Differential Cryptanalysis and Boomerang Cryptanalysis of LBlock

Jiageng Chen, Atsuko Miyaji

Differential Cryptanalysis and Boomerang Cryptanalysis of LBlock

Jiageng Chen and Atsuko Miyaji*

School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{jg-chen, miyaji}@jaist.ac.jp

Abstract. LBlock is a lightweight block cipher proposed in ACNS 2011. It has a 64-bit block size and 80-bit key size which is the typical parameter setting accepted by most of the recent proposed lightweight block ciphers. It has fast hardware implementation efficiency and it still remains rather secure considering the recent results and the security margin it provides. In this paper, we investigate the differential behavior of the cipher in detail and propose (multiple) differential attack and boomerang attack against it. We are able to construct 15-round multiple differential paths which can lead to 17-round attack with complexity as low as $2^{67.52}$. Also 16-round boomerang distinguisher can be build which leads us to 18-round boomerang (rectangle) attack with complexity $2^{70.8473}$. These are the best differential attacks for LBlock in the single key scenario, which helps us understanding the differential behavior of the cipher.

Keywords: LBlock, ultra lightweight block cipher, multiple differential cryptanalysis, boomerang attack, ladder switch

1 Introduction

Lightweight block ciphers have attracted much of the research attention due to the cheap computational cost in both hardware and software implementation which is suitable for resource-restricted devices such as RFID tag and so on. The security margin they provide, although reduced compared with the traditional block ciphers, is considered to be reasonable given the cost of information being protected. Generally speaking, key size is usually chosen to be 80 bits, while the popular versions of block size are 32, 48 and 64 bits. The first famous block cipher that was widely considered to be lightweight is PRESENT [4]. After that, many lightweight block ciphers have been proposed such as KATAN/KTANTAN family [5], TWINE [11], PRINTcipher [7], LBlock [13] and so on. Compared with AES which was selected through competitions, lightweight block ciphers get started only recently, and the lack of enough cryptanalysis will prevent those

* This study is partly supported by Grant-in-Aid for Scientific Research (A) (21240001).

ciphers from being adopted by the industrial world. In this paper, we target one of the recent proposed cipher LBlock, which being as a recent cipher, still needs a lot of security analysis to be performed on it before we are able to have confidence in its security.

In ACNS2011, LBlock [13] was proposed as a lightweight block which targets fast hardware and software implementation. It is designed using a 32-round Feistel structure with a 64-bit block size and 80-bit key size. In the original paper, the authors gave several attacks against LBlock, among which the impossible differential attack is the best one that can attack 20 rounds. This record was later improved by [8] and [6] to 21 and 22 rounds using the same impossible differential technique. For differential related attack, the original paper only mentioned the active S-Boxes from which it drew the conclusion that no useful differential path is available for more than 15 rounds. Later in [9], the authors first analyze the differential behavior in detail and proposed 12 and 13 rounds attack which improved the bound in the original paper.

In this paper, we further investigate the differential behavior of LBlock and proposed two attacks in single key scenario. The first one is differential attack by using single differential and multiple differentials. We take advantage of the multiple differential statistic model [3] to evaluate the success probability and the time complexity. 15-round single differential path with probability $2^{-61.2351}$ is found and 17-round attack can be performed based on it. $2^{74.23}$ is the best cost we can achieve by using one single path. We then take advantage of the fact that there exists a set of such efficient differential paths, which can be used to further reduce the time complexity and the data complexity in multiple differential statistic model. As a result, we can break 17-round cipher with best time complexity of $2^{67.5211}$. Secondly, we apply the boomerang attack to further investigate the short differential behavior of LBlock instead of a long one. We are able to build a 16-round boomerang distinguisher including an eight-round upper trail and an eight-round lower trail. This cannot be achieved without applying the ladder switch technique in the middle of the switching point, which can help us to escape three active S-Boxes. The key recovery phase follows the rectangle procedure which can as a result break 18 rounds of the cipher with complexity $2^{70.8437}$. The results are summarized in Table 1.

Table 1. Single key scenario attacks against LBlock

# Round	Methods	Time Complexity	Data Complexity	Source
18	Integral Attack	$2^{62.3}$	$2^{62.3}$	[13]
22	Impossible Differential Attack	$2^{79.28}$	2^{58}	[6]
20	Impossible Differential Attack	2^{63}	$2^{72.7}$	[13]
21	Impossible Differential Attack	$2^{62.5}$	$2^{73.7}$	[8]
13	Differential Attack	$2^{42.08}$	$2^{42.08}$	[9]
17	Differential Attack	$2^{67.52}$	$2^{59.75}$	This paper
18	Boomerang Attack	$2^{70.84}$	$2^{63.27}$	This paper

This paper is organized as follows. Section 2 describes the specification of LBlock. In Section 3, we describe the differnetial and multiple differential attack against 17-round LBlock. Section 4 demonstrates the boomerang attack against 18-round LBlock with path searching and ladder switch techniques included. Finally Section 5 concludes the paper.

