JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 805-819 (2022)
DOI: 10.6688/JISE.202207_38(4).0007

Impossible Differential Cryptanalysis of BORON

XIAO-NIAN WU[1], YING-XIN LI[1], LING-CHEN LI[1],
YONG-ZHUANG WEI[1,2] AND RUN-LIAN ZHANG[1]
[1]*Guangxi Key Laboratory of Cryptography and Information Security*
Guilin University of Electronic Technology
Guangxi Guilin, 541004 P.R. China
[2]*State Key Laboratory of Cryptology*
P.O. Box 5159, Beijing, 100878 P.R. China
E-mail: {xnwu; walker_wyz; zhangrl}@guet.edu.cn;
liyx1140@163.com; lilingchen601@126.com

BORON as a novel ultra-lightweight block cipher has some favorable properties, *e.g.* faster encryption speed and particular low power encryption design for pervasive computing and so on. Currently, the security of BORON encryption algorithm has extensively received attention. However, it still appears to be an interesting and crucial task to determine BORON encryption algorithm against impossible differential attack. In this article, the security of BORON encryption algorithm against the impossible differential attack is deeply investigated. In the first place, basing on both the differential property of the S-box and the algorithm structure of BORON cipher, 160 new 6-round impossible differential distinguishers are attained by using the MILP (Mixed-Integer Linear Programming) automated search model. Moreover, an impossible differential attack on 10-round reduced BORON cipher is proposed, where the 6-round impossible differential distinguishers is further extended forward 2-round and backward 2-round respectively. It is illustrated that this new attack requires about $2^{43.52}$ chosen data, $2^{62.08}$ 10-round encryption operations, and $2^{44.52}$ bits memory. Compared with the previous results, this attack achieves the highest attacking round number and the lowest data complexity.

Keywords: BORON, impossible differential cryptanalysis, MILP (mixed-integer linear programming), S-box, lightweight block cipher

1. INTRODUCTION

Lightweight block cipher can provide extensive security protection for the devices of the resource-constrained environments, such as Internet of things, WSNs, *etc*. During the past decade, many lightweight block ciphers are proposed, for instance LBlock [1], PRESENT [2], RECTANGLE [3], *etc*. Accordingly, security analysis for the lightweight block ciphers is becoming more and more important. Normally, a new proposed lightweight block cipher needs to be verified its security and its implementation performance. In particular, the designers have to clearly determine the ability of lightweight block cipher against traditional attacks, *e.g.*, differential cryptanalysis [4], linear cryptanalysis [5], impossible differential cryptanalysis [6, 7], and so on.

Received August 7, 2020; revised May 5 & June 28 2021; accepted August 31, 2021.
Communicated by Fu-Hau Hsu.

Impossible differential cryptanalysis as one of the most effective attacks on block ciphers is independently proposed by Knudsen [6] and Biham [7]. Generally, impossible differential cryptanalysis contains two phases, *i.e.*, both the construction of impossible differential distinguisher phase and the key recovery phase. Notice that there are some new advanced search methods to quickly look for the impossible differential distinguisher, *e.g.* the Mixed Integer Linear Programming (MILP) method. More specifically, At ICISC 2011, MILP technique for counting the number of active S-boxes of the word-oriented ciphers is used to evaluate their resistance against differential and linear cryptanalysis [8]. At ASIACRYPT 2014, the MILP model regarding to the related-key differential characteristics for the bit-oriented block ciphers was presented [9]. In 2016, the automatic MILP tool regarding to search impossible differential distinguisher for the ARX ciphers is proposed in [10]. Later, at EUROCRYPT 2017, Sasaki *et al*. also proposed a new tool to search the impossible differential distinguisher [11]. Moreover, Sadeghi *et al*. used the MILP method to search the related key impossible differential distinguisher for SIMECK cipher [12].

BORON as a novel ultra-lightweight block cipher is invented by BANSOD *et al*. [13]. The encryption algorithm has many favorable advantages, *e.g.* fast encryption speed and particularly suited for resource-constrained environments. More concretely, BORON cipher uses 16 identical S-boxes in the S-layer in order to provide sufficient confusion. In the P-layer, the cipher uses 4 identical nibble-oriented replacements, 4 different cyclic shifts and XOR operations so that it can achieve better diffusion with fewer round operations. In 2019, the best differential and linear trails of BORON cipher are determined by using the SMT solver in [14]. In 2020, Li *et al*. [15] presented the integral attack on reduced-round BORON cipher. In particular, the attacks on reduced 7-, 8- and 9-round of cipher were provided by basing on a novel 6-round distinguisher. Actually, it still appears to be an interesting and crucial task to determine BORON cipher against impossible differential attack.

