A Novel Automatic Technique Based on MILP to Search for Impossible Differentials

Yong Liu¹, Zejun Xiang²(\boxtimes), Siwei Chen², Shasha Zhang², and Xiangyong Zeng¹

 ¹ Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan, China liuyong_crypto@163.com, xzeng@hubu.edu.cn
 ² School of Cyber Science and Technology, Hubei University, Wuhan, China

School of Cyber Science and Technology, Hubei University, Wuhan, China xiangzejun@hubu.edu.cn, {chensiwei_hubu,amushasha}@163.com

Abstract. The Mixed Integer Linear Programming (MILP) is a common method of searching for impossible differentials (IDs). However, the optimality of the distinguisher should be confirmed by an exhaustive search of all input and output differences, which is clearly computationally infeasible due to the huge search space.

In this paper, we propose a new technique that uses two-dimensional binary variables to model the input and output differences and characterize contradictions with constraints. In our model, the existence of IDs can be directly obtained by checking whether the model has a solution. In addition, our tool can also detect any contradictions between input and output differences by changing the position of the contradictions. Our method is confirmed by applying it to several block ciphers, and our results show that we can find 6-, 13-, and 12-round IDs for Midori-64, CRAFT, and SKINNY-64 within a few seconds, respectively. Moreover, by carefully analyzing the key schedule of Midori-64, we propose an equivalent key transform technique and construct a complete MILP model for an 11-round impossible differential attack (IDA) on Midori-64 to search for the minimum number of keys to be guessed. Based on our automatic technique, we present a new 11-round IDA on Midori-64, where 23 nibbles of keys need to be guessed, which reduces the time complexity compared to previous work. The time and data complexity of our attack are $2^{116.59}$ and 2^{60} , respectively. To the best of our knowledge, this is the best IDA on Midori-64 at present.

Keywords: IDA · Midori-64 · CRAFT · SKINNY-64 · MILP

1 Introduction

Impossible differential cryptanalysis, independently proposed by Knudsen [11] and Biham et al. [4], is one of the most well-known cryptanalysis methods. Unlike differential cryptanalysis [5], which exploits differential characteristics with a high probability, the goal of impossible differential cryptanalysis is to use differentials with a probability of zero to eliminate the key candidates that lead

to such IDs. Finding an ID $(\Delta_{in}, \Delta_{out})$ that covers as many rounds as possible is the key step in an IDA. Up to now, several approaches to finding IDs have been proposed.

Initially, the miss-in-the-middle technique was the commonly used method for detecting IDs [6]. Then, the \mathcal{U} -method [10] was proposed by Kim et al. To find an $(r_1 + r_2)$ -round ID, the attacker simultaneously propagates Δ_X and Δ_Y forward r_1 rounds and backward r_2 rounds, respectively, and checks the difference of each output word separately. If any contradiction occurs, (Δ_X, Δ_Y) is a valid $(r_1 + r_2)$ -round ID. Finally, this method was further extended, such as the UID-method [13] and the extended tool by Wu and Wang [17].

In addition, there are some automatic techniques to find IDs. In 2016, Cui et al. took the differential and linear properties of non-linear components such as S-boxes into consideration and proposed a new automatic technique that can be generalized to modular additions [8]. In 2017, Sasaki et al. proposed a new technique that can detect any contradictions between input and output differences to search for IDs [14]. However, if one wants to ensure that the cipher does not exist valid IDs, all these methods need to traverse the input and output differences, which is computationally infeasible. In 2020, Sun et al. developed a Constraint Programming (CP)-aided version of the \mathcal{U} -method called \mathcal{U}^* -method. which employs the *miss-in-the-middle* technique to search for (related-key) IDs and zero-correlation linear approximations of several SPN ciphers [16]. To utilize the information of nonzero fixed differences, they imported an integer variable ζ_{X_i} for each X_i to represent the actual difference ΔX_i . In addition, the method proposed in [16] only focuses on finding the longest distinguishers and does not consider key recovery. Recently, Hu et al. proposed a new method to detect all IDs based on MILP models with the Difference Distribution Table (DDT) considered [9]. This new method partitions the whole search space into smaller ones and some of them can be quickly determined to contain no IDs. Thus, the search space is significantly reduced, sometimes to a practical size. Then the attackers could handle the remaining candidates to check if there are any IDs.

Our Contribution. In this paper, we propose a new MILP-based technique to search for IDs. Specifically, we introduce two different types of variables to describe the state differences. A **Type-1** variable is used to describe a fixed difference pattern (can be either zero difference or nonzero difference), and a **Type-2** variable is used to describe a varied difference pattern (such as a state difference pattern after one nonlinear layer). The simplified characterization of state differences allows us to directly model both linear and nonlinear layers, instead of characterizing the propagation of difference patterns through three basic operations (branch, XOR, and S-box). Moreover, instead of traversing the input and output differences and checking if the corresponding model is **infeasible**, our technique characterizes the contradictions with several constraints and derives an ID from the solution of the corresponding model. For a fixed position of contradiction, the number of feasible solutions of the MILP model reveals the number of IDs. So, we can obtain all valid IDs by traversing all possible positions of the contradictions. To test the effectiveness of our tool, we apply it to several block ciphers, such as Midori-64 [1], CRAFT [3], and SKINNY-64 [2]. Our results show that we can find 6-, 13-, and 12-round IDs for Midori-64, CRAFT, and SKINNY-64 within a few seconds, respectively.

