IET Information Security

Research Article

Improved integral attacks on 24-round LBlock and LBlock-s

ISSN 1751-8709
Received on 18th July 2019
Accepted on 4th February 2020
E-First on 30th March 2020
doi: 10.1049/iet-ifs.2019.0353
www.ietdl.org

Yaxin Cui[1]*, Hong Xu*[1] ✉*, Wenfeng Qi*[1]

[1]*Information Engineering University, Zhengzhou, 450001, People's Republic of China*

✉ *E-mail: xuhong0504@163.com*

Abstract: LBlock is a lightweight block cipher with Feistel-SP structure proposed by Wu and Zhang in Applied Cryptography and Network Security 2011, and a modified version LBlock-s is used later in the design of the lightweight authenticated encryption cipher LAC, one of the CAESAR candidates. The best known integral attack on LBlock is presented by Zhang and Wu which can attack 23-round LBlock based on a 16-round integral distinguisher found with division property. In Selected Areas in Cryptography 2018, Eskandari *et al.* further presented a 17-round integral distinguisher of LBlock with bit-based division property using SAT solver. Using their method, the authors further find some new 17-round integral distinguishers of LBlock and use one of them to present a 24-round integral attack on LBlock. Similarly, they also find some 17-round integral distinguishers of LBlock-s and select one to present a 24-round integral attack on LBlock-s. In this way, they have improved known single-key attacks on LBlock and LBlock-s by one round.

Nomenclature

M	64-bit plaintext
C	64-bit ciphertext
K	80-bit master key
K^i	32-bit sub-key of the ith round, $0 \le i \le 31$
$K^i[l]$	lth nibble of K^i, $0 \le i \le 31$, where $K^i[0]$ is the rightmost nibble
X^i	input to the ith round or the output from the $(i-1)$th round
$X^i\{j\}$	jth bit of X^i, $0 \le j \le 63$, where $X^i\{0\}$ is the rightmost bit
$X^i_{\rm L}$	left half part of X^i
$X^i_{\rm L}[l]$	lth nibble of $X^i_{\rm L}$, $0 \le l \le 7$, where $X^i_{\rm L}[0]$ is the rightmost nibble
$X^i_{\rm L}[l]\{j\}$	jth bit of $X^i_{\rm L}[l]$, $0 \le j \le 3$
$X^i_{\rm R}$	right half part of X^i
$X^i_{\rm R}[l]$	lth nibble of $X^i_{\rm R}$, $0 \le l \le 7$
$X^i_{\rm R}[l]\{j\}$	jth bit of $X^i_{\rm R}[l]$, $0 \le j \le 3$
$Y \parallel Z$	concatenation of Y and Z
$Y \lll s$	left rotation of Y by s bits
$[i]_2$	binary form of an integer i

1 Introduction

Recently, lightweight ciphers have attracted more and more attention and have been widely used in many resource-constraint environments such as radio-frequency identification tags, sensor network, and so on. LBlock [1] is a lightweight block cipher proposed by Wu and Zhang at Applied Cryptography and Network Security (ACNS) 2011. It adopts a variant of 32-round Feistel-SP structure with a 64-bit block and an 80-bit key. Many research studies have been done on LBlock for its simple structure and fine performance, and the best known single key attacks except for biclique attack on LBlock can reach 23 rounds [2–4]. In 2012, to improve the security of LBlock against biclique and related-key attacks, Wang *et al.* [5] proposed a new key schedule algorithm with better diffusion. A variant of the new algorithm, LBlock-s, was later used in the CAESAR-authenticated encryption candidate LAC [6].

The integral attack is an important cryptanalytic technique for symmetric-key primitives, which was firstly proposed by Daemen *et al.* [7] to evaluate the security of square cipher, and was later unified as an integral attack by Knudsen and Wagner [8]. An integral attack usually contains two crucial parts. First, to construct an integral distinguisher and the most widely used integral property is the balanced property. Then, to use the integral distinguisher to discard wrong sub-keys, and partial--sum [9] and meet-in-the-middle [10] techniques are widely used in the key recovery process of the integral attack to reduce the time complexity. In EUROCRYPT 2015, Todo [11] proposed the concept of division property to carefully characterise the propagation rule of integral property and used it to find new integral distinguishers of MISTY1 [12], SIMON [13], and Simeck. Later, MILP and SAT solvers were also used to find integral distinguishers based on division property [14–16].

For integral attacks on LBlock, Wu and Zhang [1] first presented an 18-round attack with a 15-round integral distinguisher, then Sasaki and Wang [17] presented an improved 22-round integral attack using meet-in-the-middle and partial-sum techniques. In INDOCRYPT 2015, Zhang and Wu [18] found a new 16-round integral distinguisher of LBlock based on division property and used it to present a 23-round integral attack on LBlock. Later in SAC 2018, Eskandari *et al.* [19] further found a new 17-round integral distinguisher of LBlock with bit-based division property using SAT solver.

Based on the method proposed by Eskandari *et al.*, we further find some new 17-round integral distinguishers of LBlock and LBlock-s. Using the meet-in-the-middle technique and the relation between sub-keys, we select two 17-round integral distinguishers of LBlock and LBlock-s, respectively, with the least number of guessed sub-key bits, and utilise them to present 24-round integral attacks on LBlock and LBlock-s.

The remainder of this paper is organised as follows. Section 2 presents brief descriptions of LBlock and LBlock-s and shows some relations among their sub-key bits. Section 3 presents the 17-round integral distinguishers of LBlock and LBlock-s. Section 4 presents the improved 24-round integral attack on LBlock, and a detailed description of the key recovery procedure is given in the Appendix. Section 5 presents the 24-round integral attack on LBlock-s. Finally, we conclude this paper in Section 6.

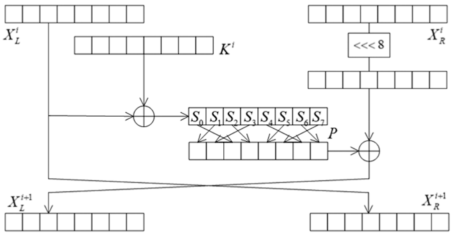

Fig. 1 *Round function of LBlock*

```
Set $(X_L^0||X_R^0) = M$, where $M$ is a 64-bit plaintext;
for $0 \le i \le 31$ do
    $X_L^{i+1} = P(S(X_L^i \oplus K^i)) \oplus (X_R^i <<< 8)$;
    $X_R^{i+1} = X_L^i$;
end for
Set $C = (X_R^{32}, X_L^{32})$ as the ciphertext.
```

Fig. 2 *Algorithm 1: specification of LBlock*

Table 1 S-boxes of LBlock

S-box	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

Table 2 Relation between adjacent sub-key bits of LBlock

t	Relation between K^i and K^{i+t}
1	$K^i[0]\{2,1,0\}, \mathscr{K}^i\{47\} \Rightarrow_S K^{i+1}[7]$
	$K^{i+1}[7] \Rightarrow_{S^{-1}} K^i[0]\{2,1,0\}$
2	$K^{i+2}[2]\{1,0\} = K^i[7]\{3,2\}, K^{i+2}[1]\{3,2\} = K^i[7]\{1,0\}$
	$K^{i+2}[1]\{1,0\} = K^i[6]\{3,2\}, K^{i+2}[0]\{3\} = K^i[6]\{1\}$
	$K^{i+2}[0]\{2\} \Rightarrow K^i[6]\{0\}, K^{i+2}[0]\{1,0\} \Rightarrow K^i[5]\{3,2\}$
3	$K^{i+3}[7] \Rightarrow_{S^{-1}} K^i[6]\{0\}, K^i[5]\{3,2,1\}$
	$K^{i+3}[6] \Rightarrow_{S^{-1}} K^i[5]\{0\}, K^i[4]\{3,2,1\}$
	$K^{i+3}[5]\{3\} = K^i[4]\{0\}, K^{i+3}[5]\{2,1,0\} = K^i[3]\{3,2,1\}$
	$K^{i+3}[4]\{3\} = K^i[3]\{0\}, K^{i+3}[4]\{2,1,0\} = K^i[2]\{3,2,1\}$
	$K^{i+3}[3]\{3\} = K^i[2]\{0\}, K^{i+3}[3]\{2,1,0\} = K^i[1]\{3,2,1\}$
	$K^{i+3}[2]\{3\} = K^i[1]\{0\}, K^{i+3}[2]\{2\} = K^i[0]\{3\}$
5	$K^{i+5}[4]\{0\} = K^i[7]\{3\}, K^{i+5}[3]\{3,2,1\} = K^i[7]\{2,1,0\}$
	$K^{i+5}[3]\{0\} = K^i[6]\{3\}, K^{i+5}[2]\{3,2\} = K^i[6]\{2,1\}$

2 Preliminaries

2.1 Notations

Throughout this paper, we use the notations listed in Nomenclature section.

