Cryptanalysis of Reduced-Round Midori64 Block Cipher

Xiaoyang Dong and Yanzhao Shen

Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, P. R. China
dongxiaoyang@mail.sdu.edu.cn

Abstract. Midori is a hardware-oriented lightweight block cipher designed by Banik *et al.* in ASIACRYPT 2015. It has two versions according to the state sizes, i.e. Midori64 and Midori128. In this paper, we explore the security of Midori64 against truncated differential and related-key differential attacks. By studying the compact representation of Midori64, we get the branching distribution properties of almost MDS matrix used by Midori64. By applying an automatic truncated differential search algorithm developed by Moriai *et al.* in SAC 1999, we get 3137 4-round truncated differentials of Midori64. In addition, we find some 2-round iterative differential patterns for Midori64. By searching the differential characteristics matching the differential pattern, we find some iterative 2-round differentials with probability of 2^{-24}, based on these differentials, a 11-round related-key differential characteristic is constructed. Then we mount a 14-round(out of 16 full rounds) related-key differential attack on Midori64. As far as we know, this is the first related-key differential attack on Midori64.

Keywords: Block Cipher, Truncated Differential, Related-Key Differential, Midori64.

1 Introduction

Midori block cipher is designed by Banik *et al.* in ASIACRYPT 2015. It has two versions, Midori64 and Midori128, which aim to reduce the energy consumption when implemented in hardware. Midori has attracted many cryptographers since born. Lin and Wu [1] gave a Meet-in-the-Middle attack on 12-round reduced Midori64. Guo *et al.* [2] introduced an invariant subspace attack against full Midori64 in weak key setting. Chen and Wang [3] gave a impossible attack on 10-round reduced Midori64. In this paper, we focus on the truncated differential and related-key differential attack on Midori64 block cipher.

Differential cryptanalysis is one of the principal attack methods on modern symmetric-key ciphers, which was firstly introduced by Biham and Shamir [4,5] to analyze DES block cipher [6] in 1990. Based on the differential attack, many methods have been developed to evaluate the security of block ciphers, such truncated differential attack [7], related-key differential attack [8,9,10], high-order differential attack [7], impossible differential attack [11,12], multiple differential attack [13] and so forth.

The truncated differential cryptanalysis was proposed by Knudsen in 1994 [7]. Different from the differential characteristic whose difference values are fully specified, the truncated differential is regarded as a set of differential characteristics that only parts of the difference values are specified. In [14], Li *et al.* proposed a meet-in-the-middle method to find truncated differential. Related-key attack [8], allow the cryptanalyst to choose appropriate relation between keys and then to predict the encryptions under these keys. The combination of the related-key attack and differential attack is called related-key differential attack [8,9,10]. The key point of these above methods is to find a (truncated/related-key) differential characteristic with high probability which covers as many rounds as possible. In [15], Matsui *et al.* introduced an approach for finding good differential and linear characteristic in DES. It finds the best n-round characteristic by first finding the best 1,2,...,$n-1$ round characteristics.

Matsui's method requires building all one round characteristics whose number depends on the size of the search space. However, it is infeasible for many modern word-oriented block ciphers, due to the state size is too large. To solve this problem, many prior works, such as [16,17,18], adopt the so-called compact representation, where each word[1] is replaced by a single bit: a word with a nonzero difference, also called an active word, is replaced by 1, else, by 0. Then a n-word cipher is represented as n-bit vector.

Many word-oriented ciphers are designed as substitution-permutation(SP) networks(SPN) or Feistel stucture with SP function, including the standard cipher AES, newly proposed cipher Midori [19] and E2 block cipher [20]. The SPN ciphers or SP functions have a layer of S-boxes(S-layer) and a linear diffusion layer(P-layer), usually the linear-layer is composed of a word rearrangement and a multiplication by a matrix A, for examples, the ShiftRow and MixColumn constitute the P-layer of AES, and the ShuffleCell and MixColumn form the P-layer of Midori. When the S-boxes are bijective(a property common to most ciphers developed in the past decades), then an active word remains active(and vice-versa) before and after the S-box. Hence, the S-layer does not change the compact representation. On the other hand, the P-layer can change the number of the active words as well as their positions, and therefore the branches are introduced here. For example, for XOR operation, if the two differences (0,0,0,0,0,0,1,0) and (0,0,0,0,0,0,1,0) were XORed, the result could be (0,0,0,0,0,0,1,0) or (0,0,0,0,0,0,0,0), i.e. two branches are introduced by XOR. Moreover, in a byre-oriented cipher, the output (0,0,0,0,0,0,0,0) is obtained only

[1] Usually, a word is a byte or nibble.

when the two active input byte differences are equal, this happens with probability of about 2^{-8}(the 8 corresponding bits of the two bytes are all equal). On the other hand, the output (0,0,0,0,0,0,0,0) is obtained with probability of $1-2^{-8}$. In [17], by the evaluating branches and their corresponding probabilities introduced by P-layer, Moriai *et al.* developed a truncated differential search algorithm, and found some truncated differentials for round-reduced E2 block cipher.