Based on the fact that there will be no nonzero fixed differences of an SPN cipher after passing through a nonlinear layer under the single-key attack scenario, our method simplifies the characterization of state differences compared with the method of Sun et al. [16]. In addition, our simplified model makes it possible to combine both the ID-search and the key-recovery, aiming to directly search for an IDA with the minimum number of keys to be guessed. Specifically, to construct an 11-round IDA on Midori-64 with the minimum number of keys to be guessed, we propose an equivalent key transform technique that can convert the guessed equivalent key nibbles of an IDA into seed key nibbles by analyzing the properties of difference propagations and studying the key schedule of Midori-64. Based on such a technique, we add additional extended rounds to the MILP model to search for an 11-round IDA on Midori-64, while the objective function is to minimize the number of guessed keys. As a result, the minimum number of keys that need to be guessed in our 11-round IDA is 23 nibbles, and the time and data complexity of our attack are $2^{116.59}$ and 2^{60} , respectively. Our results are listed in Table 1.

Organization of the paper. The rest of this paper is organized as follows. Section 2 introduces some preliminaries. Section 3 studies how to model the difference propagation of some basic operations with linear inequalities. Section 4 shows some applications of our new technique. Section 5 introduces our IDA of 11-round Midori-64. Finally, Section 6 concludes the paper.

Cipher	Distinguisher/ Key recovery attack	#Round	#ID	The searc	time n ch for	eeded to (all) IDs	Ref.
SKINNY-64	Distinguisher	12	12	1.5h		1	[9]
SKINNY-64	Distinguisher	12	12		1s		This Paper
CRAFT	Distinguisher	13	12		7d		[9]
CRAFT	Distinguisher	13	12	1s			This Paper
Midori-64	Distinguisher	6	$21248~^\dagger$	90s		This Paper	
			#Koy bit	The at	The attack complexity		
			#Rey Dit	Time	Data	Memory	-
Midori-64	Key recovery attack	10	72/128	$2^{80.81}$	$2^{62.4}$	$2^{65.13}$	[7]
Midori-64	Key recovery attack	11	128/128	$2^{121.4}$	$2^{60.8}$	$2^{96.5}$	[12]
Midori-64	Key recovery attack	$12^{\ \ddagger}$	128/128	$2^{90.51}$	$2^{61.87}$	2^{41}	[15]
Midori-64	Key recovery attack	11	128/128	$2^{116.59}$	2^{60}	$2^{92.76}$	This Paper

Table 1. Results of this paper

[†] When searching for the number of IDs of 6-round Midori-64, we only consider the case where all active input differences are equal and all active output differences are equal.

[‡] This attack excludes the pre- and post-whitening keys.

2 Preliminaries

2.1 Notations

The notations used in this paper are as follows:

- \times : multiplication of the integer ring \mathbb{Z} .
- +: addition of the integer ring \mathbb{Z} .
- \mathbb{F}_2 : finite field with two elements 0 and 1.
- \mathbb{F}_2^k : k-dimensional vector space over \mathbb{F}_2 , also denoted as $\{0,1\}^k$.
- \oplus : bitwise XOR.
- A||B: concatenation of A and B.
- |A|: the size of set A.

2.2 Impossible Differential Cryptanalysis

Impossible differential cryptanalysis was independently proposed by Knudsen [11] and Biham et al. [4]. As shown in Figure 1, the procedure of impossible differential cryptanalysis can be generally divided into three phases. The first phase is to find an ID (E_2) that covers as many rounds as possible. Once such an ID has been found, one can extend this ID in both directions and guess the keys involved in these additional rounds (E_1 and E_3). We denote by k_{in} and k_{out} the key materials involved in the transitions $\Delta_{in} \rightarrow \Delta_X$ and $\Delta_{out} \rightarrow \Delta_Y$, respectively, and further denote by c_{in} and c_{out} the minus binary logarithm of the probability of those transitions. If the intermediate state difference matches the ID for some key guesses, those key candidates should be removed (this phase is generally called the key-sieving procedure). The last phase of impossible differential cryptanalysis is to check all the remaining keys from the key-sieving procedure by one encryption.



Fig. 1. Overview of impossible differential cryptanalysis

3 Modeling Difference Propagation of Basic Operations with Linear Inequalities

The new automatic technique we proposed can be used to search for IDs of SPnetwork block ciphers with a 4×4 binary MixColumn matrix. In this paper, our MILP models are constructed in a nibble-level, and we represent a state difference pattern with two-dimensional binary variables. When characterizing a state difference pattern, there are two different types:

- Type-1. Fixed difference patterns:

In this paper, we use two-dimensional binary variables $(x, y)_1$ to represent fixed difference patterns, where $(0, 0)_1$, $(0, 1)_1$, $(1, 0)_1$, and $(1, 1)_1$ indicate that the corresponding nibbles have a difference Δ_0 , Δ_1 , Δ_2 , and Δ_3 ($\Delta_0 =$ $0, \Delta_1 \neq \Delta_2 \neq \Delta_3 \neq 0$), respectively. Note that Δ_1, Δ_2 , and Δ_3 are any fixed nonzero and unequal differences.