2.2 Specification of LBlock

LBlock is a lightweight block cipher proposed by Wu and Zhang in ACNS 2011 which adopts a variant of Feistel-SP structure with 32 rounds. The block size is 64-bit and the key size is 80-bit. The round function of LBlock is shown in Fig. 1.

2.2.1 Encryption algorithm of LBlock: Let $(X_L^i \parallel X_R^i)$ be the input to the ith round of LBlock, and K^i be the sub-key used in the ith round, $0 \le i \le 31$, and let $(X_L^0 \parallel X_R^0) = M$ be the 64-bit plaintext. The detailed encryption procedure of LBlock is shown in Algorithm 1 (see Fig. 2), where S is the S-box layer consisting of eight different 4-bit S-boxes in parallel, and P is a permutation among eight 4-bit nibbles as shown in Fig. 1.

The contents of eight S-boxes $S_0, S_1, \dots, S_7$ used in the round function and the two S-boxes S_8 and S_9 used in the key schedule are presented in Table 1.

2.2.2 Key schedule algorithm of LBlock: The 80-bit master key K is stored in a key register and denoted as $K = k_{79}, k_{78}, \dots, k_0$. Output the leftmost 32 bits of register K as sub-key K^0. For $i = 1, 2, \dots, 31$, update the key register K as follows:

(i) $K \lll 29$

$$[k_{79}k_{78}k_{77}k_{76}] = S_9[k_{79}k_{78}k_{77}k_{76}]$$

(ii) $[k_{75}k_{74}k_{73}k_{72}] = S_8[k_{75}k_{74}k_{73}k_{72}]$

$$[k_{50}k_{49}k_{48}k_{47}k_{46}] = [k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus [i]_2$$

(iii) Output the leftmost 32 bits of the current content of register K as round sub-key K^i.

Denote by $\mathscr{K}^i = \mathscr{K}^i\{79\}, \mathscr{K}^i\{78\}, \dots, \mathscr{K}^i\{0\}$ the 80-bit internal state of the key schedule function at the ith round, $0 \le i \le 31$, then the ith round sub-key is

$$K^i = K^i[7,6,5,4,3,2,1,0] = \mathscr{K}^i\{79\}, \mathscr{K}^i\{78\}, \dots, \mathscr{K}^i\{48\}$$

2.2.3 Relation between sub-keys of LBlock: Since the generation of sub-keys of LBlock is based on the rotation of the 80-bit key register where only 13 bits or four nibbles are updated irregularly in each round, we can still find some simple relations between adjacent sub-key bits. Some of them are listed in Table 2, where $\Rightarrow$ means the latter variable can be derived from the former by XORing some constants, and $\Rightarrow_S$, $\Rightarrow_{S^{-1}}$ mean the latter variable can be derived from the former by the S-box transformation or its inverse besides XORing some constants.

2.3 Specification of LBlock-s

LBlock-s is a lightweight block cipher with a similar structure as LBlock, which is used in the lightweight authenticated encryption candidate LAC. It adopts a new key schedule with better diffusion as shown in [5, 6]. All S-boxes used in the round function and the key schedule of LBlock-s are the same as S_0 presented in Table 1. The detailed key schedule of LBlock-s is as follows:

2.3.1 Key schedule algorithm of LBlock-s: The 80-bit master key K is stored in a key register and denoted as $K = k_{79}, k_{78}, \dots, k_0$. Output the leftmost 32 bits of current content of register K as sub-key K^0, then for $i = 1, 2, \dots, 31$, update the key register K as follows:

(i) $K \lll 24$

$$[k_{55}k_{54}k_{53}k_{52}] = S[k_{79}k_{78}k_{77}k_{76}] \oplus [k_{55}k_{54}k_{53}k_{52}]$$

$$[k_{31}k_{30}k_{29}k_{28}] = S[k_{75}k_{74}k_{73}k_{72}] \oplus [k_{31}k_{30}k_{29}k_{28}]$$

(ii) $[k_{67}k_{66}k_{65}k_{64}] = [k_{71}k_{70}k_{69}k_{68}] \oplus [k_{67}k_{66}k_{65}k_{64}]$

$$[k_{51}k_{50}k_{49}k_{48}] = [k_{11}k_{10}k_9k_8] \oplus [k_{51}k_{50}k_{49}k_{48}]$$

$$[k_{54}k_{53}k_{52}k_{51}k_{50}] = [k_{54}k_{53}k_{52}k_{51}k_{50}] \oplus [i]_2$$

(iii) Output the leftmost 32 bits of the current content of register K as round sub-key K^i.

2.3.2 Relation between sub-keys of LBlock-s: Although the update of the key register of LBlock-s is related to more bits than that of LBlock, only 16 bits or four nibbles are changed in each round. Hence we can still find some simple relations between

Table 3 Relation between adjacent sub-key bits of LBlock-s

t	Relation between K^i and K^{i+t}	
1	$K^{i+1}[7] = K^i[1]$,	$K^{i+1}[6] = K^i[0]$
3	$K^{i+3}[5] = K^i[7]$,	$K^{i+3}[5] \oplus K^{i+3}[4] = K^i[6]$
	$K^{i+3}[3] = K^i[5]$,	$K^{i+3}[2] = K^i[4]$
4	$K^{i+4}[5] = K^i[1]$,	$K^{i+4}[5] \oplus K^{i+4}[4] = K^i[0]$

Table 4 Reduced division trails of S_0

Reduced division trails of S_0
$(0,0,0,0) \to \{(0,0,0,0)\}$
$(0,0,0,1) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$
$(0,0,1,0) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$
$(0,1,0,0) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$
$(1,0,0,0) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$
$(0,0,1,1) \to \{(0,0,1,1),(0,1,0,1),(1,0,0,0)\}$
$(0,1,0,1) \to \{(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$
$(0,1,1,0) \to \{(0,0,1,0),(0,1,0,0)\}$
$(1,0,0,1) \to \{(0,0,1,1),(0,1,0,0),(1,0,0,0)\}$
$(1,0,1,0) \to \{(0,0,1,1),(0,1,0,0),(1,0,0,0)\}$
$(1,1,0,0) \to \{(0,0,0,1),(0,1,0,0),(1,0,0,0)\}$
$(0,1,1,1) \to \{(0,1,1,1),(1,1,0,0)\}$
$(1,0,1,1) \to \{(0,1,0,1),(1,0,1,1)\}$
$(1,1,0,1) \to \{(0,0,1,1),(0,1,0,0),(1,0,0,0)\}$
$(1,1,1,0) \to \{(0,1,0,0),(1,0,1,0)\}$
$(1,1,1,1) \to \{(1,1,1,1)\}$

Table 5 Integral distinguishers of 17-round LBlock

Constant-bit	Balanced-bit	Corresponding nibble
{34}	{2, 3, 30, 31}	$X_R^{17}[0][7]$
{35}	{2, 3, 30, 31}	$X_R^{17}[0][7]$
{36}	{5, 7, 10, 11}	$X_R^{17}[1][2]$
{38}	{5, 7, 10, 11}	$X_R^{17}[1][2]$
{41}	{5, 7, 10, 11}	$X_R^{17}[1][2]$
{43}	{5, 7, 10, 11}	$X_R^{17}[1][2]$
{46}	{12, 14, 17, 19}	$X_R^{17}[3][4]$
{47}	{12, 14, 17, 19}	$X_R^{17}[3][4]$
{48}	{12, 14, 17, 19}	$X_R^{17}[3][4]$
{51}	{12, 14, 17, 19}	$X_R^{17}[3][4]$
{54}	{22, 23, 24, 27}	$X_R^{17}[5][6]$
{55}	{22, 23, 24, 27}	$X_R^{17}[5][6]$
{58}	{22, 23, 24, 27}	$X_R^{17}[5][6]$
{59}	{22, 23, 24, 27}	$X_R^{17}[5][6]$
{61}	{2, 3, 30, 31}	$X_R^{17}[0][7]$
{63}	{2, 3, 30, 31}	$X_R^{17}[0][7]$

adjacent sub-key bits of LBlock-s, some of them are listed in Table 3.