2 LBlock

LBlock consists of a 32-round variant Feistel network with 64-bit block size and 80-bit key size. The encryption algorithm works as follows:

1. For $i = 2, 3, ..., 33$, do $X_i = F(X_{i-1}, K_{i-1}) \oplus (X_{i-2} <<< 8)$
2. Ciphertext is $C = X_{32}||X_{33}$

Here round function F contains a S-Box layer and a diffusion layer which are denoted as S and P.

$$F : \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}, (X, K_i) \rightarrow P(S(X \oplus K_i))$$

There are eight 4-bit S-Boxes for each of the nibbles. Suppose the input and output of the S-boxes are Y and Z. The S layer can be denoted as

$$Y = Y_7||Y_6||Y_5||Y_4||Y_3||Y_2||Y_1||Y_0 \rightarrow Z = Z_7||Z_6||Z_5||Z_4||Z_3||Z_2||Z_1||Z_0$$

$$Z_7 = s_7(Y_7), Z_6 = s_6(Y_6), Z_5 = s_5(Y_5), Z_4 = s_4(Y_4)$$

$$Z_3 = s_3(Y_3), Z_2 = s_2(Y_2), Z_1 = s_1(Y_1), Z_0 = s_0(Y_0)$$

For diffusion layer with the input and output of the layer being Z and U, it can be denoted as:

$$U_7 = Z_6, U_6 = Z_4, U_5 = Z_7, U_4 = Z_5, U_3 = Z_2, U_2 = Z_0, U_1 = Z_3, U_0 = Z_1$$

All the above details are concluded in Figure 1. Key schedule part is not used during the analysis so we omit the description here. Please refer to [13] for the details.

3 (Multiple) Differential Attack against 17-round LBlock

3.1 Statistical Framework of Multiple Differential Attack

[10] addressed the success probability of linear and differential attack, and this result has been used since then widely. However, the normal distribution approximation for the differential attack is not accurate, which is later improved by [12] with a hybrid distribution. When dealing with multiple differentials with different probabilities, the counter itself does no longer follows a binomial distribution, so a new formula should be used to address the success probability in this scenario. A solution is given in [3], which proposed a general framework by

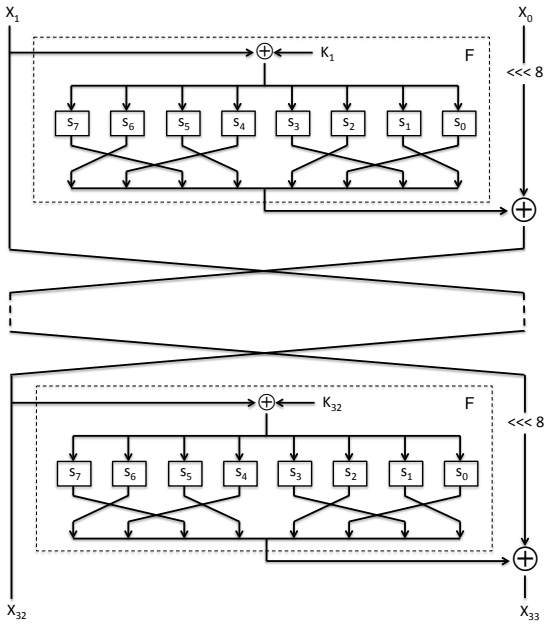

Fig. 1. LBlock

expressing the distribution of counters in terms of a hybrid distribution which include Kullback-Leibler divergence and a Poisson distribution. Please refer to the Appendix for the success probability of the multiple-differential cryptanalysis. We will evaluate the complexity based on this statistical model for the multiple differential attack in this paper.

The differential set described in [3] gives a direct way to connect with the statistical model they proposed. While it is very obvious to understand from the statistical model's point of view, the restriction of the differential set is too strong which limits the practical attack. Actually, the typical way of doing differential cryptanalysis by using a hash table can easily avoid this restriction as described in [12]. In [3], only differential sets satisfied specific conditions can be used. In [12], many to one differential paths are used. Actually, by using following algorithm, we can avoid double counting which gives us a wide number of options, and the only concern is how to optimize the result by using these multiple differential paths. In this paper, due to the property of differential paths, we found out that one to many differential paths can be used to achieve a relatively good result, which is exactly the opposite to the pattern in [12].