Our contributions.

In this paper, we apply Moriai *et al.*'s truncated differential search algorithm to the newly proposed Midori64 block cipher. We explore the branching property of Mixcolumn of Midori64 block cipher. The branches with different probabilities are computed and listed in a distribution table. We find many 4-round truncated differentials for Midori64, whose probabilities are higher than the average probability. By exhaustive search, we claim that there are no truncated differentials with more rounds for Midori64 block ciphers. As the Mixcolumn matrix are also used in Midori128, we also conclude there are no truncated differentials with more than 4 rounds for Midori128.

Moreover, for Midori64 block cipher, we find some 2-round iterative truncated differential pattern. Then, by searching the differential characteristics matching these truncated differential patterns, we find some iterative 2-round differentials with probability of 2^{-24}. Based on these differentials, 11-round related-key differential characteristics are constructed. Then we mount a 14-round related-key differential attack on Midori64. As far as we know, this is the best attack on Midori64 in respect of the attacked rounds (excluding the weak-key attack). All the results are summarized in Tab. 1.

Table 1. Summary of the Attacks on Midori64

Rounds	Attack Type	Data	Time	Memory	Source
Single-key Attack (full key space)					
10	Impossible Differential Attack	$2^{62.4}$CP	$2^{80.81}$Enc	$2^{65.13}$	[3]
10	Meet-in-the-Middle Attack	$2^{61.5}$CP	$2^{99.5}$Enc	$2^{92.7}$	[1]
11	Meet-in-the-Middle Attack	2^{53}CP	2^{122}Enc	$2^{89.2}$	[1]
12	Meet-in-the-Middle Attack	$2^{55.5}$CP	$2^{125.5}$Enc	2^{106}	[1]
Related-key Attack (full key space)					
14	Related-key Differential Attack	2^{59}CP	2^{116}Enc	2^{112}	Sec. 6
Weak-key Attack (2^{32} weak keys)					
16(full)	Invariant Subspace Attack	2CP	2^{16}Enc	–	[2]

CP: chosen-plaintext; Enc: encryption

Outline of the paper.

The rest of the paper is organized as follows: Sec. 2 gives a brief description of Midori block cipher. In Sec. 3, some related works are listed. In Sec. 4, we present the branching property of almost MDS matrix in compact representation. In Sec. 5, Moriai *et al.*'s algorithm is applied to Midori64 block cipher. Then 14-round related-key differential attack on Midori64 is introduced in Sec. 6. At last, Sec. 7 concludes this paper.

2 Brief Description of Midori

Midori [19] is a lightweight block cipher designed by Banik *et al.* at AISACRYPT 2015. It is of the Substitution-Permutation Network(SPN). There are two versions of Midori with state sizes of 64-bit and 128-bit, denoted as Midori64 and Midori128, respectively. They are designed to reduce energy consumption when implemented in hardware. Midori64 has 64-bit state size and its key size is 128-bit. It uses the following 4×4 array as a data expression:

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}$$

where the size of each cell is 4-bit. Please note, $S[i] = s_i, i = 0, 1, ..., 15$, which is used in Sec. 6.

The round function F of Midori64 is composed of the following 4 operations:

- **SubCell(SC):** Apply the non-linear $4\times$ 4 S-box in parallel on each nibble of the state.

Table 2. SubCell

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
$Sb(x)$	0xc	0xa	0xd	0x3	0xe	0xb	0xf	0x7	0x8	0x9	0x1	0x5	0x0	0x2	0x4	0x6

- **ShuffleCell(ShC):** Each nibble of the state is performed as follows: $(s_0, s_1, \ldots, s_{15}) \leftarrow (s_0, s_{10}, s_5, s_4, s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8)$.
- **MixColumn(MC):** Midori64 utilizes an almost MDS matrix $\mathbf{M}$ defined as follows:

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

  $\mathbf{M}$ is applied to every 4-nibble column of the state S, i.e. ${}^t(s_i, s_{i+1}, s_{i+2}, s_{i+3}) \leftarrow \mathbf{M} \cdot^t (s_i, s_{i+1}, s_{i+2}, s_{i+3})$ *and* $i = 0, 4, 8, 12$.

- **KeyAdd(AK):** The i^{th} 64-bit round key rk_i is XORed to the state S.

Before the first round, an additional KeyAdd operation is applied, and in the last round, the ShuffleCell and MixColumn operations are omitted. The total round number of Midori64 is 16. The key-schedule of Midori64 is quite simple. A 128-bit key K is denoted as two 64-bit keys k_0 and k_1, $K = k_0||k_1$. The whitening key and the last sub-key are $rk_{-1} = rk_{R-1} = k_0 \oplus k_1$, and the sub-key for round i is $rk_i = k_{(imod2)} \oplus \alpha_i$, where $0 \leq i \leq R-2$ and α_i is constant.

3 Related works

The truncated differential cryptanalysis was proposed by Knudsen in 1994 [7]. Different from the differential characteristic whose difference values are fully specified, the truncated differential is regarded as a set of differential characteristics that only part of the difference values are specified.