Some of the relations between sub-keys of LBlock or LBlock-s have also been found and used in known attacks on them [1–4, 17, 18, 20, 21]. Next, we will further use the relation to present improved integral attacks on them with new integral distinguishers.

3 Finding of new integral distinguishers of LBlock and LBlock-s

In 2015, Zhang and Wu found a new 16-round integral distinguisher of LBlock based on division property, where the state X_R^{16} is balanced when all but one bit of the plaintexts are active. With this integral distinguisher, they presented a 23-round integral attack on LBlock using the rightmost balanced nibble $X_R^{16}[0]$.

In 2018, Eskandari *et al.* proposed to use choice vectors to characterise the propagation of division property and presented a new method to efficiently find integral distinguishers of block ciphers with SAT solver. As a result, they found that there exists a 17-round integral distinguisher of LBlock with $\{\overline{34}\} \to_{17} \{2,3,30,31\}$, where only the 34th bit of the chosen plaintexts is constant, and the balanced bits appear in the leftmost two bits of $X_R^{17}[0]$ and $X_R^{17}[7]$.

Using the method proposed by Eskandari *et al.*, all valid transitions of choice vectors of **copy** operation $(a,b) \overset{\text{copy}}{\to} (a,a)$ can be given by $(0,0) \to \{(0,0)\}$, $(1,0) \to \{(1,0),(0,1)\}$, all valid transitions of choice vectors of **xor** operation $(a,b) \overset{\text{xor}}{\to} (a, a \oplus b)$ can be given by $(0,0) \to \{(0,0)\}$, $(0,1) \to \{(0,1)\}$, $(1,1) \to \{(1,1)\}$, $(1,0) \to \{(1,0),(0,1)\}$, and all valid transitions of choice vectors of an S-box can be determined by the algebraic normal form (ANF). For example, let (x_3,x_2,x_1,x_0) be the 4-bit input of S_0 and (y_3,y_2,y_1,y_0) be the corresponding output. Then the ANF of S_0 is shown as follows:

$$y_3 = 1 + x_1x_0 + x_2x_0 + x_3 + x_3x_1 + x_3x_2x_0$$

$$y_2 = 1 + x_0 + x_2x_0 + x_2x_1 + x_3 + x_3x_0$$
$$+ x_3x_2 + x_3x_2x_0 + x_3x_2x_1$$

$$y_1 = 1 + x_0 + x_2 + x_2x_0 + x_2x_1 + x_3$$

$$y_0 = x_0 + x_1 + x_2 + x_3 + x_3x_2$$

From above we know that monomial x_0 appears in y_0, y_1, and y_2, so there exist valid division trails $(0,0,0,1) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0)\}$. Since x_0 also appears in some monomials of y_3 and multiples of y_i, there also exist division trails $(0,0,0,1) \to \{(1,0,0,0),(0,0,1,1),(0,1,0,1),\ldots,(1,1,1,1)\}$. Thus the reduced division trails $(0,0,0,1) \to \{(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\}$. Similarly, all division trails of S_0 can be obtained, and the reduced division trails are shown in Table 4.

All the above transitions of different operations can be transformed into certain conjunctive normal forms. In this way, the search for integral distinguishers of LBlock can be transformed into the corresponding SAT problem and solved by SAT solver.

When only one bit of the plaintexts is constant, all the 17-round integral distinguishers of LBlock found are given in Table 5. Similarly, when only the same S-box S_0 is used, all the 17-round integral distinguishers of LBlock-s found are given in Table 6.

4 Integral attack on 24-round LBlock

Using the 17-round integral distinguishers as shown in Table 5, we will present an improved 24-round integral attack on LBlock in this section. From Table 5 we know that there exist balanced bits in any nibbles. Take the third row of Table 5 for example, when $X^{17}\{10,11\}$, i.e. $X_R^{17}[2]\{3,2\}$ are balanced, the key recovery procedure of the 24-round integral attack on LBlock is given in Fig. 3.

Similar to [17, 18], we can use the meet-in-the-middle technique to reduce the time complexity of key recovery procedure. Denote $Z^{17}[4] = S(X^{17}[5] \oplus K^{17}[5])$, since $X_R^{17}[2] = Z^{17}[4] \oplus X_L^{18}[4]$. To check whether $X_R^{17}[2]\{3,2\}$ are balanced, we need only to check whether $Z^{17}[4]\{3,2\} = X_L^{18}[4]\{3,2\}$. All the involved internal state nibbles and guessed sub-keys in the key recovery phase are shown in Fig. 3, where the internal nibbles in red square brackets are used to calculate $Z^{17}[4]$ and the internal nibbles in green round brackets are used to calculate $X_L^{18}[4]$.

Table 6 Integral distinguishers of 17-round LBlock-s

Constant-bit	Balanced-bit	Corresponding nibble
{34}	{2, 3, 30, 31}	$X_{\mathrm{R}}^{17}[0][7]$
{35}	{2, 3, 30, 31}	$X_{\mathrm{R}}^{17}[0][7]$
{38}	{6, 7, 10, 11}	$X_{\mathrm{R}}^{17}[1][2]$
{39}	{6, 7, 10, 11}	$X_{\mathrm{R}}^{17}[1][2]$
{42}	{6, 7, 10, 11}	$X_{\mathrm{R}}^{17}[1][2]$
{43}	{6, 7, 10, 11}	$X_{\mathrm{R}}^{17}[1][2]$
{46}	{14, 15, 18, 19}	$X_{\mathrm{R}}^{17}[3][4]$
{47}	{14, 15, 18, 19}	$X_{\mathrm{R}}^{17}[3][4]$
{50}	{14, 15, 18, 19}	$X_{\mathrm{R}}^{17}[3][4]$
{51}	{14, 15, 18, 19}	$X_{\mathrm{R}}^{17}[3][4]$
{54}	{22, 23, 26, 27}	$X_{\mathrm{R}}^{17}[5][6]$
{55}	{22, 23, 26, 27}	$X_{\mathrm{R}}^{17}[5][6]$
{58}	{22, 23, 26, 27}	$X_{\mathrm{R}}^{17}[5][6]$
{59}	{22, 23, 26, 27}	$X_{\mathrm{R}}^{17}[5][6]$
{62}	{2, 3, 30, 31}	$X_{\mathrm{R}}^{17}[0][7]$
{63}	{2, 3, 30, 31}	$X_{\mathrm{R}}^{17}[0][7]$

According to Fig. 3, the main time complexity of the key-recovery procedure is determined by the computation of $Z^{17}[4]$, where 80-bit sub-keys and 60-bit ciphertexts are related. Using the relation between sub-key bits as shown in Table 2 of Section 2.2, the number of sub-key bits to be guessed can be reduced from 80 to 55, and detailed relations used here are as follows:

$$K^{22}[7] \Rightarrow_{S^{-1}} K^{21}[0]\{2,1,0\}$$

$$K^{22}[0]\{2,1,0\}, \mathscr{K}^{22}\{47\} \Rightarrow_{S} K^{23}[7]$$

$$K^{23}[1]\{3,2\} = K^{21}[7]\{1,0\}$$

$$K^{23}[0]\{1,0\} \Rightarrow K^{21}[5]\{3,2\}$$

$$K^{23}[5]\{2,1,0\} \parallel K^{23}[4]\{3\} = K^{20}[3]$$

$$K^{23}[4]\{2,1,0\} \parallel K^{23}[3]\{3\} = K^{20}[2]$$

$$K^{22}[3]\{2,1,0\} \parallel K^{22}[2]\{3\} = K^{19}[1]$$

$$K^{23}[3]\{0\} = K^{18}[6]\{3\}$$

$$K^{22}[7] \Rightarrow_{S^{-1}} K^{18}[6]\{0\}$$

$$K^{22}[2]\{1,0\} \parallel K^{22}[1]\{3,2\} = K^{17}[6]\{0\} \parallel K^{17}[5]\{3,2,1\}$$

Similarly, when balanced bits appear in other nibbles, the number of sub-key bits to be guessed can also be determined. All the results are given in Table 7, where key space for $Z^{17}[j]$ means the number of sub-key bits to be guessed in calculation of $Z^{17}[j]$, and the sum keys mean the total number of sub-key bits to be guessed in the key recovery procedure.