In this article, a new MILP search model for the impossible differential distinguisher of BORON block cipher is presented by using the S-box operation constraints, where the MILP search model has less constraints and variables. Moreover, 160 6-round impossible differential distinguishers are obtained by solving the model in the Gurobi solver. An new impossible differential attack on 10-round reduced BORON cipher is proposed, where the 6-round impossible differential distinguishers is further extended forward 2-round and backward 2-round respectively. In the key recovery phase, the data complexity can be effectively reduced by using the probability of the input/output difference of S-boxes. Compared with the previous results, this attack achieves the highest attacking round number and the lowest data complexity.

2. DESCRIPTION OF BORON

BORON uses the popular SPN cipher structure [13], where the data block size is 64-bit, and the user key size is 80-bit or 128-bit. The number of iterative rounds function operation is 25. Specially, its round function includes three basic operations: AddRoundKey, Substitution Layer and Permutation Layer. These operations are depicted in Fig. 1.

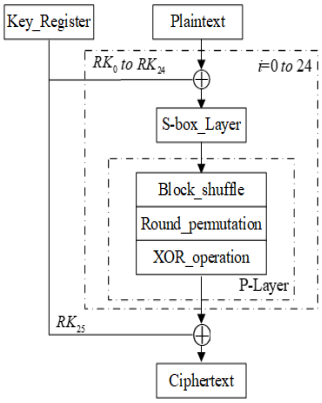

Fig. 1. Block diagram of BORON.

Let C_i be the input status of i-round of the BORON cipher, and C_{i+1} be the output (i = 0, 1, ..., 24). The update process from C_i to C_{i+1} is given below.

$$C_{i+1} = P(S(C_i)) \tag{1}$$

The Substitution Layer consists of 16 idential 4-bit S-boxes in parallel. The Permutation Layer has three operations, *i.e.*, Block_shuffle, Round_permutation and XOR_operation. The Block_Shuffle layer takes 16-bit input, and then it gives the 16-bit shuffled output. For the 64-bit block size, the Block_Shuffle operation is only repeated four times. The Round_permutation performs cyclic left shift operations on 4 16-bit blocks with 9, 7, 4 and 1-bit respectively. The XOR_operation performs XOR-ed operation after the output of Round_permutation (see Fig. 2).

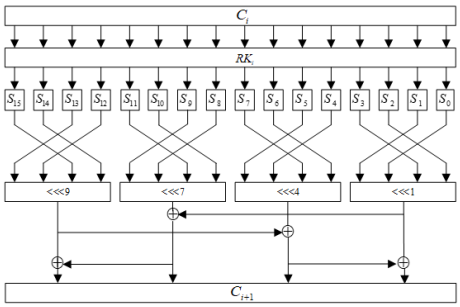

Fig. 2. Round function of BORON.

S-box_Layer The S-box_Layer of BORON uses a 4-bit input and 4-bit output S-box, S:$F_2^4 \rightarrow F_2^4$, the hexadecimal values of the S-box is given in Table 1.

Key schedule of the BORON cipher The user key size of BORON is 80- or 128-bit. 25 round sub-keys (each of size is 64-bit) are generated by the key schedule. The 80-bit user key schedule is given below.

Table 1. S-box of BORON.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	e	4	b	1	7	9	c	a	d	2	0	f	8	5	3	6

For the 80-bit user key, the master key is stored in a key register K, where it is denoted as $K = (k_{79}, ..., k_1, k_0)$. In the first place, the least significant 64 bits in the register K are extracted as the sub-keys $RK_i = (k_{63}, ..., k_1, k_0)$, $i = 0, 1, ..., 24$. Moreover, the register K is updated as follows.

$$\begin{cases} K <<< 13 \\ [k_3, k_2, k_1, k_0] \leftarrow S[k_3, k_2, k_1, k_0] \\ [k_{63}, k_{62}, k_{61}, k_{60}, k_{59}] \leftarrow S[k_{63}, k_{62}, k_{61}, k_{60}, k_{59}] \oplus RC_i, \end{cases} \tag{2}$$

where RC_i(i = 0, 1, ..., 24) is the round constant.