Definition 1. *[21] For the block cipher E with a parameter key K, the truncated differential ($\mathit{\Gamma}_{in} \xrightarrow{E} \mathit{\Gamma}_{out}$) is a set of differential characteristics, where $\mathit{\Gamma}_{in}$ is a set of input differences, and $\mathit{\Gamma}_{out}$ is a set of output differences. The expected probability of such truncated differential ($\mathit{\Gamma}_{in} \xrightarrow{E} \mathit{\Gamma}_{out}$) is defined by*

$$\mathcal{P}r(\mathit{\Gamma}_{in} \xrightarrow{E} \mathit{\Gamma}_{out}) = \frac{1}{|\mathit{\Gamma}_{in}|} \sum_{a \in \mathit{\Gamma}_{in}} \mathcal{P}r((E_K(X) \oplus E_K(X \oplus a)) \in \mathit{\Gamma}_{out}) \qquad (1)$$

$$= \frac{1}{|\mathit{\Gamma}_{in}|} \sum_{a \in \mathit{\Gamma}_{in}} \mathcal{P}r(a \to \mathit{\Gamma}_{out}). \qquad (2)$$

The average probability of the truncated differential characteristic for a random permutation is $\mathcal{P}r(\mathit{\Gamma}_{in} \xrightarrow{E} \mathit{\Gamma}_{out}) = \frac{|\mathit{\Gamma}_{out}|}{2^n}$, where n is the block size. So what we have to find is a truncated differential with probability bigger than $\frac{|\mathit{\Gamma}_{out}|}{2^n}$.

At FSE 1999, Matsui and Tokita [16] used the compact representation to find a 7-round truncated differential of E2 block cipher without IT-Function and FT-Function. In SAC 1999, Moriai *et al.* [17] developed a truncated differential search algorithm. The algorithm is shown in Alg. 1. Let $\Delta X \to \Delta Y$ be one round truncated differential pattern, where ΔX and ΔY are input and output difference in compact representation. Let $W(\Delta X \to \Delta Y)$ be the weight function of the differential pattern, it corresponds to the probability cost required to produce a pair that follows the differential pattern, e.g. if $Pr(\Delta X \to \Delta Y) = 2^{-x}$, then $W(\Delta X \to \Delta Y) = x$. $W(\Delta Y)$ and $W(\Delta Z)$ are number of bits of all active words. $\bar{W}$ is the bit number of the state in a block cipher, e.g. $\bar{W}$ of Midori64 is 64. TD_n is n-round truncated differential (not necessarily optimal).

Algorithm 1 Search for n-round Truncated Differential

```
for all $\{\varDelta X \rightarrow \varDelta Y | W(\varDelta X \rightarrow \varDelta Y) + W(\varDelta Y) < \bar{W}\}$ do
    Call NextRound($\varDelta Y, W(\varDelta X \rightarrow \varDelta Y), 2$)
end for

NextRound($\varDelta Y, w, r$)
for all $\{\varDelta Z | \varDelta Y \rightarrow \varDelta Z$ and $w + W(\varDelta Y \rightarrow \varDelta Z) + W(\varDelta Z) < \bar{W}\}$ do
    if $r = n$ and $w + W(\varDelta Y \rightarrow \varDelta Z) + W(\varDelta Z) < \bar{W}$ then
        Update $TD_n$
    else
        Call NextRound($\varDelta Z, w + W(\varDelta Y \rightarrow \varDelta Z), r + 1$)
    end if
end for
```

4 Branching Property of Almost MDS Matrix *M* in Compact Representation

In this section, we investigate the branching property of the involutive almost MDS matrix **M** showed in Equ. (3), which is used by Midori64 block cipher in the **MixColumn(MC)** operation. The input of **MC** is $x = (x_0, x_1, x_2, x_3)^T$ and output is $y = (y_0, y_1, y_2, y_3)^T$, x_i, $y_i \in F_2^c$.

$$\mathbf{M} \cdot x = y \Rightarrow \begin{pmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \tag{3}$$

We calculate all the possible outputs y for different input x, and classify them as the following four cases:

1. **case 1**: if there is only one active nibble in the input, e.g. $x_0 \neq 0$ and $x_1 = x_2 = x_3 = 0$, then the output vector $(y_0, y_1, y_2, y_3) = (0, x_0, x_0, x_0)$.
2. **case 2**: if there are two active nibbles, e.g. $x_0 \neq 0$, $x_2 \neq 0$, and $x_1 = x_3 = 0$, then,
   (a) when $x_0 = x_2$, $(y_0, y_1, y_2, y_3) = (x_0, 0, x_0, 0)$,
   (b) when $x_0 \neq x_2$, $(y_0, y_1, y_2, y_3) = (x_2, x_0 \oplus x_2, x_0, x_0 \oplus x_2)$.
3. **case 3**: if there are three active nibbles, e.g. $x_0 \neq 0$, $x_1 \neq 0$, $x_2 \neq 0$ and $x_3 = 0$, then,
   (a) when $x_0 = x_1 = x_2$, $(y_0, y_1, y_2, y_3) = (0, 0, 0, x_0)$,
   (b) when $x_0 = x_1 \neq x_2$, $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_2, x_0 \oplus x_2, 0, x_2)$,
   (c) when $x_0 = x_2 \neq x_1$, $(y_0, y_1, y_2, y_3) = (x_1 \oplus x_2, 0, x_1 \oplus x_2, x_1)$,
   (d) when $x_1 = x_2 \neq x_0$, $(y_0, y_1, y_2, y_3) = (0, x_0 \oplus x_1, x_0 \oplus x_1, x_0)$,
   (e) when $x_0 \neq x_1$, $x_0 \neq x_2$ and $x_1 \neq x_2$,