From Table 7 we know that when the balanced bits appear in $X_{\mathrm{R}}^{17}[2]$, a fewer number of sub-key bits need to be guessed in the key recovery process. Combined with Table 5, we can select the 17-round integral distinguisher $\{\overline{36}\} \rightarrow_{17} \{5, 7, \mathbf{10}, \mathbf{11}\}$, where only the 36th bit of each chosen plaintext is constant, and the balanced bits appear in the rightmost two bits of $X_{\mathrm{R}}^{17}[2]$.

Such integral distinguisher can also be written as $\{AAAAAAC_{\{0\}}A, AAAAAAAA\} \rightarrow_{17} \{????????, ????,$ where A
$?\boldsymbol{B}_{\{3,2\}}B_{\{3,1\}}?\}$
refers to the active nibble, $C_{\{0\}}$ means the rightmost bit of the

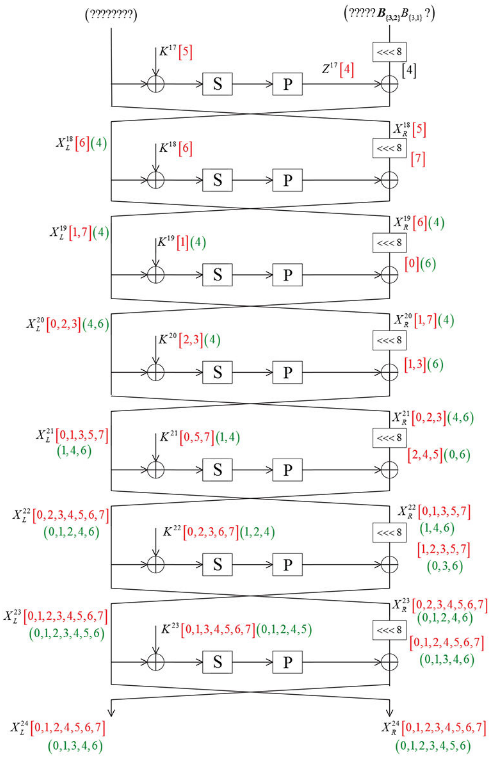

Fig. 3 *Key recovery of the 24-round integral attack on LBlock*

Table 7 Number of guessed keys for LBlock distinguishers

Balanced-bit position	Corresponding nibble	Key space for $Z^{17}[j]$	The sum keys
$X_{\mathrm{R}}^{17}[0]$	$Z^{17}[2]$	62	75
$X_{\mathrm{R}}^{17}[1]$	$Z^{17}[3]$	62	74
$X_{\mathrm{R}}^{17}[2]$	$Z^{17}[4]$	55	69
$X_{\mathrm{R}}^{17}[3]$	$Z^{17}[5]$	60	74
$X_{\mathrm{R}}^{17}[4]$	$Z^{17}[6]$	63	75
$X_{\mathrm{R}}^{17}[5]$	$Z^{17}[7]$	60	70
$X_{\mathrm{R}}^{17}[6]$	$Z^{17}[0]$	65	76
$X_{\mathrm{R}}^{17}[7]$	$Z^{17}[1]$	67	72

Table 8 Summary of computation of $\oplus Z^{17}[4]$ in 24-round integral attack of LBlock

Guessed key	bits	Texts values in the set	Data space	complexity
—	0	$X_{\rm R}^{24}[0,1,2,3,4,5,6,7] \parallel X_{\rm L}^{24}[0,1,2,4,5,6,7]$	2^{60}	2^{60}
$K^{23}[6]$	4	$X_{\rm R}^{24}[0,1,2,3,4,5,7] \parallel X_{\rm L}^{24}[0,1,2,4,5,6] \parallel X_{\rm R}^{23}[5]$	2^{56}	2^{64}
$K^{23}[4]$	8	$X_{\rm R}^{24}[0,1,2,3,5,7] \parallel X_{\rm L}^{24}[0,1,2,4,5] \parallel X_{\rm R}^{23}[4,5]$	2^{52}	2^{64}
$K^{23}[0]$	12	$X_{\rm R}^{24}[1,2,3,5,7] \parallel X_{\rm L}^{24}[0,1,4,5] \parallel X_{\rm R}^{23}[0,4,5]$	2^{48}	2^{64}
$K^{22}[0]$	16	$X_{\rm R}^{24}[1,3,5,7] \parallel X_{\rm L}^{24}[0,1,4,5] \parallel X_{\rm R}^{23}[4,5] \parallel X_{\rm R}^{22}[0]$	2^{44}	2^{64}
$K^{23}[7]$	17	$X_{\rm R}^{24}[1,3,5,7] \parallel X_{\rm L}^{24}[0,1,4] \parallel X_{\rm R}^{23}[3,4,5] \parallel X_{\rm R}^{22}[0]$	2^{44}	2^{61}
$K^{23}[1]$	21	$X_{\rm R}^{24}[1,3,5,7] \parallel X_{\rm L}^{24}[1,4] \parallel X_{\rm R}^{23}[3,4,5,6] \parallel X_{\rm R}^{22}[0]$	2^{44}	2^{65}
$K^{22}[3]$	25	$X_{\rm R}^{24}[3,5,7] \parallel X_{\rm L}^{24}[1,4] \parallel X_{\rm R}^{23}[4,5,6] \parallel X_{\rm R}^{22}[0,7]$	2^{40}	2^{69}
$K^{21}[7]$	27	$X_{\rm R}^{24}[3,5,7] \parallel X_{\rm L}^{24}[1,4] \parallel X_{\rm R}^{23}[4,6] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{21}[3]$	2^{36}	2^{67}
$K^{22}[6]$	31	$X_{\rm R}^{24}[3,5] \parallel X_{\rm L}^{24}[1,4] \parallel X_{\rm R}^{23}[4] \parallel X_{\rm R}^{22}[0,5] \parallel X_{\rm R}^{21}[3]$	2^{32}	2^{67}
$K^{21}[5]$	33	$X_{\rm R}^{24}[3,5] \parallel X_{\rm L}^{24}[1,4] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{21}[2,3]$	2^{28}	2^{65}
$K^{23}[3]$	37	$X_{\rm R}^{24}[3,5] \parallel X_{\rm L}^{24}[4] \parallel X_{\rm R}^{23}[7] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{21}[2,3]$	2^{28}	2^{65}
$K^{23}[5]$	41	$X_{\rm R}^{24}[3,5] \parallel X_{\rm R}^{23}[2,7] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{21}[2,3]$	2^{28}	2^{69}
$K^{22}[7]$	45	$X_{\rm R}^{24}[3] \parallel X_{\rm R}^{23}[2] \parallel X_{\rm R}^{22}[0,3] \parallel X_{\rm R}^{21}[2,3]$	2^{24}	2^{73}
$K^{20}[2]$	45	$X_{\rm R}^{24}[3] \parallel X_{\rm R}^{23}[2] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{21}[3] \parallel X_{\rm R}^{20}[1]$	2^{20}	2^{69}
$K^{22}[2]$	49	$X_{\rm R}^{23}[2] \parallel X_{\rm R}^{22}[0,1] \parallel X_{\rm R}^{21}[3] \parallel X_{\rm R}^{20}[1]$	2^{20}	2^{69}
$K^{20}[3]$	49	$X_{\rm R}^{23}[2] \parallel X_{\rm R}^{22}[0] \parallel X_{\rm R}^{20}[1,7]$	2^{16}	2^{69}
$K^{21}[0]$	50	$X_{\rm R}^{21}[0] \parallel X_{\rm R}^{20}[1,7]$	2^{12}	2^{66}
$K^{19}[1]$	50	$X_{\rm R}^{20}[7] \parallel X_{\rm R}^{19}[6]$	2^{8}	2^{62}
$K^{18}[6]$	52	$X_{\rm R}^{18}[5]$	2^{4}	2^{60}
$K^{17}[5]$	55	$Z^{17}[4]$	2^{4}	2^{59}

nibble is constant and other bits are active, $B_{\{3,2\}}$ means the third and the second bits of the nibble are balanced.

