus K_4^{22}) = \Delta L_6^{22}$	$g_{16} = \{K_4^{23}, K_4^{22}\}$
17	$S_3(L_3^{20} \oplus K_3^{21}) \oplus S_3(L'^{20}_3 \oplus K_3^{21}) = \Delta L_1^{21}$	$g_{17} = \{K_3^{23}, K_7^{22}, K_3^{21}\}$
18	$S_2(L_2^{19} \oplus K_2^{20}) \oplus S_2(L'^{19}_2 \oplus K_2^{20}) = \Delta L_3^{20}$	$g_{18} = \{K_1^{23}, K_4^{23}, K_6^{22}, K_5^{21}, K_2^{20}\}$

ture contains 2^{48} plaintexts and about 2^{95} plaintext pairs with the desired differential form. So, by taking 2^n structures, we have 2^{n+95} plaintext pairs with the specified difference in the plaintext side at the end of this step.

Step 2 (Filtering by ciphertext side). Encrypt 2^{n+48} plaintexts and calculate the ciphertext differential values of the 2^{n+95} chosen pairs. Keep the pairs which their ciphertext difference comply with $((*|*|0|*)|(*|*|00)|(0|*|*|0)|(0|0|0*))$. The probability of this condition is almost 2^{-32}; and $2^{n+95-32} = 2^{n+63}$ pairs pass this filter. The complexity of this step is 2^{n+48} 23-round encryption.

Step 3. According to property 2, some input differences to the S-Boxes of the algorithm cannot lead to some specific output differences. Using this property, before testing the key values on each one of 2^{n+63} pair, we can check if the pair can satisfy the cancellations or not, by using differential distribution tables of the S-Boxes.

For a chosen pair (P, P') , Cancellation 1 occurs if $S_1(L_1^0 \oplus K_1^1) \oplus S_1(L'^0_1 \oplus K_1^1 = \Delta R_6^0$. Refer to the differential distribution table of S-Box S_1 with ΔL_1^0 as the differential input to check if an input pair with such difference can lead to the differential output ΔR_6^0 or not. According to the Property 2, $2^{n+63} \times 2^{-1.4}$ pairs can satisfy cancellation 1; and after this filtration, $2^{n+61.6}$ pairs survive. Details of filtration for the other cancellations are given in Table 3.

For each one of the 2^{n+42} survived pairs repeat the following steps.

Step 4 (Cancellations 1, 2, 3, 4, 12, 13, 14). The key values of groups 1, 2, 3, 4, 12, 13 and 14 can be determined by referring to the tables A_1,A_3, A_2, A_5, A_0, A_5, and A_6, respectively (since we know the input pair values and the desired output difference of the corresponding F-functions). Set the flags of the obtained key values for the cancellations 1, 2, 3, 4, 12, 13, and 14 in tables U_1, U_2, U_3, U_4, U_{12}, U_{13}, and U_{14}, respectively. This step contains $(7+(7\times 2^{1.4}))$ memory accesses to find the key values and setting the flags.

Step 5 (Cancellations 5, 10, 11). In step 3, we have filtered the differential pairs, using the differential distribution tables of the S-Boxes corresponding to cancellations 1-18, except 5, 10, and 11. This filtration was possible since the differential input and output of those cancellations were known having the values of the plaintexts and ciphertexts (Table 3). However, since either the differential input or output of S-Boxes corresponding to cancellations 5, 10, and 11 is ΔL_5^1 (Table 4) which is dependent to the subkey K_7^1, we put off verification for these cancellations until we