i. $x_0 \oplus x_1 \oplus x_2 = 0$, then $(y_0, y_1, y_2, y_3) = (x_1 \oplus x_2, x_0 \oplus x_2, x_0 \oplus x_1, 0)$,

ii. $x_0 \oplus x_1 \oplus x_2 \neq 0$, then $(y_0, y_1, y_2, y_3) = (x_1 \oplus x_2, x_0 \oplus x_2, x_0 \oplus x_1, x_0 \oplus x_1 \oplus x_2)$.

4. **case 4**: if there are four active nibbles,

   (a) when $x_0 \oplus x_1 \oplus x_2 = 0$, and

      i. $x_3 \neq x_0$,$x_3 \neq x_1$ and $x_3 \neq x_2$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_3, x_1 \oplus x_3, x_2 \oplus x_3, 0)$,

      ii. $x_3 = x_0$,$x_3 \neq x_1$ and $x_3 \neq x_2$, then $(y_0, y_1, y_2, y_3) = (0, x_1 \oplus x_3, x_2 \oplus x_3, 0)$,

      iii. $x_3 \neq x_0$,$x_3 = x_1$ and $x_3 \neq x_2$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_3, 0, x_2 \oplus x_3, 0)$,

      iv. $x_3 \neq x_0$,$x_3 \neq x_1$ and $x_3 = x_2$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_3, x_1 \oplus x_3, 0, 0)$,

   (b) when $x_0 \oplus x_1 \oplus x_3 = 0$,

      i. $x_2 \neq x_0$,$x_2 \neq x_1$ and $x_2 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_2, x_1 \oplus x_2, 0, x_3 \oplus x_2)$,

      ii. $x_2 = x_0$,$x_2 \neq x_1$ and $x_2 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (0, x_1 \oplus x_2, 0, x_3 \oplus x_2)$,

      iii. $x_2 \neq x_0$,$x_2 = x_1$ and $x_2 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_2, 0, 0, x_3 \oplus x_2)$,

      iv. $x_2 \neq x_0$,$x_2 \neq x_1$ and $x_2 = x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_2, x_1 \oplus x_2, 0, 0)$,

   (c) when $x_0 \oplus x_2 \oplus x_3 = 0$,

      i. $x_1 \neq x_0$,$x_1 \neq x_2$ and $x_1 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_1, 0, x_2 \oplus x_1, x_3 \oplus x_1)$,

      ii. $x_1 = x_0$,$x_1 \neq x_2$ and $x_1 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (0, 0, x_2 \oplus x_1, x_3 \oplus x_1)$,

      iii. $x_1 \neq x_0$,$x_1 = x_2$ and $x_1 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_1, 0, 0, x_3 \oplus x_1)$,

      iv. $x_1 \neq x_0$,$x_1 \neq x_2$ and $x_1 = x_3$, then $(y_0, y_1, y_2, y_3) = (x_0 \oplus x_1, 0, x_2 \oplus x_1, 0)$,

   (d) when $x_1 \oplus x_2 \oplus x_3 = 0$,

      i. $x_0 \neq x_1$,$x_0 \neq x_2$ and $x_0 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (0, x_0 \oplus x_1, x_0 \oplus x_2, x_0 \oplus x_3)$,

      ii. $x_0 = x_1$,$x_0 \neq x_2$ and $x_0 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (0, 0, x_0 \oplus x_2, x_0 \oplus x_3)$,

      iii. $x_0 \neq x_1$,$x_0 = x_2$ and $x_0 \neq x_3$, then $(y_0, y_1, y_2, y_3) = (0, x_0 \oplus x_1, 0, x_0 \oplus x_3)$,

      iv. $x_0 \neq x_1$,$x_0 \neq x_2$ and $x_0 = x_3$, then $(y_0, y_1, y_2, y_3) = (0, x_0 \oplus x_1, x_0 \oplus x_2, 0)$,

   (e) else then, $(y_0, y_1, y_2, y_3) = (x_1 \oplus x_2 \oplus x_3, x_0 \oplus x_2 \oplus x_3, x_0 \oplus x_1 \oplus x_3, x_0 \oplus x_1 \oplus x_2)$.