- Type-2. Varied difference patterns:

The second type corresponds to the case that we only care whether the nibble difference is inactive, active, or unknown (can be either active or inactive). In this case, we use two-dimensional binary variables $(x, y)_2$ to represent such difference patterns, where $(0, 0)_2$, $(0, 1)_2$, and $(1, 0)_2$ indicate that the corresponding nibble difference is inactive, active, and unknown, respectively.

In the following, we will introduce difference propagation rules and show how to model SubCell (SB), ShuffleCell (SC), and MixColumn (MC) by linear inequalities (The linear inequalities in this paper are obtained by SageMath ³).

Modeling SB (SB⁻¹) Since SB is a permutation, which means an (in)active input difference will always result in an (in)active output difference. Let $(x_0, x_1)_i$ and $(y_0, y_1)_j$ $(i, j \in \{1, 2\})$ be the input and output differences of SB respectively. There are two cases for the difference propagation of SB that appeared in our MILP model:

- Case 1: $(x_0, x_1)_1 \xrightarrow{\text{SB}} (y_0, y_1)_2$, where the input and output differences of SB belong to Type-1 and Type-2 difference patterns respectively, then there are 4 possible difference transitions for SB shown in Table 2. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} -x_0 + y_1 \ge 0, \\ -x_1 + y_1 \ge 0, \\ x_0 + x_1 - y_1 \ge 0, \\ y_0 = 0. \end{cases}$$

- Case 2: $(x_0, x_1)_2 \xrightarrow{\text{SB}} (y_0, y_1)_2$, where both the input and output differences of SB belong to Type-2 difference patterns, then there are 3 possible difference transitions for SB shown in Table 2. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} x_0 - y_0 = 0, \\ x_1 - y_1 = 0, \\ -x_0 - x_1 + 1 \ge 0. \end{cases}$$

³ http://www.sagemath.org/

Table 2. Possible difference transitions for $SB(SB^{-1})$

Case	Input $(x_0, x_1)_1$	Output $(y_0, y_1)_2$	Case	Input $(x_0, x_1)_2$	Output $(y_0, y_1)_2$
Case 1	$(0,0)_1$ the others	$(0,0)_2$ $(0,1)_2$	Case 2	$(0,0)_2 \\ (0,1)_2 \\ (1,0)_2$	$(0,0)_2$ $(0,1)_2$ $(1,0)_2$

Modeling SC (SC^{-1}) Since SC is a nibble-wise permutation, we only need to permute the state difference patterns accordingly. Note that the input and output difference patterns are of the same type.

Modeling MC (MC⁻¹) Denote (m_0, m_1, m_2, m_3) and (n_0, n_1, n_2, n_3) as the input and output of MC, respectively, then

$$n_i = \bigoplus_{0 \le j \le 3} (t_i^j \times m_j),$$

where $t_i^j \in \{0, 1\}$ and $0 \le i \le 3$. When the output nibble n_i is equal to one input nibble (i.e., $\sum_{0 \le j \le 3} t_i^j = 1$), there are three cases for the difference transitions of n_i (one output nibble of MC) that appeared in our MILP model:

- **Case 1:** $(x_0, x_1)_1 \xrightarrow{\mathsf{MC}} (y_0, y_1)_1$, where both the input and output differences of MC belong to Type-1 difference patterns, then there are 4 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} x_0 - y_0 = 0, \\ x_1 - y_1 = 0. \end{cases}$$

- Case 2: $(x_0, x_1)_1 \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where the input and output differences of MC belong to Type-1 and Type-2 difference patterns respectively, then there are 4 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} -x_0 + y_1 \ge 0, \\ -x_1 + y_1 \ge 0, \\ x_0 + x_1 - y_1 \ge 0, \\ y_0 = 0. \end{cases}$$

- Case 3: $(x_0, x_1)_2 \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where both the input and output differences of MC belong to Type-2 difference patterns, then there are 3 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} x_0 - y_0 = 0, \\ x_1 - y_1 = 0, \\ -x_0 - x_1 + 1 \ge 0. \end{cases}$$

When the output nibble n_i is the sum of two input nibbles (i.e., $\sum_{0 \le j \le 3} t_i^j = 2$), there are two cases for the difference transitions of n_i (one output nibble of MC) that appeared in our MILP model:

- Case 4: $[(x_0, x_1)_1, (x_2, x_3)_1] \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where the input and output differences of MC belong to Type-1 and Type-2 difference patterns respectively, then there are 16 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} -x_1 + x_3 + y_1 \ge 0, \\ x_1 - x_3 + y_1 \ge 0, \\ -x_0 + x_2 + y_1 \ge 0, \\ x_0 - x_2 + y_1 \ge 0, \\ x_0 + x_1 + x_2 + x_3 - y_1 \ge 0, \\ -x_0 + x_1 - x_2 + x_3 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 - x_3 - y_1 + 2 \ge 0, \\ -x_0 - x_1 - x_2 - x_3 - y_1 + 4 \ge 0, \\ y_0 = 0. \end{cases}$$