Table 3. Details of step 3

Cancellation NO.	Corresponding S-Box	Diff. Input and Output	Equivalent Diff. Input and Output	Complexity (memory access)	No. of Survived Pairs
1	S_1	$(\Delta L_1^0, \Delta R_6^0)$	$(\Delta L_1^0, \Delta R_6^0)$	2^{n+63}	$2^{n+61.6}$
2	S_3	$(\Delta L_3^0, \Delta R_7^0)$	$(\Delta L_3^0, \Delta R_7^0)$	$2^{n+61.6}$	$2^{n+60.2}$
3	S_2	$(\Delta L_2^0, \Delta R_1^0)$	$(\Delta L_2^0, \Delta R_1^0)$	$2^{n+60.2}$	$2^{n+58.8}$
4	S_5	$(\Delta L_5^0, \Delta R_2^0)$	$(\Delta L_5^0, \Delta R_2^0)$	$2^{n+58.8}$	$2^{n+57.4}$
6	S_7	$(\Delta L_7^1, \Delta R_3^1)$	$(\Delta L_5^0, \Delta R_3^0)$	$2^{n+57.4}$	2^{n+56}
7	S_6	$(\Delta L_6^1, \Delta R_5^1)$	$(\Delta L_4^0, \Delta R_5^0)$	2^{n+56}	$2^{n+54.6}$
8	S_1	$(\Delta L_1^2, \Delta R_6^2)$	$(\Delta L_7^0, \Delta R_4^0)$	$2^{n+54.6}$	$2^{n+53.2}$
9	S_3	$(\Delta L_3^2, \Delta R_7^2)$	$(\Delta L_1^0, \Delta R_5^0)$	$2^{n+53.2}$	$2^{n+51.8}$
12	S_0	$(\Delta R_0^{23}, \Delta L_2^{23})$	$(\Delta R_0^{23}, \Delta L_2^{23})$	$2^{n+51.8}$	$2^{n+50.4}$
13	S_5	$(\Delta R_5^{23}, \Delta L_4^{23})$	$(\Delta R_5^{23}, \Delta L_4^{23})$	$2^{n+50.4}$	2^{n+49}
14	S_6	$(\Delta R_6^{23}, \Delta L_7^{23})$	$(\Delta R_6^{23}, \Delta L_7^{23})$	2^{n+49}	$2^{n+47.6}$
15	S_1	$(\Delta L_1^{21}, \Delta L_0^{22})$	$(\Delta L_3^{23}, \Delta R_0^{23})$	$2^{n+47.6}$	$2^{n+46.2}$
16	S_4	$(\Delta L_4^{21}, \Delta L_6^{22})$	$(\Delta L_6^{23}, \Delta R_6^{23})$	$2^{n+46.2}$	$2^{n+44.8}$
17	S_3	$(\Delta L_3^{20}, \Delta L_1^{21})$	$(\Delta L_5^{23}, \Delta R_3^{23})$	$2^{n+44.8}$	$2^{n+43.4}$
18	S_2	$(\Delta L_2^{19}, \Delta L_3^{20})$	$(\Delta L_6^{23}, \Delta R_5^{23})$	$2^{n+43.4}$	2^{n+42}

Table 4. Differential input and output of the S-Boxes corresponding to the cancellations 5, 10, and 11.

Cancellation NO.	Corresponding S-Box	Differential Input and Output	Equivalent Differential Input and Output
5	S_5	$(\Delta L_5^1, \Delta R_2^1)$	$(\Delta L_5^1, \Delta L_2^0)$
10	S_7	$(\Delta L_7^3, \Delta R_3^3)$	$(\Delta L_5^1, \Delta L_1^0)$
11	S_3	$(\Delta L_3^4, \Delta R_7^4)$	$(\Delta L_7^0, \Delta L_5^1)$

guess the values for K_7^1 in this step. For 2^4 possible values of K_7^1, calculate 2^4 values of $(L_5^1, {L'}_5^1)$. Check the differential distribution table of S-Boxes S_5, S_7 and S_3 for cancellations 5, 10 and 11, respectively; and discard the values of K_7^1 that their corresponding values in the differential distribution tables are zero. According to property 2, the probability of surviving from these filtrations is $(2^{-1.4})^3 = 2^{-4.2}$; and $2^4 \times 2^{-4.2} = 2^{-0.2}$ values of K_7^1 pass this step. This step contains $(22)^4$ F-function and $2^4 + 2^{2.6} + 2^{1.2}$ memory accesses.

Step 6 (Cancellation 5). For $2^{-0.2}$ possible values of K_7^1, calculate $2^{-0.2}$ values of $(L_5^1, {L'}_5^1)$. Refer to the $(L_5^1, {L'}_5^1, \Delta R_2^1)^{th}$ row of table A_5 to find the values of K_5^2. Set the flags of the achieved values for $(K_7^1|K_5^2)$ in table U_5. This step contains $2 \times 2^{-0.2}$ F-function and $(2^{-0.2} + 2^{1.2})$ memory accesses.

Step 7 (Cancellation 6). For 2^4 possible values of K_6^1, calculate 2^4 values of $(L_7^1, {L'}_7^1)$. Refer to the $(L_7^1, {L'}_7^1, \Delta R_3^1)^{th}$ row of table A_7 to find the values of K_7^2. Set the flags of the achieved values for $(K_6^1|K_7^2)$ in table U_6. This step contains 2×2^4 F-function and $(2^4 + 2^{5.4})$ memory accesses.