3. AUTOMATED SEARCH OF IMPOSSIBLE DIFFERENTIAL DISTINGUISHER of BORON VIA MILP METHOD

The core idea of the MILP method is to transform the problem of searching for an impossible differential distinguisher into a mathematical optimization problem. There are two parts of this model, *i.e*. inequality constraints and objective function. It is necessary to make the objective function empty when we start to search the impossible differential distinguishers.

3.1 Construction MILP Model

The main operations of the block cipher include linear operations (cyclic_shift, XOR_operation, *etc*.) and nonlinear operations (S-box). In this case, it is mainly constrained for input/output differences of XOR_operations, cyclic_shift, and S-boxes. Note that the shift_operation only changes the bit position, it cannot be directly described in the model. The constraints on XOR_operation and S-box operation are given below.

The XOR Operation Let $c = a \oplus b$, for bit-level XOR_operation, the two input differences are a and b, and the output difference is c, (see Fig. 3).

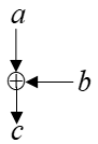

Fig. 3. The XOR operation.

Let $(a, b, c) \in \{(0,0,0),(1,1,0),(1,0,1),(0,1,1)\}$ be the possible differential patterns. Based on the logical condition, the inequality constraints on the bit-level XOR operation are as follows.

$$\begin{cases} a+b-c \geq 0 \\ a+c-b \geq 0 \\ b+c-a \geq 0 \\ a+b+c \leq 2 \end{cases} \tag{3}$$

The S-box Operation Suppose $(x_3,x_2,x_1,x_0) \xrightarrow{s-box} (y_3,y_2,y_1,y_0)$, where x_i and y_j respectively denote the i-bit input difference and j-bit output difference. There are two cases as below.

(1) The input/output difference of the S-box is zero or nonzero. Let variable A_t be an indicator. $A_t = 1$ if and only if the input difference of the S-box is fixed to non-zero, that is, the t-th S-box is active; otherwise $A_t = 0$. The inequality constraints are given below.

$$\begin{cases} x_3 - A_t \leq 0 \\ x_2 - A_t \leq 0 \\ x_1 - A_t \leq 0 \\ x_0 - A_t \leq 0 \\ x_3 + x_2 + x_1 + x_0 - A_t \leq 0 \end{cases} \tag{4}$$

For bijective S-boxes, a nonzero input difference will induce a nonzero output difference and vice versa. Then the following the inequality constraints should be added. -0.05in

$$\begin{cases} 4\sum_{j=1}^{n} y_j - \sum_{i=1}^{n} x_i \leq 0 \\ 4\sum_{i=1}^{n} x_i - \sum_{j=1}^{n} y_j \leq 0 \end{cases} \tag{5}$$

(2) For the specific value of the non-zero input/output differential of the S-box, the model contains three steps. In the first place, one needs to construct a two-dimensional array according to the specific value of the non-zero input/output difference of the S-box. Moreover, the convex hull of two-dimensional array can be obtained by SageMath software. Finally, the inequalities of S-box is simplified by the algorithm proposed in [16].

3.2 Constructing MILP Model for the Round Function of BORON

The round function of BORON cipher consists of S-box_Layer, P_layer and AddRoundKey. We only focus on considering the impossible differential distinguisher under the single key scenario. Correspondingly, we mainly describe the inequalities of the S-

box_Layer and P_layer of the BORON cipher. The S-box_Layer is a 4-bit S-box and the P_layer mainly contains Block_shuffle, Round_permutation and XOR_operation.

3.2.1 The constraints for S-boxes

The input/output differences of the S-box of BORON cipher are described in Table 2. If the input difference is 0010, the possible patterns combine with the output difference can be expressed as arrays [0010, 0011], [0010, 0101], [0010, 1011], [0010, 1101]. Similarly, all input/output differences can be expressed as a set *Points*, *Points* =[[0000, 0000], [0001, 0101], [0001, 0110], [0001, 1010], [0001, 1101], [0001, 1110], [0001, 1111], [0010, 0011], [0010, 0101], [0010, 1011], [0010, 1101],..., [1110, 0010], [1110, 0011], [1110, 0100], [1110, 0110], [1110, 0111], [1110, 1000], [1110, 1101], [1111, 0111], [1111, 1000], [1111, 1001], [1111, 1110]].

