Improved Multiple Impossible Differential Cryptanalysis of Midori128

Mohamed Tolba, Ahmed Abdelkhalek, and Amr M. Youssef

Concordia Institute for Information Systems Engineering,
Concordia University, Montréal, Quebéc, Canada

Abstract. Midori128 is a lightweight block cipher proposed at ASIACRYPT 2015 to achieve low energy consumption per bit. Currently, the best published impossible differential attack on Midori128 covers 10 rounds without the pre-whitening key. By exploiting the special structure of the S-boxes and the binary linear transformation layer in Midori128, we present impossible differential distinguishers that cover 7 full rounds including the mix column operations. Then, we exploit four of these distinguishers to launch multiple impossible differential attack against 11 rounds of the cipher with the pre-whitening and post-whitening keys.

Keywords: Cryptanalysis, Impossible differential cryptanalysis, Block ciphers, Midori128.

1 Introduction

To address the design of lightweight block ciphers having low energy consumption, Banik *et al.* [1] identified some design choices that are energy efficient and proposed a new family of lightweight block ciphers, namely, Midori. This block cipher follows the SPN design approach and has two variants, Midori128 and Midori64, using a 128-bit secret key and a 128/64-bit block, respectively. Since Midori64 has been completely broken in [2,3], in this paper we focus on Midori128. The state of Midori128 has 16 bytes presented as a 4×4 matrix. For the linear layer, Midori128 uses an almost MDS binary matrix to optimize the area and signal delay. In order to compensate its low branch number and increase the number of active S-boxes, Midori128 utilizes an optimal cell-permutation layer. In the non-linear layer, Midori128 utilizes a small-delay lightweight 4-bit S-boxes to construct its 8-bit S-box.

While the designers of Midori do not claim resistance under the related, known or chosen-key attack models [1], a related key attack against the cipher was presented in [4]. In the single-key attack model, the cipher has been analyzed in [5,6,7,8]. In particular, Chen *et al.* [5] presented 10-round impossible differential attack without the pre-whitening key utilizing a 6-round distinguisher. Later on, Sasaki and Todo [7] proposed a new tool to find impossible differential distinguishers for symmetric-key primitives, and they presented a 7-round distinguisher for Midori128 but the last round in their distingusier contains the sub cell operation only.

In this paper, we improve the previous results of the impossible differential cryptanalysis against Midori128. More specifically, we present several impossible differential distinguishers against Midori128 that cover 7 full rounds (including the linear transformation layer of the the last round). These distinguishers exploit the structure of the S-boxes that are used in Midori128 along with the binary nature of the mix column operation. In particular, we exploit that each S-box of the 4 different 8-bit S-boxes that are used in Midori128 is composed of two 4-bit S-boxes (see Fig. 1). Then, we choose differences that activate only one of these 4-bit S-boxes to find such distinguishers. Then, using four of these impossible differential distinguishers, we present 11-round multiple impossible differential attack against Miroi128 including the pre-whitening and post-whitening keys.

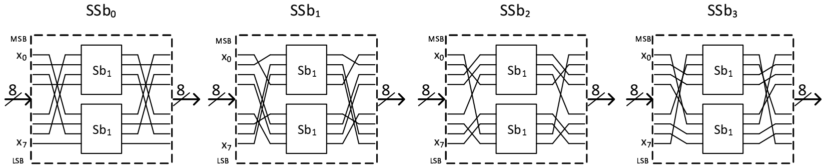

Fig. 1. SSb_0, SSb_1, SSb_2, and SSb_3 [1]

2 Specifications of Midori128

Midori128 follows the Substitution Permutation Networks (SPNs) structure iterating over 20 rounds. Each round, except the last one, consists of S-layer (*SubCell*), P-layer (*ShuffleCell* and *MixColumn*) and a key-addition layer (*KeyAdd*). The last round contains only the S-layer. Moreover, before the first and after the last rounds, pre-whitening and post-whitening are performed using *WK*. In what follows, we show how these operations update the 128-bit state S.