- Case 5: $[(x_0, x_1)_2, (x_2, x_3)_2] \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where both the input and output differences of MC belong to Type-2 difference patterns, then there are 9 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} -x_0 - x_1 - x_2 - x_3 + 2y_0 + y_1 \ge 0, \\ -y_0 - y_1 + 1 \ge 0, \\ 2x_0 + x_1 + 2x_2 + x_3 - 2y_0 - y_1 \ge 0, \\ x_1 + x_3 - y_1 \ge 0, \\ -x_0 - x_1 + y_0 + y_1 \ge 0, \\ -x_2 - x_3 + y_0 + y_1 \ge 0. \end{cases}$$

When the output nibble n_i is the sum of three input nibbles (i.e., $\sum_{0 \le j \le 3} t_i^j = 3$), there are two cases for the difference transitions of n_i (one output nibble of MC) that appeared in our MILP model:

- Case 6: $[(x_0, x_1)_1, (x_2, x_3)_1, (x_4, x_5)_1] \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where the input and output differences of MC belong to Type-1 and Type-2 difference patterns

respectively, then there are 64 possible difference transitions for MC shown in Table 3. The following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} x_0 + x_1 + x_2 + x_3 + x_4 + x_5 - 4y_0 - y_1 \ge 0, \\ -x_0 - x_1 + x_2 - 3x_3 - 3x_4 + x_5 + 4y_0 + 3y_1 + 4 \ge 0, \\ -x_0 - x_1 - 3x_2 + x_3 + x_4 - 3x_5 + 4y_0 + 3y_1 + 4 \ge 0, \\ -x_0 - x_2 - x_4 + y_1 + 2 \ge 0, \\ -y_0 - y_1 + 1 \ge 0, \\ -x_0 + x_2 + x_4 + y_1 \ge 0, \\ x_0 - 3x_1 - x_2 - x_3 - 3x_4 + x_5 + 4y_0 + 3y_1 + 4 \ge 0, \\ x_0 - x_2 + x_4 + y_1 \ge 0, \\ x_1 + x_3 - x_5 + y_1 \ge 0, \\ -x_0 - x_1 - x_2 - x_3 + x_4 + x_5 - 2y_0 - y_1 + 4 \ge 0, \\ x_0 + x_1 - x_2 - x_3 - x_4 - x_5 - 2y_0 - y_1 + 4 \ge 0, \\ x_0 + x_1 - x_2 - x_3 - x_4 - x_5 - 2y_0 - y_1 + 4 \ge 0, \\ x_0 + x_1 - x_2 + x_3 - x_4 - x_5 - 2y_0 - y_1 + 4 \ge 0, \\ x_0 + x_1 - x_2 + x_3 - x_4 + x_5 - 2y_0 - y_1 + 4 \ge 0, \\ x_0 + x_1 - x_2 + x_3 - x_4 + x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 - x_2 + x_3 - x_4 + x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 + x_3 - x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 + x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 + x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 - x_3 + 3x_4 + x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 - x_3 + 3x_4 + x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 - x_1 + x_2 - x_3 + 3x_4 + x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0, \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 - 2y_0 - y_1 + 2 \ge 0. \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 + 2y_0 + y_1 + 2 \ge 0. \\ x_0 + x_1 + x_2 - x_3 + x_4 - x_5 + 2y_0 + y_1 + 2 \ge 0. \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 + 2y_0 + y_1 + 2 \ge 0. \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 + 2y_0 + y_1 + 2 \ge 0. \\ x_0 + x_1 - x_2 - x_3 + x_4 - x_5 + 2y_0 + y_1 + 2 \ge 0. \\ x_0 + x_1 - x_2 - x_3 + x_4 - x$$

- Case 7: $[(x_0, x_1)_2, (x_2, x_3)_2, (x_4, x_5)_2] \xrightarrow{\mathsf{MC}} (y_0, y_1)_2$, where both the input and output differences of MC belong to Type-2 difference patterns, then there are 27 possible difference transitions for MC shown in Table 3. The

following linear inequalities are sufficient to describe these transitions:

$$\begin{cases} -x_0 - x_1 - x_2 - x_3 - x_4 - x_5 + 3y_0 + y_1 \ge 0, \\ -y_0 - y_1 + 1 \ge 0, \\ -x_0 - x_1 + y_0 + y_1 \ge 0, \\ -x_2 - x_3 + y_0 + y_1 \ge 0, \\ -x_4 - x_5 + y_0 + y_1 \ge 0, \\ 2x_0 + x_1 + 2x_2 + x_3 + 2x_4 + x_5 - 2y_0 - y_1 \ge 0, \\ x_1 + x_3 + x_5 - y_1 \ge 0. \end{cases}$$