Moreover, the differential propagation of the S-box is described by the inequalities. As a result, 230 inequalities are obtained for all possible input/output difference propagation patterns. Finally, 23 inequalities are remained by the algorithm proposed in [16]. So in this case, we need only 23 inequalities to describe the differential property of S_0 in Eqs.(6). And since there are 16 S-boxes, we actually require 368 inequalities to describe the constraints in total.

$$\begin{cases} -x_3+x_1+x_0-y_2-y_1 \geq 0 \\ x_3-3x_2+x_1-x_0-3y_3-2y_2-3y_1-y_0 \geq -10 \\ -x_3-2x_2+2x_1-x_0-y_3-2y_2-2y_1-y_0 \geq -10 \\ x_3-2x_2+2x_1-x_0-y_3-2y_2-2y_1-y_0 \geq -4 \\ -2x_3-2x_2+x_0+2y_2+y_1+y_0 \geq 0 \\ \cdots \\ -2x_3-2x_2-x_1+2x_0-2y_3+22y_2-y_1+y_0 \geq -7 \\ -x_3-2x_2-2x_1-x_0-y_3+2y_2-2y_1+y_0 \geq -7 \\ -2x_3-3x_2-3x_1+x_0-3y_3+y_2-2y_1-y_0 \geq -11 \\ 7x_3-3x_2-3x_1-2x_0-3y_3-2y_2+y_1-y_0 \geq -11 \end{cases} \tag{6}$$

3.2.2 The constraints for the linear layer

The inequalities for Block_shuffle, Round_permutation and XOR_operation in the BORON cipher are described as follows.

Let m_i be the input difference of the Block_shuffle and Round_permutation, and n_i be the output difference of the Block_shuffle and Round_permutation. The Block_shuffle and Round_permutation of the 16-bit in the BORON cipher are characterized by inequalities as Eq. (7). Similarly, the Block_shuffle and Round_permutation of the remaining three 16-bit are characterized by inequalities to obtained 64 inequalities.

$$\begin{cases} m_0 - n_8 = 0 \\ m_1 - n_9 = 0 \\ m_2 - n_{10} = 0 \\ m_3 - n_{11} = 0 \\ \cdots \\ m_{12} - n_4 = 0 \\ m_{13} - n_5 = 0 \\ m_{14} - n_6 = 0 \\ m_{15} - n_7 = 0 \end{cases} \tag{7}$$

For the XOR_operation of the linear diffusion layer structure in the BORON cipher, it is described according to the XOR_operation characterizations in Section 3.1. Let l_i be the input difference of XOR_operation and q_i be the output difference. In this case, the XOR_operation of the first 16-bit in the 64-bit block are characterized by inequalities as Eq. (8). At last, the XOR_operations of the remaining three 16-bit are characterized by inequalities to obtain 256 inequalities.

$$\begin{cases} l_0 + l_{16} - q_0 \geq 0 \\ l_0 - l_{16} + q_0 \geq 0 \\ -l_0 + l_{16} + q_0 \geq 0 \\ l_0 + l_{16} + q_0 \leq 2 \\ l_1 + l_{17} - q_1 \geq 0 \\ l_1 - l_{17} + q_1 \geq 0 \\ -l_1 + l_{17} + q_1 \geq 0 \\ l_1 + l_{17} + q_1 \leq 2 \\ \cdots \\ l_{14} + l_{30} - q_{14} \geq 0 \\ l_{14} - l_{30} + q_{14} \geq 0 \\ -l_{14} + l_{30} + q_{14} \geq 0 \\ l_{14} + l_{30} + q_{14} \leq 2 \\ l_{15} + l_{31} - q_{15} \geq 0 \\ l_{15} - l_{31} + q_{15} \geq 0 \\ -l_{15} + l_{31} + q_{15} \geq 0 \\ l_{15} + l_{31} + q_{15} \leq 2 \end{cases} \tag{8}$$

3.3 The Automatic Searching Algorithm for Impossible Differential Distinguisher

The python interface provided by the Gurobi is used to search the whole process of **Algorithm 1**. The experiment is performed in the environment PC (Intel(R) Core(TM) i5-8265U, 8GB RAM, Windows10).

Table 2. The input and output differential distribution of BORON's S-box.