- *SubCell*: A nonlinear layer applies one of four 8-bit S-boxes, namely SSb_0, SSb_1, SSb_2, and SSb_3, on each byte of the state S in parallel, where $s_i \leftarrow SSb_{(i \; mod \; 4)} \; [s_i]$, $0 \leq i \leq 15$. As shown in Fig. 1, each 8-bit S-box SSb_i is composed of 8-bit input permutation, p_i, two 4-bit S-boxes (Sb_1) and 8-bit output permutation, p_i^{-1}.
- *ShuffleCell*: The bytes of the state S is permuted as follows: $(s_0, s_1, \cdots, s_{15}) \leftarrow (s_0, s_{10}, s_5, s_{15}, s_{14}, s_4, \; s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8)$.
- *MixColumn*: Each column in S is multiplied by a binary matrix M, where

$$M = M^{-1} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

- *KeyAdd*: 128-bit round key RK_i is XORed with the state S.

Key Schedule. The pre-whitening and post-whitening keys WK in Midori128 are the master key K, the round keys $RK_i = K \oplus \beta_i$, $0 \leq i \leq 18$, where β_i is a constant.

The following notations are used throughout the rest of the paper:

- x_i, y_i, z_i, w_i: The 128-bit input to the *SubCell*, *ShuffleCell*, *MixColumn* and *KeyAdd* operations, respectively, at round i.
- $x_i[j]$: The j^{th} byte of x_i, $0 \leq j < 16$.
- $x_i[j_t](resp.x_i[j_b])$: The top (resp. bottom) 4-bit of $p_1(x_i[j])$.
- $x_i[j, \cdots, l]$: The bytes $j, \cdots, l$ of x_i.
- $\Delta x_i, \Delta x_i[j]$: The difference at state x_i and byte $x_i[j]$, respectively.

3 7-round Impossible Differential Distinguishers of Midori128

Impossible Differential attacks, proposed independently by Biham *et al.* [9] and Knudsen [10] aims to find an impossible differential characteristic, i.e., one that occurs with a probability exactly 0 to act as a distinguisher. Then, the obtained distinguisher can be utilized in a key recovery attack by prepending and/or appending additional analysis rounds. Finally, the keys that are used in the analysis rounds to reach the impossible differential distinguisher are considered wrong keys, and hence they are excluded. In order to reduce the data complexity of the attack, multiple impossible differentials can be utilized [11].

All of our 7-round distinguishers begin with one active byte $w_1[i]$, $0 \leq i \leq 15$, such that only the top (resp. bottom) 4 bits of $p_{i \, mod \, 4}(w_1[i])$ are active and ends with two active bytes $x_9[i,j](i,j \in \{4l, 4l{+}1, 4l{+}2, 4l{+}3\}, 0 \leq l < 4, i \neq j)$ where only the top (resp. bottom) 4 bits of $p_{i \, mod \, 4}(x_9[i])$ and $p_{i \, mod \, 4}(x_9[j])$ are active. It can be verified that there are 24 such impossible differential distinguishers out of the possible $2 \times 16 \times 4 \times \binom{4}{2} = 768$, where we have 16 possible positions for $w_1[i]$, and for $x_9[i,j]$ we have 4 columns where in each column we have $\binom{4}{2}$ combinations for the positions of i, j, and in each one of these patterns we can activate the top or bottom 4-bit S-box. One of these distinguishers is illustrated in Fig. 2 and is based on the following propositions.

Proposition 1. *Let $\Delta = xx0000xx$, where 0 and x denote the inactive and active/inactive bits, respectively, and at least one of the x bits should be active. The probability of $\Delta \xrightarrow{SSb_1} \Delta = 1$.*

Proof. This property follows from the structural properties of the S-box, as shown in Fig. 3. As depicted in this figure, after applying the input bit permutation p_1, the input of the top 4-bit S-box Sb_1 has the difference $xxxx$ (and