In the compact representation, the input and output of **MC** are described as 4-bit vectors, i.e. the active nibble is replaced by 1, else by 0. For example, the input vector and output vector in case 1 are (1,0,0,0) and (0,1,1,1) in compact representation. In case 2 (a), when the input vector is (1,0,1,0), the output vector is (1,0,1,0) with one equality condition of $x_0 = x_2$ whose probability is 2^{-4}. In the above 4 cases, one equality condition means the probability is 2^{-4}, and for two equality conditions, the probability is 2^{-8} and so on. Besides, if there are no equality conditions, we denote the probability as $1-\varepsilon$. To be simple, we represent the 4-bit vector compact representation with a hexadecimal number, called *simplified compact representation*. For example, (1,0,0,0) is denoted as 0x8 and (1,0,1,0) as 0xa in simplified compact representation. Then we get the following branching property table in Tab. 3 with probability distribution for MC, where the row is input vector, and column is output vector, and the units store the probabilities. 4 represents 2^{-4}, 8 represents 2^{-8} and 1 represents $1-\varepsilon$. For example, if the input of **MC** is 0x7 in simplified compact representation, then the possible outputs of **MC** are 0x7,0x8,0xb,0xd,0xe,0xf with probabilities 2^{-4},2^{-8},2^{-4},2^{-4},2^{-4} and $1-\varepsilon$, where $\varepsilon = 2^{-4} + 2^{-8} + 2^{-4} + 2^{-4} + 2^{-4}$.

Table 3. Branching Property Table with Probability Distribution for **MC**

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0x3	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	1
0x4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0x5	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1
0x6	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	1
0x7	0	0	0	0	0	0	0	4	8	0	0	4	0	4	4	1
0x8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0x9	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1
0xa	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1
0xb	0	0	0	0	8	0	0	4	0	0	0	4	0	4	4	1
0xc	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1
0xd	0	0	8	0	0	0	0	4	0	0	0	4	0	4	4	1
0xe	0	8	0	0	0	0	0	4	0	0	0	4	0	4	4	1
0xf	0	0	0	8	0	8	8	4	0	8	8	4	8	4	4	1

5 Truncated Differential Search for Midori64

In the compact representation, the 16 nibbles of the state of Midori64 are replaced by a 16-bit vector, if the nibble is active, then the corresponding bit is 1, else 0. Moreover, the compact representation of the state is simplified to be four hexadecimal numbers. For example, the compact representation (0111000000100000) is simplified as 0x7020. Obviously, in the compact representation the **SC** and **ShC** operations do not introduce branches, the branching occurs only because of the **MC** operation. When applying Alg. 1 to find truncated differential, we use Tab. 3 to calculate the branches and the weight function of the differential pattern $W(\Delta X \rightarrow \Delta Y)$, i.e. for a given input ΔX of **MC**, a possible output ΔY, $W(\Delta X \rightarrow \Delta Y) = -log_2(Pr(\Delta X \xrightarrow{MC} \Delta Y))$(when $Pr(\Delta X \xrightarrow{MC} \Delta Y) = 1 - \varepsilon$, $W(\Delta X \rightarrow \Delta Y) = 0$). For example in Fig. 1, in the compact representation, if the input difference pattern of **MC** is ΔX =0x7020, then there are 6 possible output ΔYs after **MC**, the weight function of the differential pattern $W(\Delta X \rightarrow \Delta Y)$ is 4,8,4,4,4,0 for different branches, respectively.

$W(\Delta Y)$ and $W(\Delta Z)$ are calculated by multiplying the number of active nibbles with 4. $\bar{W}$ is 64. We find 3137 4-round truncated differentials. Fig. 2 shows one of them whose probability is 2^{-44}, where the average probability is $\frac{|\Gamma_{out}|}{2^{n-1}} = 2^{-48}$. By exhaustive search, we claim that there are no such truncated differentials with more than 4 rounds for Midori64. The truncated differential is so short that we remove the corresponding attacks.

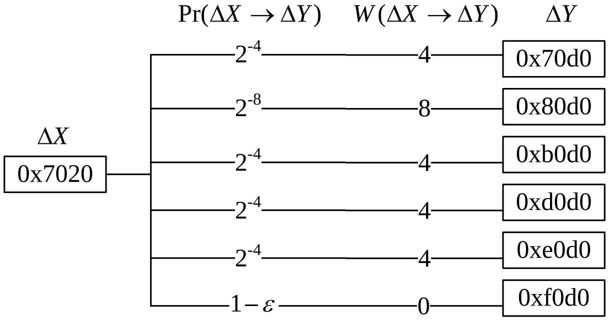

Fig. 1. Branch Weight on Midori64

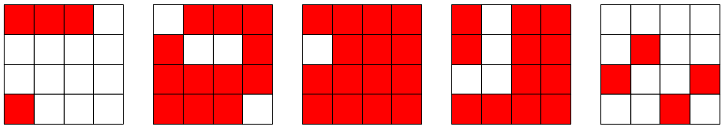

Fig. 2. 4-Round Truncated Differential of Midori

6 14-Round Related-Key Differential Attack on Midori64

During the truncated differential search phase, we find many iterative 2-round truncated differentials, including four 2-round differentials of 4 → 12 → 4 such as Equ. 4 in compact representation, 24 2-round differentials of 6 → 10 → 6 such as Equ. 5, 12 2-round differentials of 5 → 13 → 5 such as Equ. 6.