3.2 Notations and configurations for the (multiple)differential cryptanalysis

The key recovery algorithm was pretty well summarized by [12], and we summarize it in the Appendix. By applying the algorithm, differential path will not be double counted and statistical model of [3] can still be used. Related notations and configurations that are required for the further reading are described as follows.

- m: the block size of the block cipher.
- k: the key size of the block cipher.
- $|\Delta_0|$: the number of differentials.
- p_i: the probability of the differential with input difference Δ_0^i.
- N_p: the number of plain texts bits involved in the active S-boxes in the first round for all differentials.
- N_c: the number of cipher texts bits involved in the non-active S-boxes in the last round deriving from Δ_r.
- N_s: the number of samples pairs required of the cryptanalysis.
- β: the filtering probability for the ciphertext pairs.
- p_f : the filtering probability for the ciphertext pairs according to active S-boxes, $p_f = \beta \cdot 2^{N_C}$.
- l: the size of the candidate key list is 2^l.
- n_k: the number of guessed sub key bits in the last $R - r$ rounds.
- N: Data complexity is 2^N.
- $2^{N_{st}}$ structures are constructed.

3.3 Strategies for finding differentials for LBlock

Iterative differential paths are widely used in differential cryptanalysis such as DES and PRESENT, etc. However, we found that iterative differential path will not lead to better result here for LBlock. We search the iterative differential path and found that due to the 4-bit block permutation, the iterative path is rather well controlled, this could be seen from Table 2, which shows the active S-Boxes in each round.

Table 2. AS for Iterative Differential Paths

	2R	3R	4R	5R	6R	7R	8R
AS	9	10	16	20	18	20	18

Thus we switch to non-iterative differential path. Due to the property that the internal bits will be permutated only between the S-boxes, we could consider first to run a truncated differential search which treats all the input to the 4-bit S-box as 0 or 1 for r round characteristic. By running the truncated differential search, first we confirm the result of smallest active S-Boxes in each round of

[13], especially, for round 14 and 15 we are interested in, the number of the smallest active S-boxes is 30 and 32. Given the smallest differential probability is 2^{-2} for each S-Box, 15 rounds seems to be the maximum bound for differential attack. Then for each of the structure candidate which has achieved the best number of active S-boxes, we derive the specific differentials by using branch and bound algorithm. All the differential paths with probability greater than 2^{-72} are considered, which is mostly decided based on the experimental experience. It shows that further smaller probability paths will not make any improvements on the total probability any more. We list the truncated differential path with the largest probability along with the corresponding concrete differential paths we found in Table 3.

Table 3. Best 15-round Differential Paths

Truncated Diff	Best Diff	$log_2(Prob)$	#Diff with $Prob < 2^{-64}$
0000000011010000 ↓ 0001111000100100	0000000011030000, 0003222000200100 0000000011030000, 0003422000200900 0000000011030000, 0003522000200900 0000000011030000, 0003622000200100 0000000011030000, 000*b*222000200100 0000000011030000, 000*b*422000200900 0000000011030000, 000*b*522000200900 0000000011030000, 000*b*622000200100	−61.2351	1290

We can derive from Table 3 that for the truncated form ($0000000011010000 \rightarrow 00011110001001000$), differential paths with the largest probability $2^{-61.2351}$ can be found. All the other truncated forms we found have a smaller probability than this one and we omit the description here. We also list the number of specific differential paths with probability larger than the average value 2^{-64}, which forms a structure that we can take advantage in multiple differential cryptanalysis.

3.4 Key recovery attack on 17-round LBlock using single differential path

For reaching 16 and 17 rounds, given the output truncated differential $\Delta_{15} = (00011110, 00100100)$, we can get $(00011110, 00100100) \rightarrow (11011011, 00011110) \rightarrow (1 \star \star 11111, 11011011)$, where $\star$ denotes the differential status that can not be decided. In round 17, except the nibble for S-Boxes S_2 and S_5, differentials are involved for other S-Boxes, and thus we target $6 \times 4 = 24$ bits of k_{17}. In round 16, active S-boxes involve S_1, S_2, S_3 and S_4, thus we target $4 \times 4 = 16$ bits of k_{16}. In total, $n_k = 40$ bits. $N_p = 4 \times 3 = 12$ bits according to the truncated input differential. Assuming the data complexity is 2^N, then the number of structure can be derived as $N_{st} = N - 12$, and each structure contains 2^{12} plaintexts. At the beginning, we have in total $2^{N-12} \cdot 2^{2\times 12-1} = 2^{N+11}$