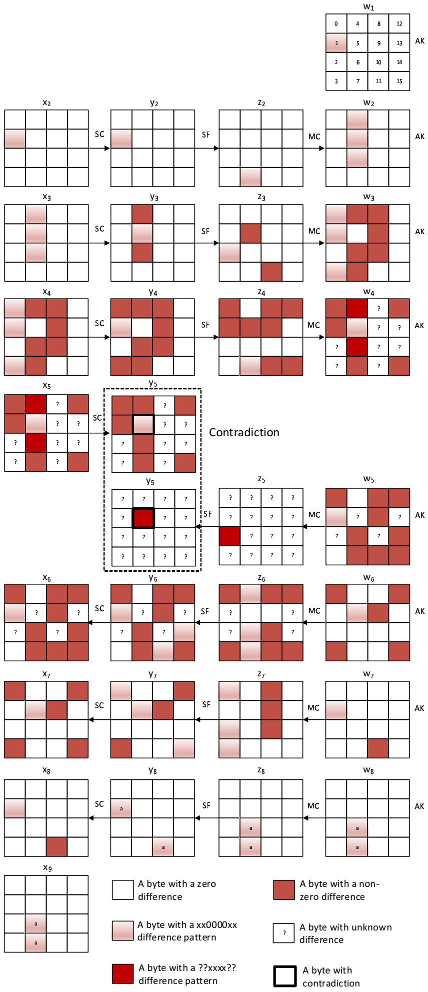
w1
0 4 8 12
1 5 9 13
2 6 10 14
3 7 11 15
x2 y2 z2 w2
x3 y3 z3 w3
x4 y4 z4 w4
x5 y5
Contradiction
y5 z5 w5
x6 y6 z6 w6
x7 y7 z7 w7
x8 y8 z8 w8
x9
SC
SF
MC
AK
?
a
A byte with a zero difference
A byte with a non-zero difference
A byte with a xx0000xx difference pattern
A byte with unknown difference
A byte with a ??xxxx?? difference pattern
A byte with contradiction

Fig. 2. 7-round impossible differential distinguisher of Midori128

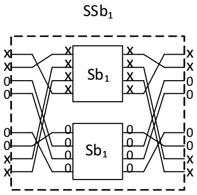

Fig. 3. The propagation of a difference pattern $\Delta = xx0000xx$ through SSb_1

hence it is active) while the input of the bottom 4-bit S-box Sb_1 has the difference 0000 (and hence it is inactive). Since the S-box Sb_1 is bijective, a non-zero input difference to the S-box, implies a non-zero output difference. Therefore, the outputs of the top and bottom 4-bit S-boxes Sb_1 have the difference $xxxx$ and 0000, respectively, with probability one. Then, applying the bit permutation p_1^{-1} makes the output of the 8-bit S-box SSb_1 has the difference pattern Δ. This property is also applicable to SSb_1^{-1}.

Proposition 2. *Let α and β have the same difference pattern Δ defined in proposition 1, and α is not necessarily equal to β. The input (hexadecimal) difference $(0\alpha00, 0000, 0000, 0000)$ cannot propagate to the output (hexadecimal) difference $(0000, 00\beta\beta\ , 0000, 0000)$ after complete 7 rounds of Midori128.*

Proof. The difference patten of α and β implies that only the top 4-bit S-box Sb_1 of the 8-bit S-box SSb_1 will be active. From the forward direction, the difference pattern α will be preserved until the internal state x_3. Then applying SSb_0 and SSb_2 on $\Delta x_3[4]$ and $\Delta x_3[6]$, respectively, will change it in a way that guarantees that the two 4-bit S-boxes are active and applying SSb_1 on $\Delta x_3[5]$ will preserve the difference pattern α (see proposition 1). Then, the differences in the internal state y_3 can be propagated in the same manner until state x_5 that will have only one byte preserving the α difference pattern at $\Delta y_5[5]$. From the other direction, since $\Delta x_9[6] = \Delta x_9[7]$ and each one of these differences preserve the β difference pattern, $\Delta z_8[6]$ will be equal to $\Delta z_8[7]$ and each one of these resulting differences will preserve the β difference pattern. From proposition 1, applying SSb_1^{-1} on $\Delta y_8[1]$ preserves the β difference pattern, while applying SSb_3^{-1} on $\Delta y_8[11]$ ensures that the top and bottom 4 bits after applying the bit input permutation p_1 of SSb_1 are active. Then, the differences can be propagated in the same manner until state w_5. From the shuffle and mix column operations, we know that $\Delta y_5[5] = \Delta z_5[2] = \Delta w_5[0] \oplus \Delta w_5[1] \oplus \Delta w_5[3]$ which means that the bottom 4-bit of $p_1(\Delta y_5[5])$ are active. However, from the forward direction, we know that the bottom 4-bit of $p_1(\Delta y_5[5])$ are inactive as this byte satisfies the α difference pattern. Therefore, there is a contradiction between the byte $\Delta y_5[5]$ in the forward and backward directions, and hence the whole truncated differential characteristic holds with probability exactly 0. The previous proposition also holds for α and β have the difference pattern $00xxxx00$.