$$\begin{pmatrix} 0&0&0&1\\ 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 1&1&1&0\\ 0&1&1&1\\ 1&1&0&1\\ 1&0&1&1 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 0&0&0&1\\ 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0 \end{pmatrix}. \tag{4}$$

$$\begin{pmatrix} 1&0&1&1\\ 1&0&0&0\\ 0&1&0&0\\ 1&0&0&0 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 0&1&1&1\\ 1&1&0&1\\ 1&1&0&1\\ 1&0&0&0 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 1&0&1&1\\ 1&0&0&0\\ 0&1&0&0\\ 1&0&0&0 \end{pmatrix}. \tag{5}$$

$$\begin{pmatrix} 0&0&0&1\\ 1&0&0&0\\ 0&0&1&0\\ 1&1&0&0 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 1&1&1&0\\ 0&1&1&1\\ 1&1&1&1\\ 1&0&1&1 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 0&0&0&1\\ 1&0&0&0\\ 0&0&1&0\\ 1&1&0&0 \end{pmatrix}. \tag{6}$$

iterative of S-box in Tab. 4 shows the probabilities of (0x2→0x1, 0x1→0x2, 0x2→0x4, 0x4→0x2, 0x2→0x9, 0x9→0x2,0x2→0xc, 0xc→0x2) are all 2^{-2}. We find 256 2-round differential characteristics with probability 2^{-32}, which are all match the differential pattern of Equ. (4) and all the 256 characteristics have the same input and output difference of (0x0,0x2,0x0,0x0,0x0,0x0,0x0,0x2, 0x0,0x0,0x2,0x0,0x2,0x0,0x0,0x0), such as Equ. (7). So we add all the 256 characteristics to obtain a differential with probability of $2^{-32} \times 256 = 2^{-24}$, shown in Equ. (8).

$$\begin{pmatrix} 0x0&0x0&0x0&0x2\\ 0x2&0x0&0x0&0x0\\ 0x0&0x0&0x2&0x0\\ 0x0&0x2&0x0&0x0 \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 0x9&0x1&0x4&0x0\\ 0x0&0x1&0x4&0xc\\ 0x9&0x1&0x0&0xc\\ 0x9&0x0&0x4&0xc \end{pmatrix} \xrightarrow{F} \begin{pmatrix} 0x0&0x0&0x0&0x2\\ 0x2&0x0&0x0&0x0\\ 0x0&0x0&0x2&0x0\\ 0x0&0x2&0x0&0x0 \end{pmatrix}. \tag{7}$$

Table 4. iterative of Sbox

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1	0	2	4	0	2	2	2	0	2	0	0	0	0	0	2	0
0x2	0	4	0	0	4	0	0	0	0	4	0	0	4	0	0	0
0x3	0	0	0	0	2	0	4	2	2	2	0	0	0	2	0	2
0x4	0	2	4	2	2	2	0	0	2	0	0	2	0	0	0	0
0x5	0	2	0	0	2	0	0	4	0	2	4	0	2	0	0	0
0x6	0	2	0	4	0	0	0	2	2	0	0	0	2	2	0	2
0x7	0	0	0	2	0	4	2	0	0	0	0	2	0	4	2	0
0x8	0	2	0	2	2	0	2	0	0	2	0	2	2	0	2	0
0x9	0	0	4	2	0	2	0	0	2	2	0	2	2	0	0	0
0xa	0	0	0	0	0	4	0	0	0	0	4	0	0	4	0	4
0xb	0	0	0	0	2	0	0	2	2	2	0	4	0	2	0	2
0xc	0	0	4	0	0	2	2	0	2	2	0	0	2	0	2	0
0xd	0	0	0	2	0	0	2	4	0	0	4	2	0	0	2	0
0xe	0	2	0	0	0	0	0	2	2	0	0	0	2	2	4	2
0xf	0	0	0	2	0	0	2	0	0	0	4	2	0	0	2	4

$$\begin{pmatrix} 0x0 & 0x0 & 0x0 & 0x2 \\ 0x2 & 0x0 & 0x0 & 0x0 \\ 0x0 & 0x0 & 0x2 & 0x0 \\ 0x0 & 0x2 & 0x0 & 0x0 \end{pmatrix} \xrightarrow[Pr=2^{-24}]{two\ rounds} \begin{pmatrix} 0x0 & 0x0 & 0x0 & 0x2 \\ 0x2 & 0x0 & 0x0 & 0x0 \\ 0x0 & 0x0 & 0x2 & 0x0 \\ 0x0 & 0x2 & 0x0 & 0x0 \end{pmatrix}. \tag{8}$$

Taking advantage of the 2-round differential, a 11-round related-key differential characteristic of Midori64 is constructed, showed in Fig. 3 whose probability is 2^{-56}. The key difference is $(\Delta k_0 || \Delta k_1) = (0200000200202000||0000000000000000)$.