pairs to consider. Inserting the ciphertexts into the hash table according to the nibble e_{17}^2 and e_{17}^5 will take complexity 2^N and also the same amount of memory cost. For each structure, we have 2^{23} pairs to consider at the beginning, and after inserting into the hash table, we have left $2^{23-8} = 2^{15}$ pairs. By studying the propagation of the differentials in the last two rounds, we can further filter out the pairs whose differentials are definitely impossible. Since $e_{17}^0 = e_{16}^0 = e_{15}^{10}, e_{17}^3 = e_{16}^3 = e_{15}^{13}, e_{17}^{11} = e_{16}^{13} = e_{15}^5, e_{17}^{12} = e_{16}^{14} = e_{15}^6$, we have another 16-bit filter which leaves us with $2^{15-16} = 2^{-1}$ pairs. Also we have $e_{17}^1 = e_{16}^1 = S_4(e_{15}^3)$, $e_{17}^4 = e_{16}^4 = S_2(e_{15}^5)$, $e_{17}^6 = e_{16}^6 = S_3(e_{15}^4)$, $e_{17}^7 = e_{16}^7 = S_1(e_{15}^6)$, $e_{17}^9 = e_{15}^3 \oplus S_4(e_{15}^{13})$ and $e_{17}^{10} = e_{15}^4 \oplus S_7(e_{15}^{10})$. Thus for the nibbles $1, 4, 6, 7, 9, 10$ in the ciphertexts, some of the differentials are not possible according to the differential tables. We compute the average probability that an output difference can be achieved given an input difference for $S_0, ..., S_7$, and we find that $P_{S_0} \approx \cdots \approx P_{S_7} \approx 0.4267$. Thus after this filtering, there remains $2^{-1} \cdot (0.4267)^6 = 2^{-8.37}$ pairs for each structure and $2^{N-12-8.37} = 2^{N-20.37}$ pairs in total. For each of these pairs, we check whether the corresponding input differences are legal or not. This step will take computational cost $2^{N-20.37}$ and $2^{N-20.37-12} = 2^{N-32.37}$ pairs remain. Then for each of these pairs, we guess the 40-bit subkey in round 16 and 17 to decrypt the ciphertext pairs to see if it will result in the corresponding Δ_{15}. This step takes time $2^{N-32.37} \times 2^{40} = 2^{N+7.63}$. Given the size of the key candidate list 2^l, searching the candidate key list will take time 2^{40+l}. As a result, the total complexity is $2^N + 2^{N-20.37} + 2^{N+7.63} + 2^{40+l}$.

Now we have two parameters involved in the computational complexity, namely, data complexity 2^N and the size of key candidate list 2^l. Here we want the success probability to be as high as 90%. This standard can be measured by the following formula which can be derived from the framework described in Section 3.1, and also involves only the above two parameters while others are fixed.

$$2^N = -4 \cdot \frac{ln(2\sqrt{\pi}2^l 2^{-n_k})}{|\Delta_0| D(p_*||p)}$$

According to Table 3, we choose the best probability path with $p_* = 2^{-61.2351}$. Then we can derive the relations between l, N and the computational complexity as shown in Table 4.

Table 4. Size of the key candidate list, data complexity and computational cost with success probability 90%

l	34	35	36	37	38
N	63.8294	63.4343	62.8884	61.9996	59.2471
$log(Time)$	74.2300	75.0917	76.0321	77.0087	78.0007

From Table 4, we can see that if data complexity is the bottleneck, then we can choose $l = 38, N = 59.2471$ which gives the complexity cost 2^{78}. On the other hand, if the computational cost is the bottleneck, we can set $l = 34, N = 63.8294$ which gives the computational cost $2^{74.23}$. Both cases can lead to the break of the 17-round LBlock.

3.5 Key recovery attack using multiple differential paths

Let's investigate the situation where multiple differential paths are used. Table 3 demonstrates the best probability differential paths along with the largest amount of differential paths. So we investigate this truncated differential category to search of the best multiple differential path. Through the experiment, we found that the best computational complexity and the best data complexity can be both derived by using the 188 best differential paths. If we choose $l = 38$, the data complexity can be as small as $N = 53.4064$ while the computational cost is $2^{78.0}$. If we decrease the size of the key candidate list to $l = 24$, the computational complexity can be reduced $2^{67.5211}$ and the corresponding data complexity will increase to $N = 59.7523$. Of course, balance can always be achieved in between. As a result, we can see that multiple differential paths are effective for LBlock cipher. Due to the space limit, we omit these 188 paths here.