Step 8 (Cancellation 7). For 2^4 possible values of K_4^1, calculate 2^4 values of $(L_6^1, {L'}_6^1)$. Refer to the $(L_6^1, {L'}_6^1, \Delta R_5^1)^{th}$ row of table A_6 to find the values of K_6^2. Set the flags of the achieved values for $(K_4^1|K_6^2)$ in table U_7. This step contains 2×2^4 F-function and $(2^4 + 2^{5.4})$ memory accesses.

Step 9 (Cancellation 8). For $2^{1.4}$ possible values of K_2^1 from table U_3, and 2^4 possible values of K_3^2, calculate $2^{5.4}$ values of $(L_1^2, {L'}_1^2)$. Refer to the $(L_1^2, {L'}_1^2, \Delta R_6^2)^{th}$ row of table A_1 to find the values of K_1^3. Set the flags of the achieved values for $(K_2^1|K_3^2|K_1^3)$ in table U_8. This step contains $2 \times 2^{5.4}$ F-function and $(2^{5.4} + 2^{6.8})$ memory accesses.

Step 10 (Cancellation 9). For 2^8 possible values of $K_0^1|K_2^2$, calculate 2^8 values of $(L_3^2, {L'}_3^2)$. Refer to the $(L_3^2, {L'}_3^2, \Delta R_7^2)^{th}$ row of table A_3 to find the values of K_3^3. Set the flags of the achieved values for $K_0^1|K_2^2|K_3^3$ in table U_9. This step contains 2×2^8 F-function and $(2^8 + 2^{9.4})$ memory accesses.

Step 11 (Cancellation 10). For $2^{-0.2}$ possible values of K_7^1 from step 5, $2^{1.4}$ values of K_5^1 in table U_4, and 2^8 possible values of $K_4^2|K_6^3$, calculate $2^{9.2}$ values of $(L_7^3, {L'}_7^3)$. Refer to the $(L_7^3, {L'}_7^3, \Delta R_1^0)^{th}$ row of table A_7 to find the values of K_7^4. Set the flags of the achieved

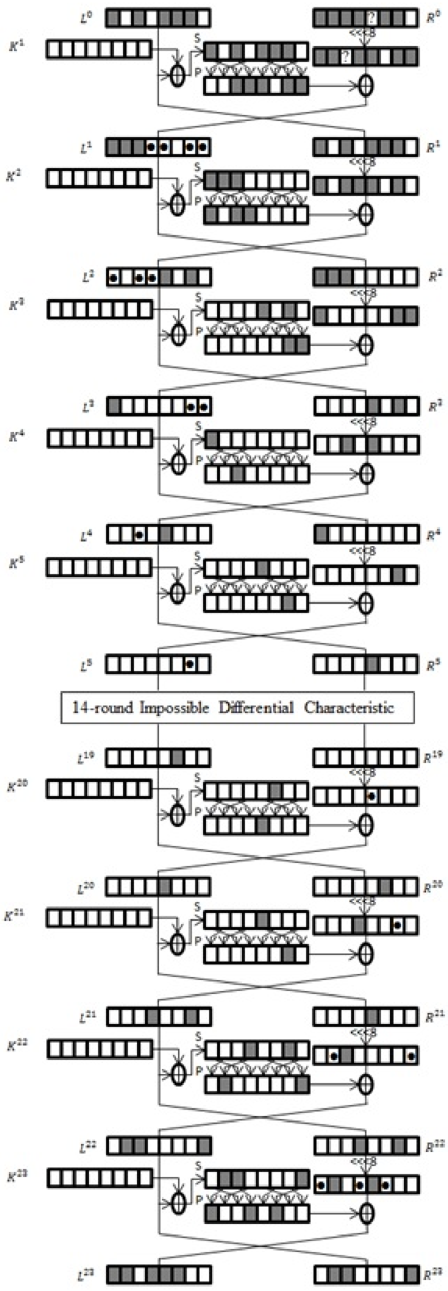

Figure 4. Round function of LBlock. White boxes represent 4-bit zero difference; black boxes present 4-bit non-zero differences; boxes containing question mark present 4-bit either zero or non-zero differences; and boxes containing black circle show the locations of the cancellations.

values for $K_7^1|K_5^1|K_4^2|K_6^3|K_7^4$ in table U_{10}. This step contains $2 \times 2^{9.2}$ F-function and $(2^{9.2} + 2^{10.6})$ memory accesses.