In our attack we exploit the following 4 impossible differential distinguishers:

$$(0\alpha00, 0000, 0000, 0000) \nrightarrow (0000, 00\beta\beta, 0000, 0000)(\alpha = \beta = xx0000xx) \quad (1)$$
$$(0\alpha00, 0000, 0000, 0000) \nrightarrow (0000, 00\beta\beta, 0000, 0000)(\alpha = \beta = 00xxxx00) \quad (2)$$
$$(0\alpha00, 0000, 0000, 0000) \nrightarrow (0000, \beta00\beta, 0000, 0000)(\alpha = \beta = xx0000xx) \quad (3)$$
$$(0\alpha00, 0000, 0000, 0000) \nrightarrow (0000, \beta00\beta, 0000, 0000)(\alpha = \beta = 00xxxx00) \quad (4)$$

As can be seen, all the four distinguishers above begin with one active byte which has only 4 active bits at position 1, and end with two active bytes where each byte has only 4 active bits at the same column. It should be noted that the obtained 24 distinguishers can be categorized into 6 groups, where each group contains 4 patterns similar to the presented above, and anyone of theses groups can be used in our attack instead of the above four distinguishers.

4 11-round Multiple Impossible Differential of Midori128

In this section, we present an 11-round multiple impossible differential attack against Midori128 involving both the pre-whitening and post-whitening keys, see Fig. 4. Throughout our analysis, we utilize the following S-box property, which holds for all bijevctive S-boxes: Given two non-zero differences, Δx and Δy, in $\mathbb{F}_{256}$, the equation: $S(x) + S(x + \Delta x) = \Delta y$ has one solution on average. In what follows, we describe the details of our attack which is decomposed of 2 phases: a data collection phase, where we generate enough message pairs to exclude the wrong keys involved in the analysis rounds and a key recovery phase, where we use the collected message pairs to identify the key candidates.

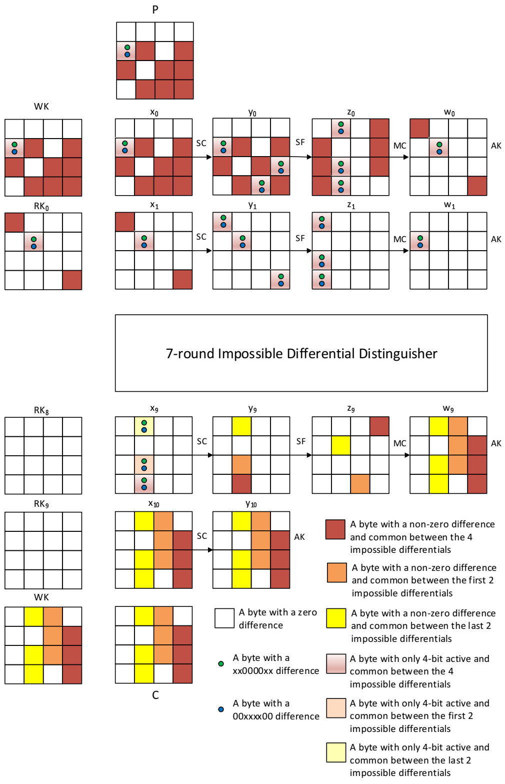
P
WK
x_0
SC
y_0
SF
z_0
MC
w_0
AK
RK_0
x_1
SC
y_1
SF
z_1
MC
w_1
AK
7-round Impossible Differential Distinguisher
RK_8
x_9
SC
y_9
SF
z_9
MC
w_9
AK
RK_9
x_{10}
SC
y_{10}
AK
WK
C
A byte with a non-zero difference and common between the 4 impossible differentials
A byte with a non-zero difference and common between the first 2 impossible differentials
A byte with a zero difference
A byte with a non-zero difference and common between the last 2 impossible differentials
A byte with a xx0000xx difference
A byte with only 4-bit active and common between the 4 impossible differentials
A byte with a 00xxxx00 difference
A byte with only 4-bit active and common between the first 2 impossible differentials
A byte with only 4-bit active and common between the last 2 impossible differentials