4 Boomerang attack against 18-round LBlock

The great idea of boomerang attack is to use two short efficient differentials instead of one long differential, hope to do better than the traditional differential attack. The boomerang distinguisher is usually denoted by a cascade cipher $E = E_1 \cdot E_0$, where E_0 has a differential $\alpha \to \beta$ with probability p and E_1 has a differential $\gamma \to \delta$ with probability q. Basic boomerang attack is an adaptive chosen ciphertext attack, and later it was extended to rectangle attack which is a non-adaptive chosen plaintext attack. The attacker encrypts many plaintext pairs with input difference α, and collects quartets which satisfy $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$ and $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$. Three conditions should be satisfied in this scenario, namely, $E_0(P_1) \oplus E_0(P_2) = E_0(P_3) \oplus E_0(P_4) = \beta$, $E_0(P_1) \oplus E_0(P_3) = \gamma$ and $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$. Figure 2 shows the boomerang structure with E_0, E_1, plaintext quartets, ciphertext quartets and the corresponding differentials. It was noted that the probability of p and q can be increased by exploiting multiple differentials as $\hat{p} = \sqrt{\sum_\beta Pr^2[\alpha \to \beta]}$ and $\hat{q} = \sqrt{\sum_\gamma Pr^2[\gamma \to \delta]}$. It is well known that if $\hat{p}\hat{q} > 2^{-n/2}$, cipher can be distinguished from a random permutation. The number of right quartets can be computed by $N^2 \cdot 2^{-n}\hat{p}^2\hat{q}^2$ given N number of plaintext pairs. In this paper, the boomerang attack used is indeed the rectangle attack but we keep the name boomerang attack for simplicity. For the details of the boomerang attack and related rectangle attack, please refer to paper [1].

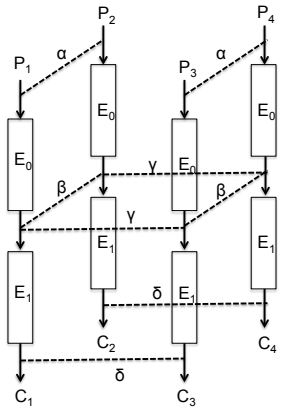

Fig. 2. Boomerang Structure

4.1 Differential path

We search the differential path of E_0 using the similar strategies with the multiple differential analysis. First we find the best truncated differential path, and then search the concrete path using the branch-and-bound algorithm. The best truncated E_0 trail is shown in Table 5. Each line of L and R represents the differential states after the current round. Round 0 denotes the initial value before the first round.

Table 5. 8-round E_0 Differential Path

Round	AS	L	R
0	0	00001010	11100000
1	2	10000000	00001010
2	3	00001000	10000000
3	4	00000000	00001000
4	4	00100000	00000000
5	5	00010000	00100000
6	6	11000000	00010000
7	8	11100000	11000000
8	11	10110011	11100000

Table 6. 8-round E_1 Differential Path

Round	AS	L	R
0	0	00011100	10111010
1	3	10100000	00011100
2	5	01000000	10100000
3	6	00000010	01000000
4	7	00000000	00000010
5	7	00001000	00000000
6	8	00000010	00001000
7	9	00100001	00000010
8	11	00011100	00100001

After searching all the concrete paths, it gives 128 paths with probability 2^{-22}, 1312 paths with probability 2^{-23}, 4672 paths with probability 2^{-24}, 7040 paths with probability 2^{-25} and 3840 paths with probability 2^{-26}. As a result, we can compute $\hat{p}$ as follows:

$$\hat{p} = \sqrt{128 \cdot 2^{-22} + 1312 \cdot 2^{-23} + 4672 \cdot 2^{-24} + 7040 \cdot 2^{-25} + 3840 \cdot 2^{-26}} = 2^{-17.1151}$$

For the lower trail E_1, obviously we need to do better than $2^{-17.1151}$ in order to launch an effective attack. We investigate the concrete paths within the same truncated structure first, and then try to gather paths with multiple input differences which will generate the same output difference. After evaluate the total probabilities, we generate several candidates with the best probability close to each other. We pick the 8-round trail in Table 6 for the attack use. The reason that we choose this trail is related to the ladder switch technique descirbed in the following section.
E_1 truncated structure gives us the best probability:

$$\hat{q} = \sqrt{16 \cdot 2^{-22} + 96 \cdot 2^{-23} + 192 \cdot 2^{-24}} = 2^{-19.1498}$$

4.2 Ladder switch

Unfortunately, the probability of $\hat{p}$ and $\hat{q}$ is still too small for us to build an effective boomerang distinguisher, which requires $\hat{p}\hat{q} > 2^{-32}$. Here we consider to apply ladder switch to increase the path probability, which was first proposed in [2] to attack the full-round AES in the related-key model. The basic idea of ladder switch is that instead of dividing the cipher into separate rounds, we can go further to divide based on concrete operations as long as they are parallel independent with each other. If the final round of E_0 or the first round of E_1 has many active S-boxes, we can consider to switch the active S-boxes to the upper or lower trail where the S-Boxes are non-active, so that we don't need to pay the corresponding probabilities. Figure 3 shows the switch given the last