Case	Input	Output	Case	Input	Output
Case 1	$\begin{array}{c} (x_0,x_1)_1 \\ (0,0)_1 \\ (0,1)_1 \\ (1,0)_1 \\ (1,1)_1 \end{array}$	$\begin{array}{c}(y_0,y_1)_1\\(0,0)_1\\(0,1)_1\\(1,0)_1\\(1,1)_1\end{array}$	Case 2	$\frac{(x_0, x_1)_1}{(0, 0)_1}$ the others	$\begin{array}{c} (y_0, y_1)_2 \\ \hline (0, 0)_2 \\ (0, 1)_2 \end{array}$
Case 3	$\begin{array}{c}(x_0,x_1)_2\\(0,0)_2\\(0,1)_2\\(1,0)_2\end{array}$	$\begin{array}{c}(y_0,y_1)_2\\(0,0)_2\\(0,1)_2\\(1,0)_2\end{array}$			
Case 4	$ \begin{array}{c} [(x_0,x_1)_1,(x_2,x_3)_1] \\ \hline [(0,0)_1,(0,0)_1] \\ [(0,1)_1,(0,1)_1] \\ [(1,0)_1,(1,0)_1] \\ [(1,1)_1,(1,1)_1] \\ \hline \\ \end{array} $ the others	$\begin{array}{c}(y_0,y_1)_2\\(0,0)_2\\(0,0)_2\\(0,0)_2\\(0,0)_2\\(0,0)_2\\(0,1)_2\end{array}$	Case 5	$ \begin{array}{c} [(x_0, x_1)_2, (x_2, x_3)_2] \\ [(0, 0)_2, (0, 0)_2] \\ [(0, 0)_2, (0, 1)_2] \\ [(0, 1)_2, (0, 0)_2] \\ \\ \end{array} \\ the others \end{array} $	$\begin{array}{c}(y_0,y_1)_2\\(0,0)_2\\(0,1)_2\\(0,1)_2\\(1,0)_2\end{array}$
Case 6	$ \begin{array}{ } \hline [(x_0,x_1)_1,(x_2,x_3)_1,(x_4,x_5)_1] \\ \hline [(0,0)_1,(0,0)_1,(0,0)_1] \\ [(0,0)_1,(0,1)_1,(0,1)_1] \\ [(0,1)_1,(0,0)_1,(0,1)_1] \\ [(0,1)_1,(0,1)_1,(0,0)_1] \\ [(0,0)_1,(1,0)_1,(1,0)_1] \\ [(1,0)_1,(0,0)_1,(1,0)_1] \\ [(1,0)_1,(1,0)_1,(0,0)_1] \\ [(1,0)_1,(1,1)_1,(1,1)_1] \\ [(1,1)_1,(0,0)_1,(1,1)_1] \\ [(1,1)_1,(1,1)_1,(0,0)_1] \\ \hline \end{array}$	$(y_0, y_1)_2$ (0, 0) ₂	Case 7	$\frac{[(x_0, x_1)_2, (x_2, x_3)_2, (x_4, x_5)_2]}{[(0, 0)_2, (0, 0)_2, (0, 0)_2]}$	$(y_0, y_1)_2$ $(0, 0)_2$
	$ \begin{array}{c} ((0,1)_1,(1,0)_1,(1,1)_1] \\ ((0,1)_1,(1,1)_1,(1,0)_1] \\ ((1,0)_1,(0,1)_1,(1,1)_1] \\ ((1,0)_1,(1,1)_1,(0,1)_1] \\ ((1,1)_1,(0,1)_1,(1,0)_1] \\ ((1,1)_1,(1,0)_1,(0,1)_1] \\ ((1,1)_1,(1,0)_1,(0,1)_1] \\ \end{array} $	$(1,0)_2$ $(0,1)_2$		$[(0,0)_2, (0,0)_2, (0,1)_2]$ $[(0,0)_2, (0,1)_2, (0,0)_2]$ $[(0,1)_2, (0,0)_2, (0,0)_2]$ the others	$(0,1)_2$ (1,0) ₂

Table 3. Possible difference transitions for $MC(MC^{-1})$

When the output nibble n_i is the sum of four input nibbles (i.e., $\sum_{0 \le j \le 3} t_i^j = 4$), there are also several cases for the difference transitions of n_i . However, since this does not occur in our applications, we omit the details when the number of input nibbles is greater than three.

So far, we have studied difference propagation rules of SB, SC, and MC. Based on the above difference propagation rules, we can construct MILP models to search for IDs of specific block ciphers.

4 Applications to Midori-64, CRAFT, and SKINNY-64

4.1 Midori-64

Midori is a family of SP-network block ciphers [1]. There are two versions of Midori: Midori-64 and Midori-128. In this paper, we only introduce Midori-64 since we are only concerned about its security. The block size of Midori-64 is 64 bits. The internal state of Midori-64 is represented as a 4×4 array and consists of 16 cells S_0, S_1, \ldots, S_{15} which has the following data structure:

$$S = \begin{pmatrix} S_0 & S_4 & S_8 & S_{12} \\ S_1 & S_5 & S_9 & S_{13} \\ S_2 & S_6 & S_{10} & S_{14} \\ S_3 & S_7 & S_{11} & S_{15} \end{pmatrix},$$

where the size of each cell is 4 bits.

The round function of Midori-64 consists of four steps: SubCell (SB), Shuffle-Cell (SC), MixColumn (MC), and KeyAdd (KA). The round number of Midori-64 is 16, and SC and MC are omitted in the last round. The overview of Midori-64 is shown in Figure 2.

– SB: A non-linear substitution step, where each cell is replaced with another cell by a bijective 4-bit S-box.