Step 12 (Cancellation 11). According to the key schedule, the most significant bit of K_2^4 is equal to the least significant bit of K_1^1. Also, the two least significant bits of K_0^3 are equal to the two most significant bits of K_5^1. Utilizing these relations among subkeys bits, for $2^{1.4}$ values of K_3^1 in table U_2, $2^{1.4}$ values of K_2^1 in table U_3, $2^{1.4}$ values of K_1^1 in table U_1, $2^{3.35}$ values of K_2^4 using table U_1, $2^{2.7}$ values of K_0^3 using table U_4, and 2^8 possible values of $K_3^2|K_1^2$, calculate $2^{18.25}$ values of (L_3^4, L'^4_3). For $2^{-0.2}$ possible values of K_7^1 from step 5, calculate $2^{-0.2}$ possible values of ΔR_7^4. For $2^{18.25} \times 2^{-0.2} = 2^{18.05}$ possible values of $(L_3^4, L'^4_3, \Delta R_7^4)$, refer to the $(L_3^4, L'^4_3, \Delta R_7^4)^{th}$ row of table A_3 to find the values of K_3^5. Set the flags of the achieved values for $K_7^1|K_3^1|K_2^1|K_1^1|K_3^2|K_1^2|K_0^3|K_2^4|K_3^5$ in table U_{11}. This step contains $2 \times (2^{18.25} + 2^{-0.2})$ F-function and $(2^{18.05} + 2^{19.45})$ memory accesses.

Step 13 (Cancellations 15). For 2^4 possible values of K_2^{23}, calculate 2^4 values of (L_1^{21}, L'^{21}_1). Refer to the $(L_1^{21}, L'^{21}_1, \Delta L_0^{22})^{th}$ row of table A_1 to find the values of K_1^{22}. Set the flags of the achieved values for $(K_2^{23}|K_1^{22})$ in table U_{15}. This step contains 2×2^4 F-function and $(2^4 + 2^{5.4})$ memory accesses.

Step 14 (Cancellation 16). For 2^4 possible values of K_4^{23}, calculate 2^4 values of (L_4^{21}, L'^{21}_4). Refer to the $(L_4^{21}, L'^{21}_4, \Delta L_6^{22})^{th}$ row of table A_4 to find the values of K_4^{22}. Set the flags of the achieved values for $(K_4^{23}|K_4^{22})$ in table U_{16}. This step contains 2×2^4 F-function and $(2^4 + 2^{5.4})$ memory accesses.

Step 15 (Cancellation 17). For 2^8 possible values of $K_3^{23}|K_7^{22}$, calculate 2^8 values of (L_3^{20}, L'^{20}_3). Refer to the $(L_3^{20}, L'^{20}_3, \Delta L_1^{21})^{th}$ row of table A_3 to find the values of K_3^{21}. Set the flags of the achieved values for $K_3^{23}|K_7^{22}|K_3^{21}$ in table U_{17}. This step contains 2×2^8 F-function and $(2^8 + 2^{9.4})$ memory accesses.

Step 16 (Cancellation 18). For 2^{16} possible values of $K_1^{23}|K_4^{23}|K_6^{22}|K_5^{21}$, calculate 2^{16} values of (L_2^{19}, L'^{19}_2). Refer to the $(L_2^{19}, L'^{19}_2, \Delta L_3^{20})^{th}$ row of table A_2 to find the values of K_2^{20}. Set the flags of the achieved values for $K_1^{23}|K_4^{23}|K_6^{22}|K_5^{21}|K_2^{20}$ in table U_{18}. This step contains 2×2^{16} F-function and $(2^{16} + 2^{17.4})$ memory accesses.

Now, we have the key values for 18 cancellations in tables ; and the combinations of these key values are the incorrect keys that should be eliminated from the candidate key space.

Step 17 (Combining the results). In this step, we combine the achieved results stored in tables $U_i, i = 1, 2, \ldots, 17, 18$ to find the incorrect key values and eliminate them. Algorithm 1 shows an efficient method for this purpose. In this algorithm, we use the redundancy of the key schedule to reduce the number of times needed to access the tables. In this way, for a chosen combination of subkeys values from tables $U_1 U_k, k = 1, 2, \ldots, 17$ that their corresponding flags are set, we just check the flags of the subkeys values in table $U_{(k+1)}$ that they comply with the chosen combination according to the redundancy of the key schedule. If their flags are set, they will come to a new combination for being discarded from the target key space.