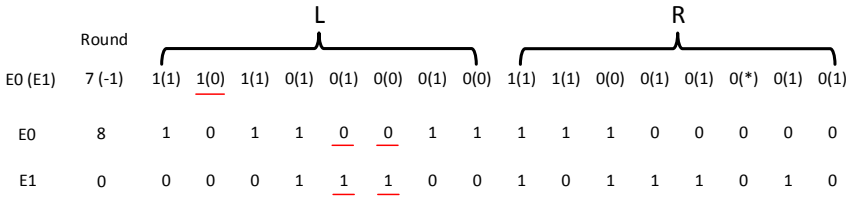

Fig. 3. Ladder switch

round of E_0 and the first round of E_1. If no switch is performed, we would pay for three active S-Boxes in the last round for E_0, and also three active S-Boxes for the first round of E_1. For nibble 10 and 11 which are active in E_1, we can set the switching point after the S-Box operation, since this two nibbles in the last round of E_0 are not active which can help us save two active S-Boxes. Due to the property of Feistel structure, we can go back one round and derive the differential path of L for E_1. The 14th nibble in round 8 of E_0 is active while

it is not active for round -1 of E_1. Thus we can set the switching point of this nibble right after the S-Box operation in round 8 of E_0 instead of the last nibble after round 8. For the differential of R in round -1 of E_1, there is one $*$ which we can not determine, but that does not affect other differential value since they all can be computed independently. In total this helps us saving three S-Boxes, one from E_0 and two from E_1. Thus the boomerang distinguisher probability is now increased to:

$$\hat{p}\hat{q} = 2^{-15.1151-15.1498} = 2^{-30.2149} > 2^{-32}$$

Namely, we are able to observe one right quartet in $2^{62.2649}$ plaintext pairs.

4.3 Key recovery procedure

We target to attack 18 rounds LBlock using 16 rounds boomerang distinguisher. Before E_0 we add one round E_b and after E_1 we add another rounds E_f. Thus the structure of the cipher can be described as $E = E_f \cdot E_1 \cdot E_0 \cdot E_b$. The structure of the chosen plaintexts is organized as $2^{31.2649}$ structures with 2^{32} plaintexts each. In each structure, we can form 2^{31} pairs that follow the input difference α. In total, we have $2^{62.2649}$ pairs which gives us $2^{124.5298}$ quartets. The number of right quartets can be computed by $(2^{31} \cdot 2^{31.2649})^2 \cdot 2^{-64} \cdot (\hat{p}\hat{q})^2 = 1$. The key recovery algorithm works as follows:

1. Generate $2^{31.2649}$ structures of 2^{32} plaintexts each. Ask for the encryption of these plaintexts and get the corresponding ciphertexts. $2^{63.2649}$ data complexity is required and $2^{63.2649}$ time complexity is required for 18-round LBlock encryption.
2. Generate $2^{12+12} = 2^{24}$ counters for the 24-bit subkeys in E_b and E_f. This costs time complexity 2^{24} memory access.
3. Insert all the $2^{63.2649}$ ciphertexts into a hash table indexed by 32 bit of non active bits of the output truncated differential $0001110000100001(\delta) \rightarrow 0001110011001110$. This gives us 2^{32} entries with $2^{31.2649}$ ciphertexts in each of the entries. There are total $2^{61.5298}$ pairs for each of the entry. Some of them can be filtered according to the differential pattern. There are 5 nibbles where the differentials are fixed according to the differential δ which can be filtered with probability $2^{-5\times4}$. There are another 3 nibbles pass through S-boxes, and thus the filter probability become $(0.4267)^3$ considering the average probabilities. So in total there remains $2^{61.5298} \cdot 2^{-5\times4} \cdot (0.4267)^3 = 2^{37.8437}$ pairs. Note that we don't need to search all the $2^{61.5298}$ pairs to generate the remaining $2^{37.8437}$ pairs. We can apply the meet in the middle approach to first sort the ciphertexts in each of the entry, and then for every ciphertexts in the entry, add the corresponding differential and check if it equals the ciphertext in the sorted table or not. The cost for each of the entry is slightly more than $2^{31.2649}$ which is not the dominant cost. In order to check if the ciphertext difference is the expected difference, we need to do $2^{32} \cdot 2^{37.8437} = 2^{69.8437}$ memory access.