– SC: Each cell of the state is permuted as follows:

 $[S_0, S_1, \dots, S_{15}] \longleftarrow [S_0, S_{10}, S_5, S_{15}, S_{14}, S_4, S_{11}, S_1, S_9, S_3, S_{12}, S_6, S_7, S_{13}, S_2, S_8].$

- MC: Left multiply each column of the state S by a 4×4 matrix M over \mathbb{F}_2^4 :

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

- KA: The round key RK_i is XORed with the state S.

- Key schedule: The key size of Midori-64 is 128 bits, and the master-key is denoted as $K = K_0 || K_1$, where K_0 and K_1 are two 64-bit seed keys. For $i = 0, 1, \ldots, 14$, the round keys are $RK_i = K_{(i+1) \mod 2} \oplus \alpha_i$, where α_i is a constant (Since α_i is known, it does not affect the guessing of round keys, we treat $K_{i \mod 2} \oplus \alpha_i$ and $K_{i \mod 2}$ as the same, $0 \le i \le 14$). The whitening key $WK = K_0 \oplus K_1$ is used as the sub-key in the last KA operations and XORed with the plaintext P before the first round of encryption. Similarly, the round keys do not affect the difference propagation.



Fig. 2. Midori-64 encryption algorithm

4.2 6-round Impossible Differential of Midori-64

In this subsection, we show how to model the nibble-wise operations of Midori-64 and present our search strategy for new IDs of Midori-64.



Fig. 3. Impossible differential cryptanalysis of 11-round Midori-64

Based on the difference propagation rules introduced in Section 3, we can construct an MILP model to characterize the difference propagation of Midori-64. We use the *miss-in-the-middle* [6] technique to search for a 6-round ID of Midori-64 with the model \mathcal{M}_1 . More specifically, we take Δ_X and Δ_Y as the input and output differences of the ID and propagate their differences forward and backward as much as possible respectively, while ensuring that there are some contradictions. In our MILP model, the input and output differences of

the ID are Δc_1 and Δc_7 respectively (as shown in Figure 3). If there are some contradictions between Δd_i and Δa_{i+1} $(1 \leq i \leq 6)$, then the MILP model can return all state difference patterns which constitute an ID. Otherwise, return **infeasible** which means we fail to find a contradiction. We use two-dimension binary variables $(a_i^0[u], a_i^1[u]), (b_i^0[u], b_i^1[u]), (c_j^0[u], c_j^1[u]), and <math>(d_j^0[u], d_j^1[u])$ to represent the state difference patterns after AK, SB, SC, and MC in the MILP model, respectively, where *i* and *j* $(0 \leq i \leq 10, 0 \leq j \leq 9)$ represent the number of rounds and the arrangement of u $(0 \leq u \leq 15)$ is the same as the state (Since AK does not affect the difference propagation, we have $\Delta d_i = \Delta a_{i+1}, 0 \leq i \leq 9$).

As shown in Figure 3, we take the contradiction between Δd_4 and Δa_5 as an example to illustrate the construction of \mathcal{M}_1 . However, we need to determine the difference pattern types first. In our impossible differential cryptanalysis, we consider the cases where the input and output differences of an ID are fixed values. In order to avoid too many computations, the active nibbles of the input and output differences are restricted to at most three distinct values. Thus, the input difference $(c_1^0[u], c_1^1[u])$ and output difference $(c_7^0[u], c_7^1[u])$ of the ID belong to Type-1 difference patterns ($0 \leq u \leq 15$). In the following, we analyze the types of intermediate difference patterns.

Forward difference propagation. Since one nibble of the output difference of MC is unknown (either zero or nonzero) when the corresponding three input nibbles have nonzero and unequal differences, we need to use a Type-2 difference pattern to characterize $(d_1^0[u], d_1^1[u])$. Thus, $(a_i^0[u], a_i^1[u]), (b_i^0[u], b_i^1[u]), (c_i^0[u], c_i^1[u]), (d_i^0[u], d_i^1[u])$ belong to Type-2, where $2 \le i \le 4$ and $0 \le u \le 15$.

Backward difference propagation. Since $(c_7^0[u], c_7^1[u])$ is a Type-1 difference pattern and SC^{-1} is a nibble-wise permutation, we only need to permute the state difference patterns accordingly. Thus, $(b_7^0[u], b_7^1[u])$ belongs to Type-1 difference patterns. Since the output difference of SB^{-1} can take more than three nonzero values when the input difference of SB^{-1} is nonzero, we need to use a Type-2 difference pattern to characterize $(a_7^0[u], a_7^1[u])$ (a Type-1 difference pattern can only characterize three nonzero differences). Thus, $(a_i^0[u], a_i^1[u])$, $(b_i^0[u], b_i^1[u])$, $(c_i^0[u], c_i^1[u])$, $(d_i^0[u], d_i^1[u])$ also belong to Type-2, where $5 \le i \le 6$ and $0 \le u \le 15$.

Once we have determined the types of difference patterns that appeared in the ID, we can choose appropriate transition rules to describe SB, SC, and MC for each round, as shown in Table 4.