Fig. 4. 11-round multiple impossible differential cryptanalysis of Midori128

Data Collection. In this phase, we use the structure technique in order to reduce the data complexity of the attack. Our structure takes all the possible values in bytes $1, 2, 5, 7, 10, 11, 13, 14, 15$ while the other bytes take fixed value. Therefore, one structure generates $2^{9\times 8} \times (2^{9\times 8} - 1)/2 \approx 2^{143}$ possible message pairs. Then, we create 4 lists $L_i, 1 \leq i \leq 4$, where each list L_i contains the pairs which satisfy the plaintext and ciphertext differences of the i^{th} impossible differential. For example, L_1 is indexed by the bottom 4 bits of $p_1(P[1])$ and the following bytes of the ciphertext $C[0, 1, 2, 3, 4, 5, 6, 7, 11, 12]$. Therefore, L_1 contains $2^{143} \times 2^{-4} \times 2^{-80} = 2^{59}$ message pairs. Similarly, each one of the other lists contains 2^{59} message pairs. We take 2^m structures in order to launch the attack, and hence we have 2^{m+59} message pairs in each list L_i. Therefore, we query the encryption oracle 2^{m+72} times.

Key Recovery. We identify the key candidates by performing the following steps in parallel for each list L_i:
Step 1. Guess $\Delta z_9[12]$ and propagate it linearly forward to compute $\Delta x_{10}[13, 14, 15]$. From the knowledge of the ciphertext, compute $\Delta y_{10}[13, 14, 15]$. Using the S-box property, we can get one solution, on average, for $y_{10}[13, 14, 15]$. Therefore, we have 2^8 values for $K[13, 14, 15]$.
Step 2. For the lists L_1 and L_3 (resp. L_2 and L_4), we guess $\Delta x_1[5_t]$ (resp. $\Delta x_1[5_b]$) and propagate it linearly backward to compute $\Delta y_0[1, 11, 14]$. From the knowledge of the plaintext, we compute $\Delta x_0[1, 11, 14]$. Using the S-box property, we get one solution, on average, for $x_0[1_t, 11, 14]$ (resp. $x_0[1_b, 11, 14]$). Therefore, we have 2^4 values for $K[1_t, 11, 14]$ (resp. $K[1_b, 11, 14]$), and in total we have $2^8 \times 2^4 \times 2^{-8} = 2^4$ values for $K[1_t, 11, 13, 14, 15]$ (resp. $K[1_b, 11, 13, 14, 15]$) because we have 8-bit filter on $K[14]$.
Step 3. Guess $\Delta x_1[15]$ and propagate it linearly backward to compute $\Delta y_0[2, 7, 13]$. From the knowledge of the plaintext, compute $\Delta x_0[2, 7, 13]$. Using the S-box property, we get one solution, on average, for $x_0[2, 7, 13]$. Therefore, we have 2^8 values for $K[2, 7, 13]$, and in total we have, $2^4 \times 2^8 \times 2^{-8} = 2^4$ values for $K[1_t, 2, 7, 11, 13, 14, 15]$ (resp. $K[1_b, 2, 7, 11, 13, 14\ , 15]$) corresponding to lists L_1 and L_3 (resp. L_2 and L_4) because we have an 8-bit filter on $K[13]$.