4. For each of the remaining ciphertext pairs, we try to test the plaintext pairs (P_1, P_2) and (P_3, P_4) to see if they can form a quartet candidate. According to the pattern in E_b, we have $1110000010101110 \rightarrow 0000101011100000(\alpha)$. Thus, 5 nibbles should be exact the same as the input difference α, and 3 nibbles go through S-Boxes. This provides filtering probability $2^{-5\times4} \times 2^{-2.678\times3}$. Also the proper plaintext pairs should be in the same structure, which takes probability 2^{-32}. As a result, the number of quartet candidates is $2^{69.8437} \times 2^{69.8437} \times 2^{-32\times2} \times (2^{-5\times4})^2 \times 0.4267^{3\times2} = 2^{28.3152}$. By using the same meet in the middle approach as in step 3, we can perform the check ((P_1, P_2) and (P_3, P_4)) with $2^{70.8437}$ memory accesses.
5. For each of the candidate quartets, we encrypt E_b and decrypt E_f using the 24-bit subkey, if the differential matches with the characteristic, add one to the corresponding subkey counter. This step takes time complexity $2^{28.3152} \times 2^{24} = 2^{52.3152}$.

Given the average probability for the each S-Box $\frac{1}{16\times0.4267} = 2^{-2.77}$, the probability for the wrong key to be suggested by a quartet is $(2^{-2.77\times3} \times 2^{-2.77\times3})^2 = 2^{-33.24}$. Then the number of subkeys suggested by one quartet is $2^{24} \times 2^{-33.24} = 2^{-9.24}$. Thus all candidate quartets suggest $2^{28.3152} \times 2^{-9.24} = 2^{19.0751}$, which means the expected number of times a wrong key gets suggested is $2^{-4.93}$. This will guarantee us to eliminate almost all the wrong keys. It is clear that step 4 dominants the time complexity which requires $2^{70.8437}$ memory access, and the data complexity is $2^{63.2649}$.

5 Conclusion

In this paper, we take a deep investigation of the differential behavior of lightweight block cipher LBlock, which was proposed recently. We are able to build 15-round non-iterative differential path based on which 17-round (multiple) differential attack is available with complexity $2^{67.52}$. Then we investigate the security of the cipher against boomerang attack. Firstly based on the optimized searching and ladder switch technique, we build a 16 rounds boomerang distinguisher which contains two 8 sub trails E_0 and E_1. Then 18-round attack is successfully applied with complexity $2^{70.8437}$. Our result doesn't pose any threat to the full round LBlock, but help us understanding the differential behavior and its strength under differential attack and boomerang attack.

References

1. Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer, 2001.
2. Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full aes-192 and aes-256. In Mitsuru Matsui, editor, *Advances in Cryptology ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2009.

3. Cline Blondeau and Benot Grard. Multiple differential cryptanalysis: Theory and practice. In Antoine Joux, editor, *Fast Software Encryption*, volume 6733 of *Lecture Notes in Computer Science*, pages 35–54. Springer Berlin Heidelberg, 2011.

4. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

5. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.

6. Ferhat Karako, Hseyin Demirci, and A.Emre Harmanc. Impossible differential cryptanalysis of reduced-round lblock. In Ioannis Askoxylakis, HenrichC. Phls, and Joachim Posegga, editors, *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*, volume 7322 of *Lecture Notes in Computer Science*, pages 179–188. Springer Berlin Heidelberg, 2012.

7. L. R. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw. PRINTcipher: A Block Cipher for IC-Printing. In S. Mangard and F.-X. Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.

8. Ya Liu, Dawu Gu, Zhiqiang Liu, and Wei Li. Impossible differential attacks on reduced-round lblock. In MarkD. Ryan, Ben Smyth, and Guilin Wang, editors, *Information Security Practice and Experience*, volume 7232 of *Lecture Notes in Computer Science*, pages 97–108. Springer Berlin Heidelberg, 2012.

9. Minier. Marine and Maria Naya-Plasencia. Some preliminary studies on the differential behavior of te lightweight block cipher LBlock. In *ECRYPT Workshop on Lightweight Cryptography*, pages 35–48, 2011.

10. Ali Aydin Selçuk and Ali Biçak. On probability of success in linear and differential cryptanalysis. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 174–185. Springer, 2002.

11. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In LarsR. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer Berlin Heidelberg, 2013.

12. Meiqin Wang, Yue Sun, Elmar Tischhauser, and Bart Preneel. A model for structure attacks, with applications to present and serpent. In Anne Canteaut, editor, *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 49–68. Springer, 2012.

13. Wenling Wu and Lei Zhang. Lblock: A lightweight block cipher. In Javier Lopez and Gene Tsudik, editors, *Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*, pages 327–344. Springer Berlin Heidelberg, 2011.