After the model characterizing the difference propagation of 6-round Midori-64 is constructed, we need to make sure that there is a contradiction. In other words, there exists an u such that $\Delta d_4[u] = 0$ and $\Delta a_5[u] \neq 0$, or $\Delta d_4[u] \neq 0$ and $\Delta a_5[u] = 0$, for $0 \leq u \leq 15$. In either case, we have successfully detected a contradiction. All cases of $((d_4^0[u], d_4^1[u])_2, (a_5^0[u], a_5^1[u])_2)$ are shown in Table 5, where the dummy variable t[u] is an indicator of contradictions $(0 \leq u \leq 15)$. When t[u] = 1, $(d_4^0[u], d_4^1[u])_2$ and $(a_5^0[u], a_5^1[u])_2$ constitute a valid contradiction. When t[u] = 0, $(d_4^0[u], d_4^1[u])_2$ and $(a_5^0[u], a_5^1[u])_2$ do not contradict each other.

Cipher	Direction	$\mathbf{Input} \to \mathbf{Output}$	$\mathbf{Input} ightarrow \mathbf{Output}$	Propagation rule
		$\Delta c_1 \xrightarrow{MC} \Delta d_1$	Type-1 \xrightarrow{MC} Type-2	Case 6 of MC
	Forward	$\Delta a_i \xrightarrow{SB} \Delta b_i, 2 \le i \le 4$	Type-2 \xrightarrow{SB} Type-2	Case 2 of SB
		$\Delta b_i \xrightarrow{SC} \Delta c_i, 2 \le i \le 4$	Type-2 \xrightarrow{SC} Type-2	Case 2 of SC
		$\Delta c_i \xrightarrow{MC} \Delta d_i, 2 \le i \le 4$	Type-2 \xrightarrow{MC} Type-2	Case 7 of MC
Midori-64		$\Delta c_7 \xrightarrow{SC^{-1}} \Delta b_7$	Type-1 $\xrightarrow{SC^{-1}}$ Type-1	Case 1 of SC^{-1}
	Backward	$\Delta b_7 \xrightarrow{SB^{-1}} \Delta a_7$	Type-1 $\xrightarrow{SB^{-1}}$ Type-2	Case 1 of SB^{-1}
	Dackward	$\Delta d_i \stackrel{MC^{-1}}{\longrightarrow} \Delta c_i, 5 \le i \le 6$	Type-2 $\xrightarrow{MC^{-1}}$ Type-2	Case 7 of MC^{-1}
		$\Delta c_i \xrightarrow{SC^{-1}} \Delta b_i, 5 \le i \le 6$	Type-2 $\xrightarrow{SC^{-1}}$ Type-2	Case 2 of SC^{-1}
		$\Delta b_i \xrightarrow{SB^{-1}} \Delta a_i, 5 \le i \le 6$	Type-2 $\xrightarrow{SB^{-1}}$ Type-2	Case 2 of SB^{-1}
		$\Delta b_0 \xrightarrow{SC} \Delta c_0$	Type-1 \xrightarrow{SC} Type-1	Case 1 of SC
		$\Delta c_0 \xrightarrow{SB} \Delta a_1$	Type-1 \xrightarrow{SB} Type-2	Case 1 of SB
	Forward	$\Delta a_i \xrightarrow{MC} \Delta b_i, 1 \le i \le 6$	Type-2 \xrightarrow{MC} Type-2	Case 7, 5, 3, 3 of MC †
		$\Delta b_i \xrightarrow{SC} \Delta c_i, 1 \le i \le 6$	Type-2 \xrightarrow{SC} Type-2	Case 2 of SC
CRAFT		$\Delta c_i \xrightarrow{SB} \Delta a_{i+1}, 1 \le i \le 5$	Type-2 \xrightarrow{SB} Type-2	Case 2 of SB
	Backward	$\Delta b_{13} \xrightarrow{MC^{-1}} \Delta a_{13}$	Type-1 $\xrightarrow{MC^{-1}}$ Type-2	Case 6, 4, 2, 2 of MC^{-1}
		$\Delta c_i \xrightarrow{SC^{-1}} \Delta b_i, 7 \le i \le 12$	Type-2 $\xrightarrow{SC^{-1}}$ Type-2	Case 2 of SC^{-1}
		$\Delta b_i \xrightarrow{MC^{-1}} \Delta a_i, 7 \le i \le 12$	Type-2 $\xrightarrow{MC^{-1}}$ Type-2	Case 7, 5, 3, 3 of MC^{-1}
		$\Delta a_i \xrightarrow{SB^{-1}} \Delta c_{i-1}, 8 \le i \le 13$	Type-2 $\xrightarrow{SB^{-1}}$ Type-2	Case 2 of SB^{-1}
		$\Delta a_0 \xrightarrow{SB} \Delta b_0$	Type-1 \xrightarrow{SB} Type-2	Case 1 of SB
SKINNY-64		$\Delta b_i \xrightarrow{SC} \Delta c_i, 0 \le i \le 5$	Type-2 \xrightarrow{SC} Type-2	Case 2 of SC
	Forward	$\Delta c_i \xrightarrow{MC} \Delta a_{i+1}, 0 \le i \le 5$	Type-2 \xrightarrow{MC} Type-2	Case 7, 3, 5, 5 of MC
		$\Delta a_i \xrightarrow{SB} \Delta b_i, 1 \le i \le 5$	Type-2 \xrightarrow{SB} Type-2	Case 2 of SB
		$\Delta a_{12} \xrightarrow{MC^{-1}} \Delta c_{11}$	Type-1 $\xrightarrow{MC^{-1}}$ Type-2	Case 2, 6, 4, 4 of MC^{-1}
	Backward	$\Delta c_i \stackrel{SC^{-1}}{\longrightarrow} \Delta b_i, 6 \le i \le 11$	Type-2 $\xrightarrow{SC^{-1}}$ Type-2	Case 2 of SC^{-1}
		$\Delta a_i \xrightarrow{MC^{-1}} \Delta c_{i-1}, 7 \le i \le 11$	Type-2 $\xrightarrow{MC^{-1}}$ Type-2	Case 3, 7, 5, 5 of MC^{-1}
		$\Delta b_i \xrightarrow{SB^{-1}} \Delta a_i, 7 \le i \le 11$	Type-2 $\xrightarrow{SB^{-1}}$ Type-2	Case 2 of SB^{-1}