Input difference	Output difference
0000	0000
0001	0101, 0110, 1010, 1101, 1110, 1111
0010	0011, 0101, 1011, 1101
0011	0010, 0101, 0110, 1101, 1110, 1111
0100	0011, 0101, 0111, 1001, 1011, 1101
0101	0001, 0011, 0110, 0111, 1000, 1010, 1100, 1101
0110	0010, 0100, 1000, 1010, 1100, 1110
0111	0001, 0010, 0100, 0101, 0110, 0111, 1000, 1011
1000	0011, 0110, 1011, 1100, 1110, 1111
1001	0001, 0010, 0100, 1001, 1010, 1100
1010	0011, 0100, 0110, 1011, 1110, 1111
1011	0001, 0010, 0100, 1001, 1010, 1100
1100	0001, 0101, 0110, 0111, 1000, 1010, 1011, 1100
1101	0010, 0011, 0100, 0101, 1010, 1011, 1100, 1101
1110	0001, 0010, 0011, 0100, 0110, 0111, 1000, 1101
1111	0111, 1000, 1001, 1110

There are 64×64 input/output difference pairs in Algorithm 1. And an impossible differential distinguisher will be obtained if the model has no solution. Actually in this simulation, 160 6-round impossible differential distinguishers can be obtained by searching 4096 cases (the simulation spends about 456.6 seconds). Some part of 6-round impossible differential distinguishers are listed in Table 3, where v_i $(0 \leq i \leq 7)$ indicates that the difference of the ith bit of the word is 1, and the remaining bit differences are 0, where 0 indicates the difference of the word is zero.

4. IMPOSSIBLE DIFFERENTIAL ATTACK ON BORON CIPHER

In this section, the probability of input/output differences of S-boxes of BORON cipher are discussed. Moreover, by using the differential property of the S-box, the 10-round impossible differential attack on BORON cipher is proposed.

4.1 The Differential Property of S-boxes

For BORON cipher, if the input difference is 0010, then the output difference should be 0011, 0101, 1011, or 1101. If the 0-th bit (*i.e.*, the rightmost bit or the lowest bit) of these output differences is 1, and the rest bits of the output difference are fixed to 0 or 1, one can denote these unknown difference as "*", but the output difference is marked by "***1". For the S-box used in BORON cipher, **Property 1** below can be existed.

Property 1: Once the input (or output difference) of the S-box is fixed, then the bit positions of the corresponding output (or input) difference can also be easily confirmed. For example, one can check the results in Table 4, where "*" indicates the unknown difference.

Algorithm 1: Automatic Search for Impossible Difference Distinguisher of BORON
Input: 64-bit input difference set C, 64-bit output difference set P;
Output: r-round impossible differential distinguisher *List*;

```
Initialization;
Constructing MILP model M of impossible differential distinguisher for BORON;
// The model uses the "lp" file format;
for i in C do
   for j in P do
      M ← create_model(i, j);
      // The constraints of the input/output difference are written into the model file M;
      Using Gurobi to solve the model M;
      if M.status == 3 do // M.status==3 means the model has no solution;
          List.append(i, j);
       end if
    end for
 end for
 return List;
```

Table 3. The impossible differential distinguishers of BORON.

Input difference	Output difference
(0,0,0,0,v_6,0,0,0)	(0,0,0,v_4,0,0,0,0)
(0,0,0,0,v_6,0,0,0)	(0,0,0,0,0,0,0,v_5)
(0,0,0,0,v_6,0,0,0)	(0,0,0,0,0,0,0,v_7)
(0,0,0,0,v_6,0,0,0)	(0,v_2,0,0,0,0,0,0)
(0,0,0,0,v_6,0,0,0)	(0,0,0,0,0,0,v_0,0)

Actually, these differential features of the S-box can be very helpful for increasing the round number of the impossible differential distinguishers.

On the other hand, there always exists certain probability distribution for given output and input difference pairs. For example, if the output difference is fixed to (0010), and then the input difference can be one of the eight values, *i.e.*, $\{(0001), (0011), (0101), (1001), (1011), (1101), (0111), (1111)\}$. Specially, If the eight results are marked as ***1, then the probability (Pr) for each input difference is 1/8. The input/output differential probability of the S-box in the BORON cipher is summarized in **Property 2** below.

Property 2: The differential probability can be determined, if the input and output difference (listed in Table 5) are given. In fact, by using the input/output differential probability distribution of the S-box above, the data complexity can be effectively reduced in the key recovery phase.

4.2 Key Recovery

In this section, the impossible differential attack on the BORON cipher is discussed. In the first place, a 6-round impossible differential distinguisher is selected, *i.e.*, (0,0,0,0,v_6,0,0,0) $\not\rightarrow$ (0,0,0,0,0,0,0,v_5) , which is one of the 160 6-round impossible differential distinguishers searched in Section 3.3. Moreovere, based on the S-box input/output difference feature of Property 1, the selected distinguisher is respectively extended 2

Table 4. The input/output differential property of the S-box of BORON.