Appendix

Probability evaluation for (multiple)differential cryptanalysis

The success probability of the multiple-differential attack can be derived as follows [3]. First let $p_* = \frac{\sum_{i,j} p_*^{(i,j)}}{|\Delta_0|}$ and $p = \frac{|\Delta|}{2^m|\Delta_0|}$. p_* denotes the average probability of the multiple differentials and p denote average probability for the wrong key case. Define $G_*(\tau) = G(\tau, p_*), G(\tau) = G(\tau, p)$, which is defined as follows:

$$G(\tau, q) = \begin{cases} G_-(\tau, q), & \text{if } \tau < q - 3\sqrt{q/N_s} \\ 1 - G_+(\tau, q), & \text{if } \tau > q + 3\sqrt{q/N_s} \\ G_p(\tau, q), & \text{otherwise} \end{cases}$$

$$G_- = e^{-N_s D(\tau||q)} \cdot [\frac{q\sqrt{1-\tau}}{(q-\tau)\sqrt{2\pi\tau N_s}} + \frac{1}{\sqrt{8\pi\tau N_s}}]$$

$$G_+ = e^{-N_s D(\tau||q)} \cdot [\frac{(1-q)\sqrt{\tau}}{(\tau-q)\sqrt{2\pi N_s(1-\tau)}} + \frac{1}{\sqrt{8\pi\tau N_s}}]$$

$$D(\tau||q) = \tau ln(\frac{\tau}{q}) + (1-\tau) ln(\frac{1-\tau}{1-q}) \text{ (Kullback-Leibler divergence)}$$

And the success probability is defined as follows:

$$P_S \approx 1 - G_*[G^{-1}(1 - \frac{l-1}{2^{n_k} - 2}) - 1/N_s],\ G^{-1}(y) = min\{x|G(x) \geq y\}$$

Note that the above formula is also effective in the case of single differential path where $|\Delta| = |\Delta_0| = |\Delta_r| = 1$.

Key recovery procedure for (multiple)differential cryptanalysis

Key recovery procedure is summarized in Table 7.

Complexity

Denote by $T_{1-1}, T_{1-2}, T_{1-3}, T_{1-4}$ and T_3 the time complexity for step 1-1, 1-2, 1-3, 1-4 and step 3. We ignore step 2 since it is negligible compared with other steps. At beginning, we have in total $2^{N_{st}+2N_p-1}$ pairs to consider. In step 1-1, we store all the ciphertext in the memory, so we need $2^{N_{st}+N_p}$ memory accesses, as well as the same amount of memory storage. After 1-1, we filter out some pairs and we are left with $2^{N_{st}} \cdot 2^{2N_p-1} \cdot 2^{-N_c}$ pairs. The rest of the process is summarized in the following Table 8.

For step 3, the complexity can be simply computed as $T_3 = 2^l \cdot 2^{k-n_k}$. And $T_{1-1} + T_{1-2} + T_{1-3} + T_{1-4} + T_3$ will be the total complexity.

Table 7. Key recovery attack in the (multiple)differential cryptanalysis scenario

Input: 2^N plaintexts and corresponding ciphertexts.

Output: Master secret key K.

1: For each structure $2^{N_{st}}$, do

1-1. Insert all the ciphertexts into a hash table indexed by N_c bits of the non-active S-boxes in the last round.

1-2. For each entry with the same N_c bit values, check if the input difference is any one of the total $|\Delta_0|$ possible input differences. If a pair satisfies one input difference, then go to the next step.

1-3. For the pairs in each entry, check whether the output differences of active S -boxes in the last round can be caused by the input difference of the previous rounds according to the differential distribution table. Go to the next step if passes.

1-4. Guess n_k bits sub keys to decrypt the ciphertext pairs to round r and check if the obtained output difference at round r is equal to Δ_r. If so, add one to the corresponding counter. 2: Choose the list of l best key candidates from the counters.

3: For each key candidate in the list, do:

3-1. Test if the corresponding key is the correct master key or not.

Table 8. Complexity evaluation for step 1

-	complexity	remaining pairs after the step
Beginning	-	$2^{N_{st}} \cdot 2^{2N_p-1}$
1-1	$T_a = 2^{N_{st}+N_p}$	$2^{N_{st}} \cdot 2^{2N_p-N_c-1}$
1-2	$T_b = 2^{N_{st}+2N_p-N_c-1}$	$\|\Delta_0\| \cdot 2^{N_{st}+N_p-N_c-1}$
1-3	$T_c = \|\Delta_0\| \cdot 2^{N_{st}+N_p-N_c-1}$	$\|\Delta_0\| \cdot 2^{N_{st}+N_p-N_c-1} \cdot p_f$
1-4	$T_d = \|\Delta_0\| \cdot 2^{N_{st}+N_p-N_c-1} \cdot p_f \cdot 2^{n_k}$	-