Then add 1 round on the top and two rounds at the bottom to attack 14-round reduced Midori64, shown in Fig. 4. The attack procedures are as follows.

1. *Structures.* Choose 2^x pairs of structures. In the paired structures, there are 2^{48} plaintexts in one structure with $P = (x_0, \alpha_0, x_1, x_2, x_3, x_4, x_5, \alpha_1, x_6, x_7, \alpha_2, x_8, \alpha_3, x_9, x_{10}, x_{11})$, and 2^{48} plaintexts in its paired one with $P = (x_{12}, \alpha_0 \oplus 0x2, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, \alpha_1 \oplus 0x2, x_{18}, x_{19}, \alpha_2 \oplus 0x2, x_{20}, \alpha_3 \oplus 0x2, x_{21}, x_{22}, x_{23})$, where x_i takes all the possible values and α_j are fixed constant. For one pair of structures, we pick one plaintext in one structure and one in the other structure to construct a pair. Totally, 2^{96+x} pairs are obtained.
2. *Attack.* Guess $k_0 \oplus k_1[0]$, encrypt a nibble of the pairs to state Y_1 and use the difference $0x2$ to filter pairs. There are 2^{92+x} pairs left.
3. Similar to step 2, we repeat the guess and filter procedure for $k_0 \oplus k_1[2, 3, 4, 5, 6, 8, 9, 11, 13, 14, 15]$ one nibble by one nibble. There are 2^{48+x} pairs left at last.

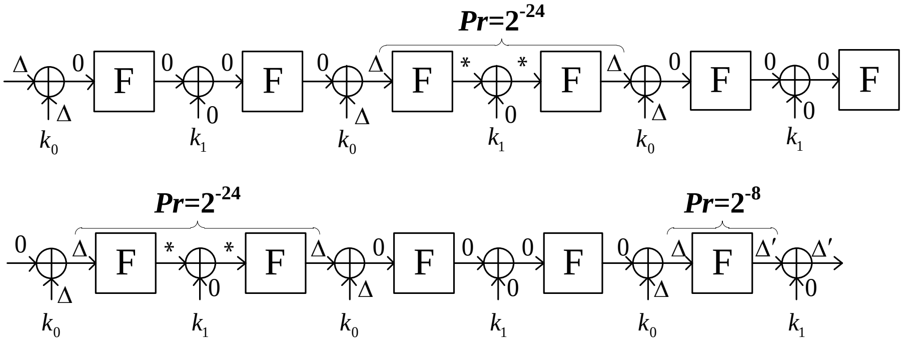

Fig. 3. 11-round Related-key Differential Characteristic of Midori64

4. Guess $k_0 \oplus k_1[1,7,10,12]$ one by one, decrypt the left pairs to state W'_{13} and use the difference $\varDelta W'_{13}[1,7,10,12] = 0$ to filter pairs. There are 2^{32+x} pairs left.
5. Guess $MC^{-1}(k_0)[0,2,3,4,5,6,8,9,11,13,14,15]$ one by one, decrypt to the state X_{13} and use the difference $0x1$ to filter pairs. There are 2^{-16+x} pairs left on average.

Choose $x = 10$, there are $2^{-16+10} = 2^{-6}$ pairs left for a random key. However, For the right key there are $2^{48+10-56} = 4$ pairs expected to left. The data complexity is $2^{48+10+1} = 2^{59}$ chosen plaintexts. The time complexity of step 1 is 2^{59} 14-round encryptions; step 2 costs about $2^{116+4} \times \frac{1}{14} \times \frac{1}{16} = 2^{120-3.8-4} = 2^{111.2}$ 14-round encryptions; The total time complexity of step 2-step 3 is $2^{111.2} = 2^{114.7}$ 14-round encryptions. Step 4 costs $2^{111.2} = 2^{113.7}$ 14-round encryptions. Step 5 costs $2^{111.2} = 2^{114.7}$ 14-round encryptions. So the total time complexity is about $2^{59} + 2^{114.7} + 2^{113.7} + 2^{114.7} = 2^{116}$ 14-round encryptions. The memory cost is 2^{112} 112-bit words to store the counters and keys.

7 Conclusion

In this paper, we explore the security of Midori64 against truncated differential and related-key differential attacks. By studying the compact representation of Midori64, we get the branching distribution properties of almost MDS matrix used by Midori64. By applying an automatic truncated differential search algorithm developed by Moriai *et al.* in SAC 1999, we get many 4-round truncated differentials of Midori64. Moreover, we find some 2-round iterative differential patterns for Midori64. By searching the differential characteristics matching the