Input difference	0010	0100	0110	*1**	*1**	1111
Output difference	***1	***1	***0	0111	1000	*0**

Table 5. The property of the input/output differential probability of the S-box.

Input difference	***1	****	****	****	**10	0***	1000	000*
Output difference	0010	0***	*000	1**0	****	****	*1**	****
Probability	1/8	1/8	1/8	1/4	1/4	1/2	1/8	1/8

rounds forward and 2 rounds backward. In this case, a 10-round impossible differential attack is performed.

The portion round subkeys of rounds 1, 2 and 9, 10 are guessed so that the wrong round subkeys are filtered via the 6-round impossible differential distinguisher. Based on the S-box input/output differential probability of the property 2, the ciphertext pairs in the data set are selected, and the data complexity in the key recovery phase is estimated. In the key recovery phase, the plaintext-ciphertext pairs that meet the corresponding difference conditions are retained. The key recovery procedure is given in Fig. 4, which is described below.

Step 1: Select 2^m data set;

Step 2: Guess the 24-bit key values of $RK_0[3\sim0]$, $RK_0[15\sim12]$, $RK_0[39\sim36]$, $RK_0[47\sim44]$, $RK_0[51\sim48]$ and $RK_0[55\sim52]$. The difference of these selected data pairs should satisfy ΔS_0. The remained pairs should be about 2^{m-13} since the probability of S(****)=0***, S(****)=*000, S(****)=1**0, S(****)=000*, S(****)=*000 and S(****)=0*** be 1/2, 1/8, 1/4, 1/8, 1/8 and 1/2, respectively.

Step 3: Guess the 24-bit key values of $RK_9[3\sim0]$, $RK_9[15\sim12]$, $RK_0[35\sim32]$, $RK_0[47\sim44]$, $RK_0[51\sim48]$ and $RK_0[63\sim60]$. The difference of these selected data pairs should satisfy ΔC_9. The remained pairs should be about 2^{m-28} since the probability of S^{-1}(****)=000*, S^{-1}(****)=**10, S^{-1}(****)=000*, S^{-1}(****)=**10, S^{-1}(****)=000* and S^{-1}(****)=**10 be 1/8, 1/4, 1/8, 1/4 and 1/8, respectively.

Step 4: Guess the 12-bit key values of $RK_1[27\sim24]$, $RK_1[35\sim32]$ and $RK_0[55\sim52]$. The difference of these selected data pairs should satisfy ΔS_1. The remained pairs should be about 2^{m-39} since the probability of S(****)=0100, S(*1**)=1000 and S(****)=0100 be 1/16, 1/8 and 1/16, respectively.

Step 5: Guess the 4-bit key values of $RK_8[7\sim4]$. According to $\Delta C_8 = C_8 \oplus C_8'$, $C_8 = (S^{-1}(P^{-1}(C_9)))\oplus RK_8$, each pair in the remained data set is calculated. As the probability of S^{-1}(***1) = 0010 is 1/8, the difference of these selected data pairs should satisfy $\Delta C_8[7\sim4]$ = **10. Finally the remaining pairs should be about 2^{m-42}.

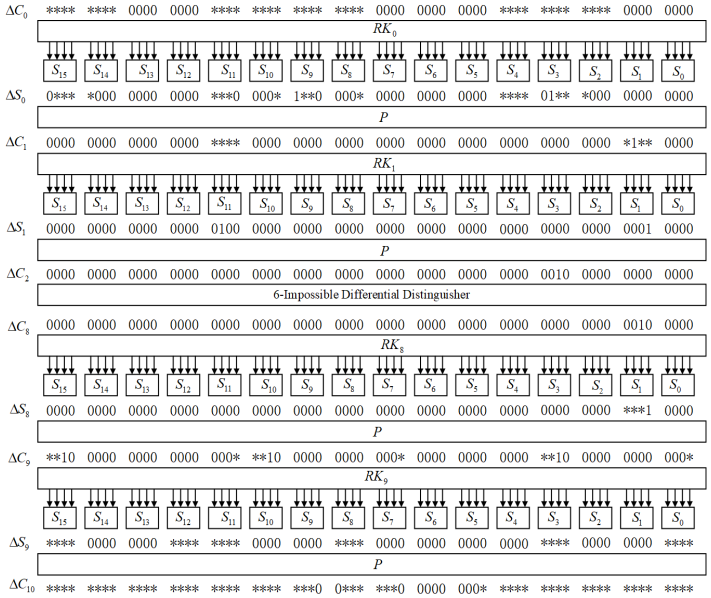

Fig. 4. Impossible difference attack of BORON.