Step 4. Repeat *Step 3*, but guess $\Delta x_1[0]$ instead of $\Delta x_1[15]$. Consequently, we have 2^8 values for $K[5, 10, 15]$, and in total we have $2^4 \times 2^8 \times 2^{-8} = 2^4$ values for $K[1_t, 2, 5, 7, 10, 11, 13, 14, 15]$ (resp. $K[1_b, 2, 5, 7, 10, 11\ , 13, 14, 15]$) corresponding to lists L_1 and L_3 (resp. L_2 and L_4) because we have 8-bit filter on $K[15]$.
Step 5. For the lists L_1 and L_3 (resp. L_2 and L_4), guess $\Delta w_1[1_t]$ (resp. $\Delta w_1[1_b]$) and propagate it linearly backward to deduce $K[0, 5_t, 15]$ (resp. $K[0, 5_b, 15]$) using the S-box property. Therefore, we have 2^4 values for $K[0, 5_t, 15]$ (resp. $K[0, 5_b, 15]$), and in total we have, $2^4 \times 2^4 \times 2^{-12} = 2^{-4}$ values for $K[0, 1_t, 2, 5, 7, 10, 11, 13, 14, 15]$ (resp. $K[0, 1_b, 2, 5, 7, 10, 11, 13, 14, 15]$) because we have 12-bit filter on $K[5_t, 15]$ (resp. $K[5_b, 15]$).
Step 6. For lists L_1 and L_2, guess $\Delta z_9[11]$ and propagate it linearly forward to deduce $K[8, 9, 10]$ using the S-box property. Therefore, we have 2^8 values for $K[8, 9, 10]$, and in total we have, $2^{-4} \times 2^8 \times 2^{-8} = 2^{-4}$ values for $K[0, 1_t, 2, 5, 7, 8,$

$9, 10, 11, 13, 14, 15]$ and $K[0, 1_b, 2, 5, 7, 8, 9, 10, 11, 13, 14, 15]$ corresponding to L_1 and L_2, respectively, because we have 8-bit filter on $K[10]$. For lists L_3 and L_4, guess $\Delta z_9[5]$ and propagate it linearly forward to deduce $K[4, 6, 7]$ using the S-box property. Therefore, we have 2^8 values for $K[4, 6, 7]$, and in total we have, $2^{-4} \times 2^8 \times 2^{-8} = 2^{-4}$ values for $K[0, 1_t, 2, 4, 5, 6, 7, 10, 11, 13, 14, 15]$ and $K[0, 1_b, 2, 4, 5\ , 6, 7, 10, 11, 13, 14, 15]$ corresponding to L_3 and L_4, respectively, because we have 8-bit filter on $K[7]$.

Step 7 For list L_1 (resp. L_2), compute $\Delta x_9[6, 7]$ and keep only the keys that make $\Delta x_9[6] = \Delta x_9[7]$ where each one has the $xx0000xx$ (resp. $00xxxx00$) difference pattern. Therefore, we have $2^{-4} \times 2^{-12} = 2^{-16}$ values for $K[0, 1_t, 2, 5, 7, 8, 9, 10, 11, 13, 14, 15]$ (resp. $K[0, 1_b, 2, 5, 7, 8, 9, 10, 11, 13, 14\ , 15]$), i.e., we remove 1 key value for the 92-bit key $K[0, 1_t, 2, 5, 7, 8, 9, 10, 11, 13\ , 14, 15]$ (resp. $K[0, 1_b\ , 2, 5, 7, 8, 9, 10, 11, 13, 14, 15]$) after processing 2^{16} message pairs. For list L_3 (resp. L_4), compute $\Delta x_9[4, 7]$ and keep only the keys that make $\Delta x_9[4] = \Delta x_9[7]$ and each one has the $xx0000xx$ (resp. $00xxxx00$) difference pattern. Therefore, we have $2^{-4} \times 2^{-12} = 2^{-16}$ values for $K[0, 1_t, 2, 4, 5, 6, 7, 10, 11, 13, 14, 15]$ (resp. $K[0, 1_b, 2, 4, 5, 6, 7, 10, 11, 13, 14, 15]$).