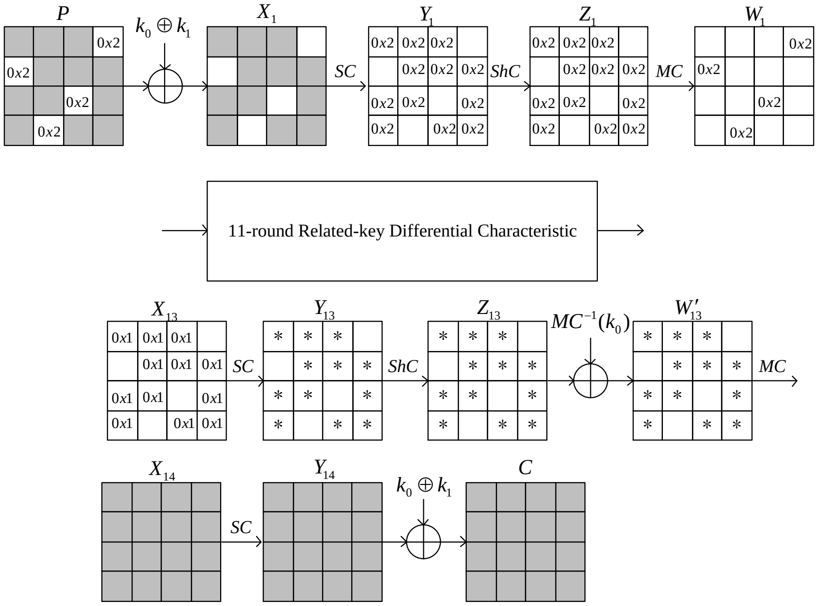

Fig. 4. 14-round Related-Key Differential Attack on Midori64

differential pattern, we find some iterative 2-round differentials with probability of 2^{-24}, based on these differentials, a 11-round related-key differential characteristic is constructed. Then we mount a 14-round(out of 16 full rounds) related-key differential attack on Midori64. As far as we know, this is the first related-key differential attack on Midori64. For Midori128, there are similar 2-round differentials, however, its round key equals to the master key, we can not connect them to get a long related-key differential characteristic.

8 Acknowledgement

We would like to thank the reviewers of SAC 2016 for their meaningful comments. One of the reviewers suggest that the paper could enjoy another round of improvements for the 14-round related-key differential attack. However, we could not find a sound method to achieve this improvement. If the reviewer or others are willing to improve this paper, we welcome the possible cooperation.

References

1. Lin, L. and Wu, W. Meet-in-the-middle attacks on reduced-round midori-64 Cryptology ePrint Archive, Report 2015/1165 (2015) `http://eprint.iacr.org/2015/`

1165.

2. Guo, J., Jean, J., Nikolić, I., Qiao, K., Sasaki, Y., and Sim, S. M. Invariant subspace attack against full midori64 Cryptology ePrint Archive, Report 2015/1189 (2015) `http://eprint.iacr.org/`.
3. Zhan, C. and Xiaoyun, W. Impossible differential cryptanalysis of midori Cryptology ePrint Archive, Report 2016/535 (2016) `http://eprint.iacr.org/2016/535`.
4. Biham, E. and Shamir, A. (1991) volume **537**, of LNCS : Springer pp. 2–21.
5. Biham, E. and Shamir, A. (1991) *J. Cryptology* **4(1)**, 3–72.
6. National Bureau of Standards (1977) In In FIPS PUB 46, Federal Information Processing Standards Publication : .
7. Knudsen, L. R. (1995) volume **1008**, of LNCS : Springer pp. 196–211.
8. Biham, E. and Shamir, A. (1994) *J. Cryptology* **7(4)**, 229–246.
9. Biryukov, A., Khovratovich, D., and Nikolic, I. (2009) volume **5677**, of LNCS : Springer pp. 231–249.
10. Biryukov, A. and Khovratovich, D. (2009) volume **5912**, of LNCS : Springer pp. 1–18.
11. Knudsen, L. (1998) *NIST AES Proposal*.
12. Biham, E., Biryukov, A., and Shamir, A. (1999) volume **1592**, of LNCS : Springer pp. 12–23.
13. Blondeau, C. and Gérard, B. (2011) volume **6733**, of LNCS : Springer pp. 35–54.
14. Li, L., Jia, K., Wang, X., and Dong, X. (2015) In Fast Software Encryption Springer : pp. 48–70.
15. Matsui, M. (1994) In Advances in CryptologyEUROCRYPT'94 Springer : pp. 366–375.
16. Matsui, M. and Tokita, T. (1999) In Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings : pp. 71–80.
17. Moriai, S., Sugita, M., Aoki, K., and Kanda, M. (1999) In Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings : pp. 106–117.
18. Biryukov, A. and Nikolić, I. (2010) Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to aes, camellia, khazad and others In Advances in Cryptology–EUROCRYPT 2010 pp. 322–344 Springer.
19. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., and Regazzoni, F. (2014) Midori: A block cipher for low energy In Advances in Cryptology–ASIACRYPT 2015 pp. 411–436 Springer.
20. Wu, W. and Zhang, L. (2011) In Applied Cryptography and Network Security Springer : pp. 327–344.
21. Blondeau, C. (2013) Improbable differential from impossible differential: on the validity of the model In Progress in Cryptology–INDOCRYPT 2013 pp. 149–160 Springer.