4.3 Complexity Analysis

In the key recovery phase, 64-bit round subkeys are guessed, and then the wrong keys can be filtered. Actually, there are about $2^{64}\times(1-2^{-4})^{2^{m-42}}$ candidate keys remaining. When the number of remaining candidate keys is small than 1, the correct key can be recovered well, that is $2^{64}\times(1-2^{-4})^{2^{m-42}} \leq 1$, where $m \approx 43.52$. In this case, the error rate is $Error$=$(1-2^{-4})^{2^{m-42}} \approx e^{-2^{m-38.05}} \approx e^{-2^{5.47}} \approx 2^{-2^{5.99}}$.

Moreover, the data complexity of this attack is about 2^m=$2^{43.52}$ chosen data, and the memory complexity is about $2\times 2^m = 2^{44.52}$ encryption operations. In particular, according to the key schedule, we can easily recover the master key. Therefore, the time complexity of the attack is about $2^{62.08}$ 10-round encryption operations in total, (see Table 6).

4.4 The Comparison of Previous Results with Our Results

The results comparison are shown in Table 7.

In [14], 8-round differential trials of BORON cipher were searched based on SAT/SMT, and a 9-round key recovery attack was performed after extended one round. An automatic search method based on the bit-division property was also used to search 6-round integral distinguisher in [15]. At the same time, for 7-, 8-, and 9-round of BORON cipher, the key recovery attacks were proposed in this work. However, in our new attack method, the impossible differential distinguisher on the BORON cipher is automatically searched, where some 6-round impossible differential distinguishers are attained. Moreover, using the input/output differential property of the S-box, a 10-round key recovery attack is performed after extended respectively 2 rounds forward and 2 rounds backward.

Table 6. The time complexity of the impossible differential attack on BORON.

Step	Guess	Size	Time complexity
(1)	/	0	$2^m \times \frac{1}{10}$
(2)	$RK_0[3\sim0]$, $RK_0[15\sim12]$, $RK_0[39\sim36]$, $RK_0[47\sim44]$, $RK_0[51\sim48]$ and $RK_0[55\sim52]$	24	$2^m \times 2^{24} \times \frac{1}{10}$
(3)	$RK_9[3\sim0]$, $RK_9[15\sim12]$, $RK_0[35\sim32]$, $RK_0[47\sim44]$, $RK_0[51\sim48]$ and $RK_0[63\sim60]$	24	$2^{m-13} \times 2^{24} \times 2^{24} \times \frac{1}{10}$
(4)	$RK_1[27\sim24]$, $RK_1[35\sim32]$ $RK_0[55\sim52]$	12	$2^{m-13} \times 2^{24} \times 2^{24} \times 2^{12} \times \frac{1}{10}$
(5)	$RK_8[7\sim4]$	4	$2^{m-13} \times 2^{24} \times 2^{24} \times 2^{12} \times 2^4 \times \frac{1}{10}$

Table 7. The different cryptanalysis of BORON cipher.

Attack types	Attack rounds	Data complexity	Time complexity	Memory complexity	Resource
Differential Cryptanalysis	9	2^{63}	2^{56}	2^{24}	[14]
Integral Cryptanalysis	7	2^{54}	$2^{54.19}$	-	[15]
	8	2^{54}	$2^{58.34}$	-	
	9	$2^{57.90}$	$2^{94.06}$	-	
Impossible Differential Cryptanalysis	10	$2^{43.52}$	$2^{62.08}$	$2^{44.52}$	New

It is illustrated that this new attack requires about $2^{43.52}$ chosen data, $2^{62.08}$ 10-round encryption operations, and $2^{44.52}$ bits memory. Compared with the previous results, this attack achieves the highest attacking round number and the lowest data complexity.

5. CONCLUSION

In this article, a new impossible differential cryptanalysis of BORON cipher is presented. More concretely, 160 6-round impossible differential distinguishers are attained by using a new MILP model. Moreover, the 10-round impossible differential attack on reduced BORON cipher is performed by respectively extending a 6-round distinguisher to 2 rounds forward and 2 rounds backward. Finally, compared to previous known results, a favorable attacking round numbers and a lesser data complexity of this new attack are achieved.