For instance, for the values of the subkey in $g_1 = K_1^1$, first we chose those that their flags are set in U_1. Then, for each one of the chosen subkey values, we just check the flags of the subkey values of $g_2 - g_1g_2$ in table U_2; where g_ig_j presents the key bits common between g_i and g_j. In this way, if g_1 and g_2 have any bit in common, we just check the flags of the values in U_2 that they comply with the chosen value from U_1 in terms of the chosen value for the common bits from U_1. Generalizing this method for accessing the

Algorithm 1 Eliminating the combinations of the achieved results in tables $U_i, i = 1, \ldots, 18$ from target key space.

```
for 2^(n+42) pairs do
  for the subkey values in g1 do
    if the flag is set in U1 then
      for the the subkey values in g2 − g1g2 do
        if the flag is set in U2 then
          for the the subkey values in g3 − g1g3 − g2g3
          do
            if the flag is set in U3 then
              ⋮
              for the the subkey values in g18 − g1g18 −
              g2g18 − g3g18 − ... g17g18 do
                if the flag is set in U18 then
                  eliminate the 73-bit key value from
                  the target key space
                end if
              end for
              ⋮
            end if
          end for
        end if
      end for
    end if
  end for
end for
```

table $U_{(k+1)}$ for a chosen combination from tables $U_1 \ldots U_k, k = 1, 2, \ldots, 17$, we minimize the time complexity of this step by using the redundancy of the key schedule. In other word, we do not access tables $U_i, i = 1, 2, 3, \ldots, 18$ to check the flags of the subkeys values that cannot be correct according to the key schedule; and reduce the time complexity of this step in this way. Considering the redundancy of the key schedule, there are 2^{73} candidates for the key value in tables $U_i, i = 1, \ldots, 18$, for being eliminated from target key space [10]. Among these 2^{73} values, we just pick those that their corresponding flags are set in tables $U_i, i = 1, \ldots, 18$. The probability of this condition being true for each key value is $2^{(15(-2.6)+3(-4))} = 2^{(-51)}$, where $2^{(-2.6)}$ is the probability for each one of the 15 conditions $1, 2, \ldots, 18$, excluding conditions 5,10 and 11, and $2^{(-4)}$ is the probability for each one of the conditions 5, 10, and 11 holding true . So, the time complexity of this step is $2^{(n+42)} \times 2^{73} \times 2^{(-51)} \times 19 \cong 2^{(n+68.2)}$ memory accesses (18 memory accesses for tables $U_i, i = 1, \ldots, 18$, and one memory access for eliminating the incorrect key value).

4.5 Complexity

Since the attack contains $18 \times 4 = 72$ bits sieving, condition bits, the probability of being chosen for each key value after studying each pair is 2^{-72}. So, the probability of not being chosen for each key value after studying each pair is $(1 - 2^{-72})$. The number of information-key-bits is 73 [10], and after studying m pairs, $N = 2^{73}(1 - 2^{-72})^m$ key values remain in the target key space. Putting $m = 2^{74}$, the number of remained key values is $N = 2^{67.2}$; and we find the correct key among the remained key by exhaustive search. From $m = 2^{n+63} = 2^{74}$, n is 11; and the data complexity of attack is $2^{n+48} = 2^{59}$ chosen plaintexts.

Since we eliminate the incorrect key values from the target key space, we need a memory of size 2^{73} to keep the whole possible values for the target key bits and eliminate the incorrect values among them. So, the memory complexity of the attack is 2^{73}; and sizes of the other used memories are negligible.

The time complexities of the attack's steps are indicated in Table 5. Dominant parts of time complexity refer to steps 3 and 17. Putting $n = 11$, the time complexity of attack is equivalent to $(2^{11+65} + 2^{11+68.2}) \times (\frac{1}{8} \times \frac{1}{23}) = 2^{71.8}$ 23-round encryption.

5 Conclusion

In this paper, we presented a new method for impossible differential cryptanalysis, based on separating target key space into subspaces. To this aim, we search the subspaces separately; then, we extend the combinations of the results to the whole target key space. We applied this method on 23 rounds of LBlock. The presented cryptanalysis requires 2^{59} chosen plaintexts and the time complexity is equivalent to $2^{71.8}$ encryptions, which is the best achieved results to the best of our knowledge. The achieved improvement is independent to those that are proposed so far; and studying their superposition property may lead to more efficient impossible differential attacks. Moreover, despite the fact that the proposed method does not necessarily lead to the most efficient impossible differential attack for all block cipher algorithms, its consideration is important while evaluating the security of an algorithm against this cryptanalysis method.