4.1 Key recovery

We can present a 24-round integral attack on LBlock by appending seven rounds to the above 17-round integral distinguisher as shown in Fig. 3. The main key recovery phase is as follows:

(i) Choose 2^{63} plaintexts that are constant at the 36th bit and active at the remaining bits.
(ii) Compute $\oplus Z^{17}[4]$ by guessing the 55-bit key.
(iii) Compute $\oplus X_{\rm L}^{18}[4]$ by guessing the 47-bit key independently.
(iv) Find matches between the two results, and get the corresponding 69-bit key as key candidates. By judging whether $\oplus Z^{17}[4]\{3,2\} = \oplus X_{\rm L}^{18}[4]\{3,2\}$, we could obtain 2^{67} key candidates.
(v) For 2^{67} key candidates, we exhaustively search the remaining 11-bit key to recover the master key.

A detailed computation procedure and sub-key relation is given in the Appendix. A summary of computation of $\oplus Z^{17}[4]$ in step 2 is given in Table 8, where 60-bit ciphertexts are related and totally 55-bit sub-keys are guessed. More specifically, the time complexity for each step is estimated as the product of the previous data size and the total number of guessed bits. Thus the total complexity to compute $\oplus Z^{17}[4]$ is almost

$$(4 \times 2^{64} + 2^{61} + 3 \times 2^{65} + 5 \times 2^{69} + 2 \times 2^{67} + 2^{73}$$
$$+2^{66} + 2^{62} + 2^{60} + 2^{59}) \times \frac{1}{8} \times \frac{1}{24} \simeq 2^{73.46} \times \frac{1}{8} \times \frac{1}{24} \simeq 2^{65.87}$$

24-round encryptions of LBlock. The procedure to compute $\oplus X_{\rm L}^{18}[4]$ is similar, where only 48-bit sub-keys should be guessed and the time complexity is almost $2^{51.04}$ 24-round encryptions.

From the above analysis, we know that the time complexity of the key recovery procedure is determined by exhaustively searching in step 5. Thus the time complexity of the attack is almost $2^{67} \times 2^{11} = 2^{78}$ 24-round encryptions, where 2^{63} chosen plaintexts are used, and $2^{60} \times 15 \times 4 \times 2^{-3} = 2^{61}$ bytes of memory is needed to save 15 nibbles of ciphertexts involved in the computation of $X_{\rm R}^{17}[2]$.

5 Integral attack on 24-round LBlock-s

Similar to above, using the 17-round integral distinguishers shown in Table 6 of Section 3, we can present an integral attack on 24-round LBlock-s as follows.

Take the first row of Table 5 for example, when $X^{17}\{2,3\}$, i.e. $X_{\rm R}^{17}[0]\{3,2\}$ are balanced, the key recovery procedure of the 24-round integral attack on LBlock-s is given in Fig. 4. Denote $Z^{17}[2] = S(X^{17}[0] \oplus K^{17}[0])$, since $X_{\rm R}^{17}[0] = Z^{17}[2] \oplus X_{\rm L}^{18}[2]$. We need only to check whether $\oplus Z^{17}[2]\{3,2\} = \oplus X_{\rm L}^{18}[2]\{3,2\}$. All the involved internal state nibbles and guessed sub-keys in the key recovery phase are shown in Fig. 4, where the internal nibbles in red square brackets are used to calculate $Z^{17}[2]$ and the internal nibbles in green round brackets are used to calculate $X_{\rm L}^{18}[2]$.

According to Fig. 4, the main time complexity of the key-recovery procedure is determined by the computation of $Z^{17}[2]$, where 80-bit sub-keys and 60-bit ciphertexts are related. Using the relation between sub-key bits as shown in Table 3 of Section 2.3, the number of sub-key bits to be guessed can be reduced from 80 to 56. In the following, we list utilised relations in detail

$$K^{22}[0] = K^{23}[6], K^{21}[0] = K^{22}[6]$$

$$K^{20}[0] = K^{21}[6], K^{20}[5] = K^{23}[3]$$

$$K^{19}[0] = K^{23}[5] \oplus K^{23}[4], K^{18}[0] = K^{22}[5] \oplus K^{22}[4]$$

When balanced bits appear in other nibbles, the number of sub-key bits to be guessed can also be determined by the same method. All the results are given in Table 9. In this case, we choose the balanced bits appearing in $X_{\rm R}^{17}[0]$, where a fewer number of sub-key bits needed to be guessed in the key recovery process. Combined with Table 6, we select the best 17-round integral

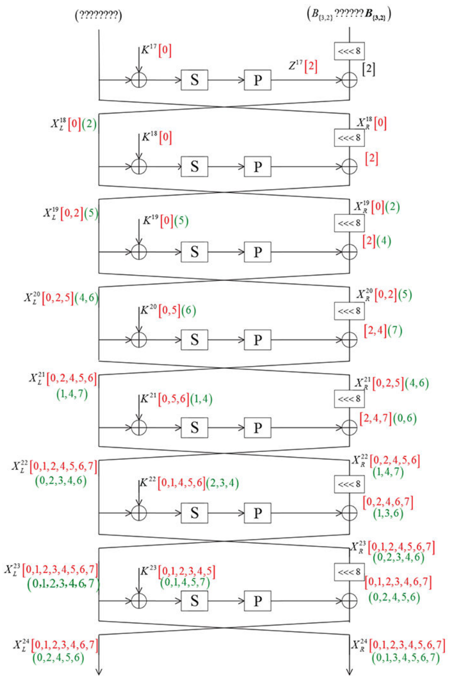
(????????)
$(B_{\{3,2\}}??????\boldsymbol{B}_{\{3,2\}})$
$K^{17}[0]$
$Z^{17}[2]$
<<< 8
[2]
$X_L^{18}[0](2)$
$X_R^{18}[0]$
$K^{18}[0]$
[2]
$X_L^{19}[0,2](5)$
$X_R^{19}[0](2)$
$K^{19}[0](5)$
[2](4)
$X_L^{20}[0,2,5](4,6)$
$X_R^{20}[0,2](5)$
$K^{20}[0,5](6)$
[2,4](7)
$X_L^{21}[0,2,4,5,6]$ (1,4,7)
$X_R^{21}[0,2,5](4,6)$
$K^{21}[0,5,6](1,4)$
[2,4,7](0,6)
$X_L^{22}[0,1,2,4,5,6,7]$ (0,2,3,4,6)
$X_R^{22}[0,2,4,5,6]$ (1,4,7)
$K^{22}[0,1,4,5,6](2,3,4)$
[0,2,4,6,7] (1,3,6)
$X_L^{23}[0,1,2,3,4,5,6,7]$ (0,1,2,3,4,6,7)
$X_R^{23}[0,1,2,4,5,6,7]$ (0,2,3,4,6)
$K^{23}[0,1,2,3,4,5]$ (0,1,4,5,7)
[0,1,2,3,4,6,7] (0,2,4,5,6)
S
P
$X_L^{24}[0,1,2,3,4,6,7]$ (0,2,4,5,6)
$X_R^{24}[0,1,2,3,4,5,6,7]$ (0,1,3,4,5,6,7)

Fig. 4 *Key recovery of the 24-round integral attack on LBlock-s*

Table 9 Number of guessed sub-key bits for different distinguishers of LBlock-s

Balanced-bit position	Corresponding nibble	Key space for $Z^{17}[j]$	The sum keys
$X_R^{17}[0]$	$Z^{17}[2]$	56	68
$X_R^{17}[1]$	$Z^{17}[3]$	60	76
$X_R^{17}[2]$	$Z^{17}[4]$	60	64
$X_R^{17}[3]$	$Z^{17}[5]$	64	76
$X_R^{17}[4]$	$Z^{17}[6]$	60	68
$X_R^{17}[5]$	$Z^{17}[7]$	64	72
$X_R^{17}[6]$	$Z^{17}[0]$	60	72
$X_R^{17}[7]$	$Z^{17}[1]$	72	80

distinguisher $\{\overline{34}\} \rightarrow_{17} \{\mathbf{2}, \mathbf{3}, 30, 31\}$, where only the 34th bit of each chosen plaintext is constant, and the balanced bits appear in the leftmost two bits of $X_R^{17}[0]$.