ACKNOWLEDGMENT

This article is supported in part by the National Natural Science Foundation of China (62062026, 61872103), in part by the Foundation of Science and Technology on Commu-

nication Security Laboratory (6142103190103), in part by the Key Research and Development Plan of Guangxi (guike AB18281019), in part by the Innovation Research Team Project of Guangxi (2019GXNSFGA245004), in part by the scientific research project of young innovative talents of Guangxi (guike AD20238082), and in part by the Guangxi Natural Science Foundation Project (2020GXNSFBA297076).

REFERENCES

1. W. Wu and L. Zhang, "Lblock: a lightweight block cipher," in *Proceedings of International Conference on Applied Cryptography and Network Security*, 2011, pp. 327-344.
2. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, pp. 450-466.
3. W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms," *Science China Information Sciences*, Vol. 58, 2015, pp. 1-15.
4. A. S. E. Biham, "Differential cryptanalysis of des-like cryptosystems," *Journal of Cryptology*, Vol. 4, 1991, pp. 3-72.
5. M. Mitsuru, "Linear cryptanalysis method for des cipher," in *Proceedings of Eurocrypt: Workshop on the Theory and Application of of Cryptographic Techniques*, 1994, pp. 386-397.
6. L. Knudsen, "Deal-a 128-bit block cipher," *Complexity*, Vol. 258, 1998, p. 216.
7. E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, 1999, pp. 12-23.
8. N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Proceedings of International Conference on Information Security and Cryptology*, 2011, pp. 57-76.
9. S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, "Automatic security evaluation and (related-key) differential characteristic search: application to simon, present, lblock, des (l) and other bit-oriented block ciphers," in *Proceedings of International Conference on Theory and Application of Cryptology and Information Security*, 2014, pp. 158-178.
10. T. Cui, S. Chen, K. Fu, M. Wang, and K. Jia, "New automatic tool for finding impossible differentials and zero-correlation linear approximations," *Science China-Information Sciences*, Vol. 64, 2021, pp. 1-3.
11. Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects," in *Proceedings of Annual International Conference on Theory and Applications of Cryptographic Techniques*, 2017, pp. 185-215.
12. S. Sadeghi and N. Bagheri, "Security analysis of simeck block cipher against related-key impossible differential," *Information Processing Letters*, Vol. 147, 2019, pp. 14-21.

13. G. Bansod, N. Pisharoty, and A. Patil, "Boron: an ultra-lightweight and low power encryption design for pervasive computing," *Frontiers of Information Technology & Electronic Engineering*, Vol. 18, 2017, pp. 317-331.
14. H. Liang and M. Wang, "Cryptanalysis of the lightweight block cipher boron," *Security and Communication Networks*, Vol. 2019, 2019, pp. 1-13.
15. Y. Li, M. Liang, H. Lin, and S. Wang, "Integral attack on reduced-round boron based on bit-based division property," in *Journal of Physics: Conference Series*, 2020, pp. 016-022.
16. Y. Sasaki and Y. Todo, "New algorithm for modeling s-box in milp based differential and division trail search," in *Proceedings of International Conference for Information Technology and Communications*, 2017, pp. 150-165.

Xiao-Nian Wu received the MS degree in Computer Science and Technology from National University of Defense Technology, Changsha, China in 2004. He is currently an Associate Professor with School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His main research interests include information security, distribution computing.

Ying-Xin Li received the MS degree in Information Security from Guilin University of Electronic Technology, Guilin, China in 2020. His main research interests include the analysis of cryptographic algorithm.

Ling-Chen Li received the Ph.D. degree in Computer Science and Technology from the University of Chinese Academy of Sciences, Beijing, China in 2019. Currently she is a teacher with School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. Her main research interests include the design and cryptanalysis of block cipher.

Yong-Zhuang Wei received the MS and the Ph.D. degrees in Cryptology from Xidian University, Xi'an, China, in 2004 and 2009, respectively. Since July 2011, he has been doing research with the State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing, China. Since September 2014, he joined the Guangxi Key Laboratory of Cryptography and Information Security at Guilin University of Electronic Technology, where he is currently employed as a Professor. He is now a member of Chinese Association for Cryptologic Research (CACR). His current research interests include Boolean functions, stream ciphers, block ciphers, and hash functions.

Run-Lian Zhang received the Ph.D. degree in Computer Science and Technology from Xi'an Jiaotong University, Xi'an, China in 2010. Currently she is an Associate Professor with School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. Her main research interests include information security, distribution computing.