Table 5. Impossible differential cryptanalysis of 23-round LBlock

Step No.	Target Group	Target Key Bits	No. of Survived Key Values per Pair	Time Complexity
2	None			2^{n+48} 23-round encryption
3	None			2^{n+65} memory accesses
4	1	K_1^1	$2^{1.4}$	$2^{n+42} \times 7(1+2^{1.4})$ memory accesses
	2	K_3^1	$2^{1.4}$	
	3	K_2^1	$2^{1.4}$	
	4	K_5^1	$2^{1.4}$	
	12	K_0^{23}	$2^{1.4}$	
	13	K_5^{23}	$2^{1.4}$	
	14	K_6^{23}	$2^{1.4}$	
5	5,10,11	K_7^1	$2^{-0.2}$	$2^{n+42} \times 2 \times 2^4$ F-function $2^{n+42} \times (2^4+2^{2.6}+2^{1.2})$ memory accesses
6	5	K_7^1	$2^{-0.2}$	$2^{n+42} \times 2 \times 2^{-0.2}$ F-function
		K_5^2	$2^{1.4}$	$2^{n+42} \times (2^{-0.2}+2^{1.2})$ memory accesses
7	6	K_6^1	2^4	$2^{n+42} \times 2 \times 2^4$ F-function
		K_7^2	$2^{1.4}$	$2^{n+42} \times (2^4+2^{5.4})$ memory accesses
8	7	K_4^1	2^4	$2^{n+42} \times 2 \times 2^4$ F-function
		K_6^2	$2^{1.4}$	$2^{n+42} \times (2^4+2^{5.4})$ memory accesses
9	8	K_2^1	$2^{1.4}$	$2^{n+42} \times 2 \times 2^{5.4}$ F-function
		K_3^2	2^4	$2^{n+42} \times (2^{5.4}+2^{6.8})$ memory accesses
		K_1^3	$2^{1.4}$	
10	9	K_0^1	2^4	$2^{n+42} \times 2 \times 2^8$ F-function
		K_2^2	2^4	$2^{n+42} \times (2^8+2^{9.4})$ memory accesses
		K_3^3	$2^{-0.2}$	
11	10	K_7^1	$2^{1.4}$	
		K_5^1	2^4	$2^{n+42} \times 2 \times 2^{9.2}$ F-function
		K_4^2	2^4	
		K_6^3	2^4	$2^{n+42} \times (2^{9.2}+2^{10.6})$ memory accesses
		K_7^4	$2^{1.4}$	
12	11	K_7^1	$2^{-0.2}$	
		K_3^1	$2^{1.4}$	
		K_2^1	$2^{1.4}$	
		K_1^1	$2^{1.4}$	$2^{n+42} \times 2 \times 2^{18.25}$ F-function
		K_3^2	2^4	
		K_1^2	2^4	$2^{n+42} \times (2^{18.05}+2^{19.45})$ memory accesses
		K_0^3	2^4	
		K_2^4	2^4	
		K_3^5	$2^{1.4}$	
13	15	K_2^{23}	2^4	$2^{n+42} \times 2 \times 2^4$ F-function
		K_1^{22}	$2^{1.4}$	$2^{n+42} \times (2^4+2^{5.4})$ memory accesses
14	16	K_4^{23}	2^4	$2^{n+42} \times 2 \times 2^4$ F-function
		K_4^{22}	$2^{1.4}$	$2^{n+42} \times (2^4+2^{5.4})$ memory accesses
15	17	K_3^{23}	2^4	$2^{n+42} \times 2 \times 2^8$ F-function
		K_7^{22}	2^4	$2^{n+42} \times (2^8+2^{9.4})$ memory accesses
		K_3^{21}	$2^{1.4}$	
16	18	K_1^{23}	2^4	
		K_4^{23}	2^4	$2^{n+42} \times 2 \times 2^{16}$ F-function
		K_6^{22}	2^4	
		K_5^{21}	2^4	$2^{n+42} \times (2^{16}+2^{17.4})$ memory accesses
		K_2^{20}	$2^{1.4}$	
17	None			$2^{n+42} \times 2^{73} \times 2^{-51} \times 19 = 2^{n+68.2}$

References

[1] Biham E., Shamir A.: ‘Differential cryptanalysis of DES-like cryptosystems’, *Journal of Cryptology*, pp. 3–72, 1991.

[2] Biham E., Biryukov A., Shamir A.: ‘Cryptanalysis of Skipjack reduced to 31 rounds’, *Proc. of Int. Conf. EUROCRYPT’99*, LNCS 1592, pp. 12–23, 1999.

[3] Wu W., Zhang L.: ‘LBlock: A Lightweight Block

Cipher', *Proc. Int. Conf. ACNS 2011*, LNCS 6715, pp. 327–344, 2011.