Attack Complexity. As explained in the previous steps, for each list L_i, we have 92 key bits involved in the analysis rounds, and for each message pair we remove, on average, 2^{-16} values. Therefore, the probability that a wrong key is not discarded for each message pair is $1 - 2^{-16}/2^{92} = 1 - 2^{-108}$. Hence, after processing all the 2^{m+59} message pairs, we have $2^{92} \times (1 - 2^{-108})^{2^{m+59}} \approx 2^{92} \times (e^{-1})^{2^{m+59-108}} \approx 2^{92} \times 2^{-1.44 \times 2^{m-49}}$ remaining candidates for 92 bits of the key. For each list L_i, in order to balance between the time and data complexities, we evaluated the time complexity of the previous steps as a function of m (see Table 1). As a result, we choose $m = 49$. Hence, for each list L_i, we have $2^{90.56}$ remaining key candidates for 92 bits of the key. We have 88 bits in common between L_1 and L_2, and 88 bits in common between L_3 and L_4. Therefore, we have only $2^{90.56} \times 2^{90.56} \times 2^{-88} = 2^{93.12}$ remaining key candidates for 96-bit of $K[0, 1, 2, 5, 7, 8, 9, 10, 11, 13, 14, 15]$, and $2^{93.12}$ remaining key candidates for 96 bits of $K[0, 1, 2, 4, 5, 6, 7, 10, 11, 13, 14, 15]$. For the resulting 2 lists, we have 80 bits in common. Finally, we have $2^{93.12} \times 2^{93.12} \times 2^{-80} = 2^{106.24}$ remaining key candidates for 112-bit of $K[0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15]$ that can be exhaustively searched along 2^{16} values for the 16 bits of the key that are not involved in the attack. The data complexity can be determined from the data collection phase, where we take 2^m structures. Therefore, the data complexity of the attack is $2^{m+72} = 2^{49+72} = 2^{121}$ chosen plaintexts. The time complexity of Step 1 - Step 7 is $4 \times 2^{116.69} = 2^{118.69}$. Hence, the time complexity of the attack is dominated by the data collection phase and the exhaustive search which can be estimated as $2^{121} + 2^{16} \times 2^{106.24} = 2^{122.75}$. The memory complexity of the attack is dominated by storing 2^{m+59} pairs for each list L_i to remove the wrong keys. Hence, the memory complexity is given by $4 \times 4 \times 2^{49+59} = 2^{112}$ 128-bit blocks.

Table 1. Time complexity of the different steps, for each list L_i, of the attack on 11-round Midori128, where NK denotes the number of keys to be excluded.

Step	Time Complexity (in 11-round encryptions)	NK	$m = 49$
1	$2 \times 2^{m+59} \times 2^{8} \times \frac{6}{2 \times 16 \times 11} \approx 2^{m+62.13}$	2^{8}	$2^{111.13}$
2	$2 \times 2^{m+59} \times 2^{12} \times \frac{5}{2 \times 16 \times 11} \approx 2^{m+65.86}$	2^{4}	$2^{114.86}$
3	$2 \times 2^{m+59} \times 2^{12} \times \frac{6}{2 \times 16 \times 11} \approx 2^{m+66.13}$	2^{4}	$2^{115.13}$
4	$2 \times 2^{m+59} \times 2^{12} \times \frac{6}{2 \times 16 \times 11} \approx 2^{m+66.13}$	2^{4}	$2^{115.13}$
5	$2 \times 2^{m+59} \times 2^{8} \times \frac{5}{2 \times 16 \times 11} \approx 2^{m+61.86}$	2^{-4}	$2^{110.86}$
6	$2 \times 2^{m+59} \times 2^{4} \times \frac{6}{2 \times 16 \times 11} \approx 2^{m+58.13}$	2^{-4}	$2^{107.13}$
7	$2 \times 2^{m+59} \times 2^{-4} \times \frac{4}{2 \times 16 \times 11} \approx 2^{m+49.54}$	2^{-16}	$2^{98.54}$