Such an integral distinguisher can also be written as $\{AAAAAAAC_{\{2\}}, AAAAAAAA\} \rightarrow_{17} \{????????, B_{\{3,2\}}?????, ?\boldsymbol{B}_{\{3,2\}}\}$
where A refers to the active nibble, $C_{\{2\}}$ means the second bit of the nibble is constant and other bits are active, $B_{\{3,2\}}$ means the leftmost two bits of the nibble are balanced.

5.1 Key recovery

We can present a 24-round integral attack on LBlock-s by appending seven rounds to the above 17-round integral distinguisher as shown in Fig. 4. The main key recovery phase is as follows:

(i) Choose 2^{63} plaintexts that are constant at the 34th bit and active at the remaining bits.

(ii) Compute $\oplus Z^{17}[2]$ by guessing the 56-bit key.

(iii) Compute $\oplus X_L^{18}[2]$ by guessing the 40-bit key independently.

(iv) Find matches between two results, then get the corresponding 68-bit key as key candidates. By judging whether $\oplus Z^{17}[2]\{3,2\}$ equal to $\oplus X_L^{18}[2]\{3,2\}$, respectively, we could obtain 2^{66} key candidates.

(v) For 2^{66} key candidates, we exhaustively search the remaining 12-bit key to recover the master key.

A summary of computation of $\oplus X_L^{18}[2]$ in step 2 is given in Table 10, where 60-bit ciphertexts are related and totally 56-bit sub-keys are guessed, the total time complexity to compute $\oplus X_L^{18}[2]$ is almost

$$(7 \times 2^{64} + 6 \times 2^{72} + 5 \times 2^{68} + 2 \times 2^{60}) \times \frac{1}{8} \times \frac{1}{24}$$

$$\simeq 2^{74.66} \times \frac{1}{8} \times \frac{1}{24} \simeq 2^{67.08}$$

24-round encryptions of LBlock-s.

From the above analysis, we know that the time complexity of the key recovery procedure is also determined by exhaustively searching in step 5. Thus the time complexity of the attack is almost $2^{66} \times 2^{12} = 2^{78}$ 24-round encryptions, where 2^{63} chosen plaintexts are used, and $2^{60} \times 15 \times 4 \times 2^{-3} = 2^{61}$ bytes of memory is needed to save 15 nibbles of ciphertexts involved in the computation of $X_R^{17}[0]$.

6 Conclusion

LBlock and LBlock-s are two lightweight block ciphers with a similar structure and different key schedules. The best known single-key attacks on LBlock and LBlock-s can only reach 23 rounds. By carefully analysing the division property of S-boxes, we find new 17-round integral distinguishers with Eskandari *et al.*'s method. Using the meet-in-the-middle technique and the relation between sub-keys, we select two best 17-round integral distinguishers and utilise them to present 24-round integral attacks

Table 10 Summary of computation of $\oplus Z^{17}[2]$ in 24-round integral attack of LBlock-s

Guessed key	Bits	Texts values in the set	Data space	complexity
—	0	$X_R^{24}[0,1,2,3,4,5,6,7] \parallel X_L^{24}[0,1,2,3,4,6,7]$	2^{60}	2^{60}
$K^{23}[3]$	4	$X_R^{24}[0,1,2,4,5,6,7] \parallel X_L^{24}[0,2,3,4,6,7] \parallel X_R^{23}[7]$	2^{56}	2^{64}
$K^{23}[5]$	8	$X_R^{24}[0,1,2,4,6,7] \parallel X_L^{24}[0,2,3,6,7] \parallel X_R^{23}[2,7]$	2^{52}	2^{64}
$K^{23}[1]$	12	$X_R^{24}[0,2,4,6,7] \parallel X_L^{24}[2,3,6,7] \parallel X_R^{23}[2,6,7]$	2^{48}	2^{64}
$K^{22}[6]$	16	$X_R^{24}[0,2,4,6] \parallel X_L^{24}[2,3,6,7] \parallel X_R^{23}[2,7] \parallel X_R^{22}[5]$	2^{44}	2^{64}
$K^{23}[0]$	20	$X_R^{24}[0,2,4,6] \parallel X_L^{24}[3,6,7] \parallel X_R^{23}[0,2,7] \parallel X_R^{22}[5]$	2^{44}	2^{64}
$K^{23}[2]$	24	$X_R^{24}[0,2,4,6] \parallel X_L^{24}[6,7] \parallel X_R^{23}[0,1,2,7] \parallel X_R^{22}[5]$	2^{44}	2^{68}
$K^{22}[1]$	28	$X_R^{24}[2,4,6] \parallel X_L^{24}[6,7] \parallel X_R^{23}[0,2,7] \parallel X_R^{22}[5,6]$	2^{40}	2^{72}
$K^{22}[0]$	32	$X_R^{24}[4,6] \parallel X_L^{24}[6,7] \parallel X_R^{23}[2,7] \parallel X_R^{22}[0,5,6]$	2^{36}	2^{72}
$K^{21}[0]$	32	$X_R^{24}[4,6] \parallel X_L^{24}[6,7] \parallel X_R^{23}[7] \parallel X_R^{22}[5,6] \parallel X_R^{21}[0]$	2^{32}	2^{68}
$K^{21}[6]$	36	$X_R^{24}[4,6] \parallel X_L^{24}[6,7] \parallel X_R^{22}[5] \parallel X_R^{21}[0,5]$	2^{28}	2^{68}
$K^{23}[6]$	36	$X_R^{24}[4,6] \parallel X_L^{24}[6] \parallel X_R^{23}[5] \parallel X_R^{22}[5] \parallel X_R^{21}[0,5]$	2^{28}	2^{64}
$K^{23}[4]$	40	$X_R^{24}[4,6] \parallel X_R^{23}[4,5] \parallel X_R^{22}[5] \parallel X_R^{21}[0,5]$	2^{28}	2^{68}
$K^{22}[5]$	44	$X_R^{24}[6] \parallel X_R^{23}[4] \parallel X_R^{22}[2,5] \parallel X_R^{21}[0,5]$	2^{24}	2^{72}
$K^{22}[4]$	48	$X_R^{23}[4] \parallel X_R^{22}[2,4,5] \parallel X_R^{21}[0,5]$	2^{24}	2^{72}
$K^{20}[5]$	48	$X_R^{23}[4] \parallel X_R^{22}[2,5] \parallel X_R^{21}[0] \parallel X_R^{20}[2]$	2^{20}	2^{72}
$K^{21}[5]$	52	$X_R^{22}[2] \parallel X_R^{21}[0,2] \parallel X_R^{20}[2]$	2^{16}	2^{72}
$K^{20}[0]$	52	$X_R^{21}[2] \parallel X_R^{20}[0,2]$	2^{12}	2^{68}
$K^{19}[0]$	52	$X_R^{20}[2] \parallel X_R^{19}[0]$	2^{8}	2^{64}
$K^{18}[0]$	52	$X_R^{18}[0]$	2^{4}	2^{60}
$K^{17}[0]$	56	$Z^{17}[2]$	2^{4}	2^{60}

on LBlock and LBlock-s. This is the longest round that can be attacked by single-key attacks except for biclique attacks.