[4] Liu Y., Gu D., Liu Z., Li W.: 'Impossible differential attacks on reduced-round LBlock', *Proc. Int. Conf. ISPEC 2012*, LNCS 7232, pp. 97–108, 2012.

[5] Karakoc F., Demirci H., Harmanci A.E.: 'Impossible differential cryptanalysis of reduced-round LBlock', *Proc. Int. Conf. WISTP 2012*, LNCS 7322, pp. 179–188, 2012.

[6] Minier M., Naya-Plasenica M.: 'A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock', *Inf. Process. Lett.*, 112(16), pp. 624–629, 2012.

[7] Knudsen L.R.: 'DEAL A 128-bit cipher', *Technical Report*, Department of Informatics, University of Bergen, Norway, 1998.

[8] Wen L., Wang M., Zhao J.: 'Related-key impossible differential attack on reduced-round LBlock', *Journal of Computer Science and Technology*, 29(1), pp. 165–176, 2014.

[9] Chen J., Futa Y., Miyaji A., Su C.: 'Impossible differential cryptanalysis of LBlock with concrete investigation of key scheduling algorithm', *Proc. Int. Conf. NSS 2014*, LNCS 8792, pp. 184–197, 2014.

[10] Boura C., Minier M., Naya-Plasencia M., Suder V.: 'Improved Impossible Differential Attacks against Round-Reduced LBlock', *Cryptology ePrint Archive*, Report 2014/279, 2014.

[11] shakiba M., Dakhilalian M., Mala H.: 'On computational complexity of impossible differential cryptanalysis', *Inf. Process. Lett.*, 114, pp. 252–255, 2014.

[12] Boura C., Minier M., Naya-Plasencia M., Suder V.: 'Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and SIMON (Full Version)', *Cryptology ePrint Archive*, Report 2014/699, 2014.

[13] Boura C., Minier M., Naya-Plasencia M., Suder V.: 'Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and SIMON', *Proc. Int. Conf. ASIACRYPT 2014*, LNCS 8873, pp. 179–199, 2014.

[14] Soleimani H., Nyberg K.: 'Zero-Correlation Linear Cryptanalysis of Reduced-Round LBlock', *Journal of Des. Codes Cryptogr.*, pp. 683–698, 2014.

[15] Wang Y., Wu W.: 'Improved Multidimensional Zero-Correlation Linear Cryptanalysis and Applications to LBlock and TWINE', *Proc. Int. Conf. ACISP 2014*, LNCS 8544, pp. 1–17, 2014.

[16] Sasaki Y., Wang L.: 'Comprehensive Study of Integral Analysis on 22-round LBlock', *Proc. Int. Conf. ICISC 2012*, LNCS 7839, pp. 156–169, 2012.

[17] Sasaki Y., Wang L.: Meet-in-the-Middle Technique for Integral Attacks against Feistel Ciphers', *Proc. Int. Conf. SAC 2012*, LNCS 7707, pp. 235–251, 2013.

[18] Wang Y., Wu W., Yu X., Zhang L.: 'Security on LBlock against Biclique Cryptanalysis', *Proc. Int. Conf. WISA 2012*, LNCS 7690, pp. 1–14, 2012.

[19] Bogdanov A., Boura C., Rijmen V., Wang M., Wen L., Zhao J.: 'Key Difference Invarient Bias in Block Ciphers', *Proc. Int. Conf. ASIACRYPT 2013*, LNCS 8269, pp. 357–376, 2013.

[20] AlTawy R., Tolba M., Youssef A. M.: 'A Higher Order Key Partitioning Attack with Application to LBlock', *Proc. Int. Conf. C2SI 2015*, LNCS 9084, pp. 215–227, 2015.

[21] Khalesi A., Bahramgiri H., Mansuri D.: 'Impossible Differential Cryptanalysis of LBlock Through Breaking Down the Key Space', *Proc. Int. Conf. ISCISC 2014*, 2014.

Akram Khalesi received her B.S. degree in 2011 from Kashan University, Isfahan, Iran and M.S. degree in 2014 from Malek-Ashtar University, Tehran, Iran, both in Electrical Engineering. She is currently a researcher at Research Center for Development of Advanced Technologies, Tehran, Iran. Her research area includes cryptology with an emphasis on symmetric designs.

Hossein Bahramgiri received the B.S. and M.S. degrees in 2000 and 2003 respectively, from Sharif University of Technology, Tehran, Iran, and the Ph.D. degree in 2010 from University of Tehran, Iran all in Electrical Engineering. He has been a researcher in Malek-Ashtar University since 2010. His research area includes information theory and security and communication networks.