It should be noted that the use of four impossible differential distinguishers allows us to use the minimum possible data complexity which results in only one remaining message pair that satisfy the plaintext, ciphertext differences and the 2^{-108} probability of discarding wrong keys. This implies that we have, for each one of the corresponding lists, $L_i, 1 \leq i \leq 4$, a $2^{-1.44}$ filtration for the 92 key bits involved in the attack. Using the intersection between these four lists increases the sieving of remaining key candidates. On the other hand, increasing the number of impossible differential distinguishers to $v > 4$, with the same data complexity, will reduce the time complexity of the exhaustive search of the remaining candidates as it excludes more candidates, and increases the time complexity of Step 1 - Step 7 of the attack to $v \times 2^{116.69}$. Therefore, using $v > 4$ impossible differential distinguishers can only achieve some slight improvement in the overall time complexity of the attack.

5 Conclusion

In this paper, we have presented 7-round impossible distinguishers on Midori128, which, unlike the previously best known one, cover the linear transformation of the last round. Then, we exploited 4 of these distinguishers to present an 11-round attack involving the pre-whitening and post-whitening keys. This attack improves the previous best known impossible differential attack on Midori128 which covers 10 rounds without the pre-whitening key. The time, data and memory complexities of the attack are $2^{122.75}$ encryptions, 2^{121} chosen plaintexts and 2^{112} 128-bit blocks.

References

1. S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, "Midori: A block cipher for low energy," in Advances in Cryptology ASIACRYPT 2015, ed. T. Iwata and J. Cheon, pp.411–436, Springer, 2015.
2. J. Guo, J. Jean, I. Nikolić, K. Qiao, Y. Sasaki, and S.M. Sim, "Invariant subspace attack against full Midori64." Cryptology ePrint Archive, Report 2015/1189, 2015. http://eprint.iacr.org/2015/1189.
3. Y. Todo, G. Leander, and Y. Sasaki, "Nonlinear invariant attack –practical attack on full SCREAM, iSCREAM, and Midori64." Cryptology ePrint Archive, Report 2016/732, 2016. http://eprint.iacr.org/2016/732.
4. D. Gérault and P. Lafourcade, "Related-key cryptanalysis of Midori," in Progress in Cryptology – INDOCRYPT 2016, ed. O. Dunkelman and S.K. Sanadhya, pp.287–304, Springer, 2016.
5. Z. Chen, H. Chen, and X. Wang, "Cryptanalysis of Midori128 using impossible differential techniques," in Information Security Practice and Experience, ed. F. Bao, L. Chen, R.H. Deng, and G. Wang, pp.1–12, Springer, 2016.
6. M. Tolba, A. Abdelkhalek, and A.M. Youssef, "Truncated and multiple differential cryptanalysis of reduced round Midori128," in Information Security, ed. M. Bishop and A.C.A. Nascimento, pp.3–17, Springer, 2016.
7. Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects." Cryptology ePrint Archive, Report 2016/1181, 2016. http://eprint.iacr.org/2016/1181.
8. W. Cheng, Y. Zhou, and L. Sauvage, "Differential fault analysis on Midori," in Information and Communications Security, ed. K.Y. Lam, C.H. Chi, and S. Qing, pp.307–317, Springer, 2016.
9. E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," in Advances in Cryptology - EUROCRYPT 99, ed. J. Stern, pp.12–23, Springer, 1999.
10. L. Knudsen, "DEAL: A 128-bit block cipher," Complexity, vol.258, no.2, 1998. NIST AES Proposal.
11. J. Lu, O. Dunkelman, N. Keller, and J. Kim, "New impossible differential attacks on AES," in Progress in Cryptology - INDOCRYPT 2008, ed. D.R. Chowdhury, V. Rijmen, and A. Das, pp.279–293, Springer, 2008.