7 Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (U1536101, 61521003) and the National Cryptography Development Fund of China (MMJJ20180204, MMJJ20170103).

8 References

[1] Wu, W., Zhang, L.: ‘LBlock: a lightweight block cipher’. 9th Int. Conf. on Applied Cryptography and Network Security – ACNS 2011, Nerja, Spain, June 2011, pp. 327–344

[2] Wang, Y., Wu, W.: ‘Improved multidimensional zero-correlation linear cryptanalysis and applications to LBlock and TWINE’. 19th Australasian Conf. on Information Security and Privacy – ACISP 2014, Wollongong, NSW, Australia, July 2014, pp. 1–16

[3] Boura, C., Naya-Plasencia, M., Suder, V.: ‘Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and SIMON’. Proc. Advances in Cryptology – ASIACRYPT 2014, Kaoshiung, Taiwan, R.O.C., December 2014, pp. 179–199

[4] Lin, L., Wu, W., Zheng, Y.: ‘Automatic search for key-bridging technique: applications to LBlock and TWINE’. 23rd Int. Workshop on Fast Software Encryption – FSE 2016, Bochum, Germany, March 2016, pp. 247–267

[5] Wang, Y., Wu, W., Yu, X., *et al.*: ‘Security on LBlock against Biclique cryptanalysis’. 13th Int. Workshop on Information Security Applications – WISA 2012, Jeju Island, Korea, 16–18 August 2012, pp. 1–14

[6] Zhang, L., Wu, W., Wang, Y., *et al.*: ‘LAC: A lightweight authenticated encryption cipher’. Submitted to CAESAR. Available at http://competitions.cr.yp.to/round1/lacv1.pdf, Version 1, 15 March 2014

[7] Daemen, J., Knudsen, L.R., Rijmen, V.: ‘The block cipher SQUARE’. 4th Int. Workshop on Fast Software Encryption, FSE 1997, Haifa, Israel, 20–22 January 1997, pp. 149–165

[8] Knudsen, L.R., Wagner, D.A.: ‘Integral cryptanalysis’. 9th Int. Workshop on Fast Software Encryption, FSE 2002, Leuven, Belgium, February 2002, pp. 112–127

[9] Ferguson, N., Kelsey, J., Lucks, S., *et al.*: ‘Improved cryptanalysis of Rijndael’. 7th Int. Workshop on Fast Software Encryption, FSE 2000, New York, NY, USA, April 2000, pp. 213–230

[10] Sasaki, Y., Wang, L.: ‘Meet-in-the-middle technique for integral attacks against Feistel ciphers’. 19th Int. Conf. on Selected Areas in Cryptography, SAC 2012, Windsor, ON, Canada, 15–16 August 2012, pp. 234–251

[11] Todo, Y.: ‘Structural evaluation by generalized integral property’. Proc. Advances in Cryptology – EUROCRYPT 2015, Sofia, Bulgaria, April 2015, pp. 287–314

[12] Todo, Y.: ‘Integral cryptanalysis on full MISTY1’, *J. Cryptol.*, 2017, **30**, (3), pp. 920–959

[13] Todo, Y., Morii, M.: ‘Bit-based division property and application to simon family’. 23rd Int. Workshop on Fast Software Encryption – FSE 2016, Bochum, Germany, March 2016, pp. 357–377

[14] Xiang, Z., Zhang, W., Bao, Z., *et al.*: ‘Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers’. Proc. Advances in Cryptology – ASIACRYPT 2016, Hanoi, Vietnam, December 2016, pp. 648–678

[15] Sun, L., Wang, W., Liu, R., *et al.*: ‘MILP-aided bit-based division property for ARX ciphers’, *Sci. China Inf. Sci.*, 2018, **61**, (11), pp. 118102:1–118102:3

[16] Sun, L., Wang, W., Wang, M.: ‘Automatic search of bit-based division property for ARX ciphers and word-based division property’. Proc. Advances in Cryptology – ASIACRYPT 2017, Hong Kong, China, December 2017, pp. 128–157

[17] Sasaki, Y., Wang, L.: ‘Comprehensive study of integral analysis on 22-round lblock’. 15th Int. Conf. on Information Security and Cryptology – ICISC 2012, Seoul, Korea, November 2012, pp. 156–169

[18] Zhang, H., Wu, W.: ‘Structural evaluation for generalized Feistel structures and applications to LBlock and TWINE’. Progress in Cryptology – INDOCRYPT 2015 – 16th Int. Conf. on Cryptology in India, Bangalore, India, December 2015, pp. 218–237

[19] Eskandari, Z., Kidmose, A.B., Kölbl, S., *et al.*: ‘Finding integral distinguishers with ease’. 25th Int. Conf. on Selected Areas in Cryptography – SAC 2018, Calgary, AB, Canada, August 2018, pp. 115–138

[20] Xu, H., Jia, P., Huang, G., *et al.*: ‘Multidimensional zero-correlation linear cryptanalysis on 23-round LBlock-s’. 17th Int. Conf. on Information and Communications Security – ICICS 2015, Beijing, China, 9–11 December 2015, pp. 97–108

[21] Jia, P., Xu, H., Lai, X.: ‘Impossible differential cryptanalysis of reduced round LBlock-s’, *Acta Electron. Sin.*, 2017, **45**, (4), pp. 966–973 (in Chinese)

9 Appendix: Computation procedure of ⊕Z17[4]

We need to guess the 55-bit key to compute the value of $\oplus Z^{17}[4]$ according to the key schedule. The procedure is as follows.

(i) Query 2^{63} plaintexts which are constant at one bit and are active at other bits.
(ii) Count whether each 15-nibble value $X_L^{24}[0,1,2,4,5,6,7] \parallel X_R^{24}[0,1,2,3,4,5,6,7]$ appears even or odd times, and pick the values which appear odd times.

(iii) Guess four bits of $K^{23}[6]$, and compute $X_{\mathrm{R}}^{23}[5]$ with $X_{\mathrm{L}}^{24}[7], X_{\mathrm{R}}^{24}[6]$. Compress the data into 2^{56} texts of the value of $X_{\mathrm{L}}^{24}[0,1,2,4,5,6] \parallel X_{\mathrm{R}}^{24}[0,1,2,3,4,5,7] \parallel X_{\mathrm{R}}^{23}[5]$.

(iv) Guess four bits of $K^{23}[4]$, and compute $X_{\mathrm{R}}^{23}[4]$ with $X_{\mathrm{L}}^{24}[6], X_{\mathrm{R}}^{24}[4]$. Compress the data into 2^{52} texts of the value of $X_{\mathrm{L}}^{24}[0,1,2,4,5] \parallel X_{\mathrm{R}}^{24}[0,1,2,3,5,7] \parallel X_{\mathrm{R}}^{23}[4,5]$.

(v) Guess four bits of $K^{23}[0]$, and compute $X_{\mathrm{R}}^{23}[0]$ with $X_{\mathrm{L}}^{24}[2], X_{\mathrm{R}}^{24}[0]$. Compress the data into 2^{48} texts of the value of $X_{\mathrm{L}}^{24}[0,1,4,5] \parallel X_{\mathrm{R}}^{24}[1,2,3,5,7] \parallel X_{\mathrm{R}}^{23}[0,4,5]$ appearing odd times.

(vi) Guess four bits of $K^{22}[0]$, and compute $X_{\mathrm{R}}^{22}[0]$ with $X_{\mathrm{L}}^{23}[2], X_{\mathrm{R}}^{23}[0]$. Compress the data into 2^{44} texts of the value of $X_{\mathrm{L}}^{24}[0,1,4,5] \parallel X_{\mathrm{R}}^{24}[1,3,5,7] \parallel X_{\mathrm{R}}^{23}[4,5] \parallel X_{\mathrm{R}}^{22}[0]$ appearing odd times.