Davod Mansuri received the B.S. and M.S. degrees in Applied Mathematics from University of Guilan, Rasht, Iran and Tehran University, Tehran, Iran, in 1997 and 2001, respectively. He is currently a researcher at Malek-Ashtar University since 2003. His research areas are Cryptography and Cryptanalysis systems.

Appendix

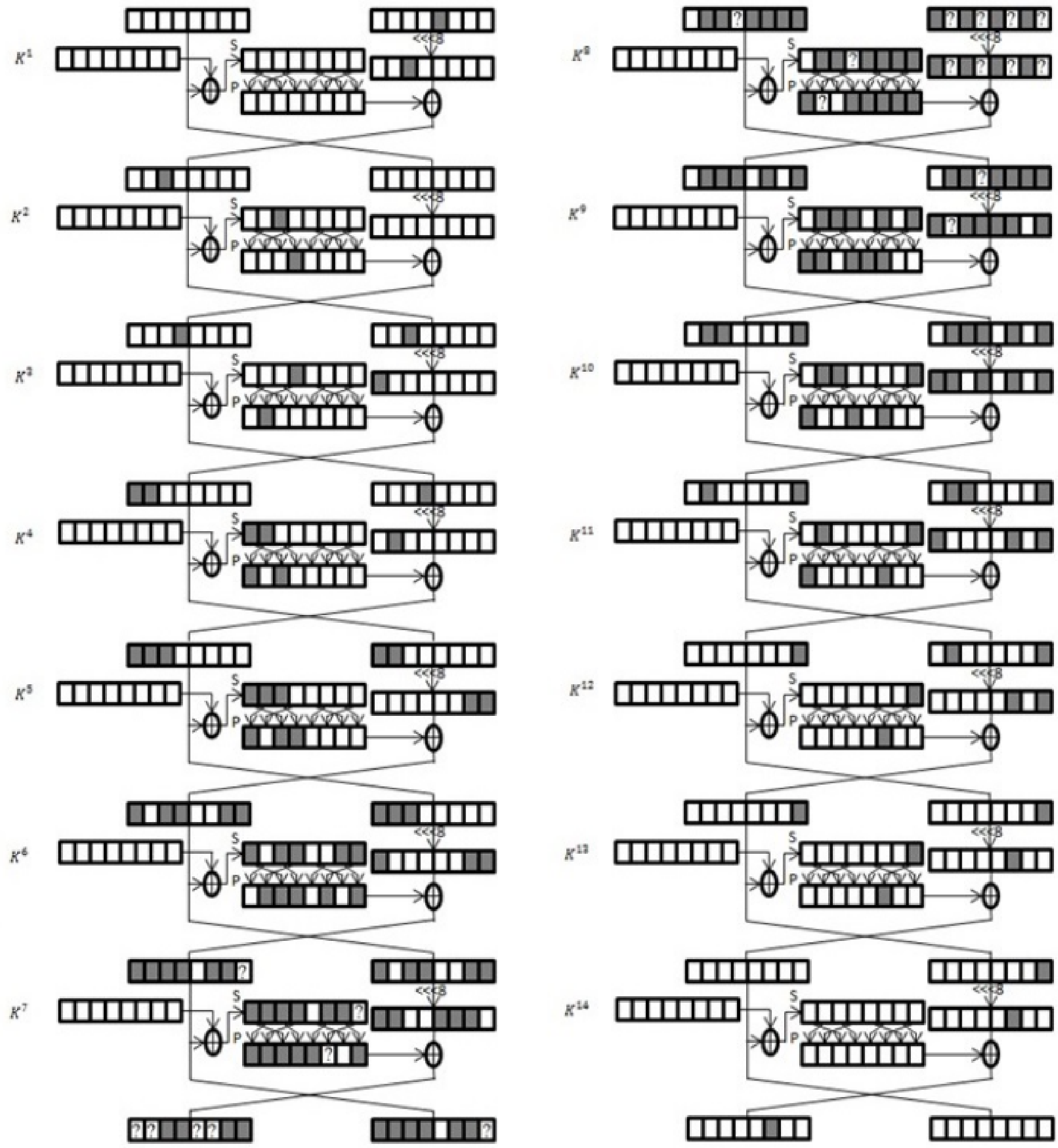

Figure 5. 14-round impossible differential characteristic. White boxes represent 4-bit zero difference; black boxes represent 4-bit non-zero differences; and boxes containing question mark represent 4-bit either zero or non-zero differences.