(vii) Owing to the key schedule, $K^{23}[7]$ is determined by rightmost three bits in $K^{22}[0]$. Then we only need to guess the rightmost one bit in $K^{23}[7]$. Compute $X_{\mathrm{R}}^{23}[3]$ with $X_{\mathrm{L}}^{24}[5], X_{\mathrm{R}}^{24}[7]$. Compress the data into 2^{44} texts of the value of $X_{\mathrm{L}}^{24}[0,1,4] \parallel X_{\mathrm{R}}^{24}[1,3,5,7] \parallel X_{\mathrm{R}}^{23}[3,4,5] \parallel X_{\mathrm{R}}^{22}[0]$.

(viii) Guess four bits of $K^{23}[1]$, and compute $X_{\mathrm{R}}^{23}[6]$ with $X_{\mathrm{L}}^{24}[0], X_{\mathrm{R}}^{24}[1]$. Compress the data into 2^{44} texts of the value of $X_{\mathrm{L}}^{24}[1,4] \parallel X_{\mathrm{R}}^{24}[1,3,5,7] \parallel X_{\mathrm{R}}^{23}[3,4,5,6] \parallel X_{\mathrm{R}}^{22}[0]$.

(ix) Guess four bits of $K^{22}[3]$, and compute $X_{\mathrm{R}}^{22}[7]$ with $X_{\mathrm{L}}^{23}[1], X_{\mathrm{R}}^{23}[3]$. Compress the data into 2^{40} texts of the value of $X_{\mathrm{L}}^{24}[1,4] \parallel X_{\mathrm{R}}^{24}[3,5,7] \parallel X_{\mathrm{R}}^{23}[4,5,6] \parallel X_{\mathrm{R}}^{22}[0,7]$.

(x) Owing to the key schedule, $K^{21}[7]$ is determined by leftmost two bits in $K^{23}[1]$. Then we only need to guess the leftmost two bit in $K^{21}[7]$. Compute $X_{\mathrm{R}}^{21}[3]$ with $X_{\mathrm{L}}^{22}[5], X_{\mathrm{R}}^{22}[7]$. Compress the data into 2^{36} texts of the value of $X_{\mathrm{L}}^{24}[1,4] \parallel X_{\mathrm{R}}^{24}[3,5,7] \parallel X_{\mathrm{R}}^{23}[4,6] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{21}[3]$ appearing odd times.

(xi) Guess four bits of $K^{22}[6]$, and compute $X_{\mathrm{R}}^{22}[5]$ with $X_{\mathrm{L}}^{23}[7], X_{\mathrm{R}}^{23}[6]$. Compress the data into 2^{32} texts of the value of $X_{\mathrm{R}}^{24}[1,4] \parallel X_{\mathrm{R}}^{24}[3,5] \parallel X_{\mathrm{R}}^{23}[4] \parallel X_{\mathrm{R}}^{22}[0,5] \parallel X_{\mathrm{R}}^{21}[3]$.

(xii) Owing to the key schedule, $K^{21}[5]$ is determined by rightmost two bits in $K^{23}[0]$. Then we only need to guess the rightmost two bits in $K^{21}[5]$. Compute $X_{\mathrm{R}}^{21}[2]$ with $X_{\mathrm{L}}^{22}[4], X_{\mathrm{R}}^{22}[5]$. Compress the data into 2^{28} texts of the value of $X_{\mathrm{R}}^{24}[1,4] \parallel X_{\mathrm{R}}^{24}[3,5] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{21}[2,3]$.

(xiii) Guess four bits of $K^{23}[3]$, and compute $X_{\mathrm{R}}^{23}[7]$ with $X_{\mathrm{L}}^{24}[1], X_{\mathrm{R}}^{24}[3]$. Compress the data into 2^{28} texts of the value of $X_{\mathrm{R}}^{24}[4] \parallel X_{\mathrm{R}}^{24}[3,5] \parallel X_{\mathrm{R}}^{23}[7] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{21}[2,3]$.

(xiv) Guess four bits of $K^{23}[5]$, and compute $X_{\mathrm{R}}^{23}[2]$ with $X_{\mathrm{L}}^{24}[4], X_{\mathrm{R}}^{24}[5]$. Compress the data into 2^{28} texts of the value of $X_{\mathrm{R}}^{24}[3,5] \parallel X_{\mathrm{R}}^{23}[2,7] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{21}[2,3]$.

(xv) Guess four bits of $K^{22}[7]$, and compute $X_{\mathrm{R}}^{22}[3]$ with $X_{\mathrm{L}}^{23}[5], X_{\mathrm{R}}^{23}[7]$. Compress the data into 2^{24} texts of the value of $X_{\mathrm{R}}^{24}[3] \parallel X_{\mathrm{R}}^{23}[2] \parallel X_{\mathrm{R}}^{22}[0,3] \parallel X_{\mathrm{R}}^{21}[2,3]$.

(xvi) Owing to the key schedule, $K^{20}[2]$ is determined by rightmost three bits in $K^{23}[4]$ and leftmost one bit of $K^{23}[3]$. Compute $X_{\mathrm{R}}^{20}[1]$ with $X_{\mathrm{L}}^{21}[3], X_{\mathrm{R}}^{21}[2]$. Compress the data into 2^{20} texts of the value of $X_{\mathrm{R}}^{24}[3] \parallel X_{\mathrm{R}}^{23}[2] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{21}[3] \parallel X_{\mathrm{R}}^{20}[1]$.

(xvii) Guess four bits of $K^{22}[2]$, and compute $X_{\mathrm{R}}^{22}[1]$ with $X_{\mathrm{L}}^{23}[3], X_{\mathrm{R}}^{23}[2]$. Compress the data into 2^{20} texts of the value of $X_{\mathrm{R}}^{23}[2] \parallel X_{\mathrm{R}}^{22}[0,1] \parallel X_{\mathrm{R}}^{21}[3] \parallel X_{\mathrm{R}}^{20}[1]$.

(xviii) Owing to the key schedule, $K^{20}[3]$ is determined by rightmost three bits in $K^{23}[5]$ and leftmost one bit of $K^{23}[4]$. Compute $X_{\mathrm{R}}^{20}[7]$ with $X_{\mathrm{L}}^{21}[1], X_{\mathrm{R}}^{21}[3]$. Compress the data into 2^{16} texts of the value of $X_{\mathrm{R}}^{23}[2] \parallel X_{\mathrm{R}}^{22}[0] \parallel X_{\mathrm{R}}^{20}[1,7]$.

(xix) Owing to the key schedule, $K^{21}[0]$ is determined by leftmost three bits in $K^{22}[7]$. Then we only need to guess the leftmost one bit in $K^{21}[0]$. Compute $X_{\mathrm{R}}^{21}[0]$ with $X_{\mathrm{L}}^{22}[2], X_{\mathrm{R}}^{22}[0]$. Compress the data into 2^{16} texts of the value of $X_{\mathrm{R}}^{21}[0] \parallel X_{\mathrm{R}}^{20}[1,7]$.

(xx) Owing to the key schedule, $K^{19}[1]$ is determined by rightmost three bits in $K^{22}[3]$ and leftmost one bit of $K^{22}[2]$. Compute $X_{\mathrm{R}}^{19}[6]$ with $X_{\mathrm{L}}^{20}[0], X_{\mathrm{R}}^{20}[1]$. Compress the data into 2^{8} texts of the value of $X_{\mathrm{R}}^{20}[7] \parallel X_{\mathrm{R}}^{19}[6]$.

(xxi) Owing to the key schedule, $K^{18}[6]$ is determined by rightmost one bit in $K^{23}[3]$ and one bit of $K^{23}[2]$. Compute $X_{\mathrm{R}}^{18}[5]$ with $X_{\mathrm{L}}^{19}[7], X_{\mathrm{R}}^{19}[6]$. Compress the data into 2^{4} texts of the value of $X_{\mathrm{R}}^{18}[5]$.

(xxii) Owing to the key schedule, $K^{17}[5]$ is determined by rightmost one bits in $K^{22}[2]$. Then we only need to guess rightmost three bits of $K^{17}[5]$. Compute $S(X_{\mathrm{L}}^{17}[5] \oplus K^{17}[5])$. Eventually, compute the values appearing odd time and get the sum $\oplus\,(S(X_{\mathrm{L}}^{17}[5] \oplus K^{17}[5]))$.