Table 4. The difference propagation rules for IDs of 6-round Midori-64, 13-roundCRAFT, and 12-round SKINNY-64

 † When the state performs the MC operation, the propagation rules for the output nibbles of the first, second, third, and fourth rows are Case 7, Case 5, Case 3, and Case 3 of MC, respectively.

Input $(d_4^0[u], d_4^1[u])_2$	Output $(a_{\varepsilon}^{0}[u], a_{\varepsilon}^{1}[u])_{2}$	Indicator $t[u]$
$(0,0)_2$	$(0,1)_2$	1
$(0,1)_2$	$(0,0)_2$	1
the	0	

 Table 5. Characterize contradictions

The following linear inequalities are sufficient to describe these contradictions:

$$\begin{cases} -d_4^0[u] - d_4^1[u] - a_5^0[u] - a_5^1[u] - t[u] + 2 \ge 0, \\ d_4^1[u] + a_5^1[u] - t[u] \ge 0, \\ d_4^0[u] + d_4^1[u] - a_5^1[u] + t[u] \ge 0, \\ -d_4^1[u] + a_5^0[u] + a_5^1[u] + t[u] \ge 0. \end{cases}$$
(1)

Put all the constraints together, the complete model \mathcal{M}_1 can be constructed, which is composed of the following inequalities.

Constraints of \mathcal{M}_1 :

1. The constraints on the input and output differences Δc_1 and Δc_7 . Note that at least one nibble of both the input and output differences should be active:

$$\sum_{0 \le u \le 15} (c_1^0[u] + c_1^1[u]) \ge 1,$$
$$\sum_{0 \le u \le 15} (c_7^0[u] + c_7^1[u]) \ge 1.$$

- 2. Construct the constraints of the difference propagation $(\Delta c_1 \rightarrow \Delta d_4, \Delta c_7 \rightarrow \Delta a_5)$ according to Table 4.
- 3. The constraints to ensure a contradiction, i.e., there is at least one u ($0 \le u \le 15$) such that a contradiction exists between $\Delta d_4[u]$ and $\Delta a_5[u]$.

$$\sum_{0\leq u\leq 15}t[u]\geq 1,$$

where t[u] is an indicator as shown in Inequality 1.

Note that we do not set any objective function for \mathcal{M}_1 , and the automatic tool Gurobi⁴ is utilized to check if \mathcal{M}_1 is **feasible**. If \mathcal{M}_1 is **feasible**, it indicates that there are valid 6-round IDs, and any valid solution constitutes the full state difference patterns. However, if \mathcal{M}_1 is **infeasible**, there are no valid 6-round IDs under such constraints. In this case, we can also check whether there is a valid ID by changing the position of the contradictions and updating \mathcal{M}_1 . As presented in Table 6, the experimental results show that 6-round IDs

⁴ http://www.gurobi.com/

Cipher	The position of the contradiction	Status	#ID
Midori-64	$d_i \nleftrightarrow a_{i+1}, i \in \{1, 2, 5, 6\}$	infeasible	0
	$d_j \nleftrightarrow a_{j+1}, j \in \{3,4\}$	feasible	10752
CRAFT	$c_i \nleftrightarrow a_{i+1}, i \in \{0, 1, 2, 3, 4, 8, 9, 10, 11, 12, 13\}$	infeasible	0
UNAFI	$c_j \nleftrightarrow a_{j+1}, j \in \{5, 6, 7\}$	feasible	12
SKINNY-64	$a_i \nleftrightarrow b_i, i \in \{0, 1, 2, 3, 4, 7, 8, 9, 10, 11\}$	infeasible	0
	$a_j \nleftrightarrow b_j, j \in \{5, 6\}$	feasible	8

Table 6. The number of IDs and the status of \mathcal{M}_1 for different positions of contradictions

exist when the contradiction position is located between d_4 and a_5 or between d_3 and a_4 .

The model \mathcal{M}_1 can also be used to search for the number of IDs. The general procedure can be divided into the following steps:

- Step 1. For a fixed position of contradiction i, build an MILP model \mathcal{M}_1 .
- Step 2. Optimize the model \mathcal{M}_1 and obtain the number of IDs N_i .
- Step 3. Obtain the values of the variables representing the input and output difference patterns and store them in S.
- Step 4. Replace the position of contradiction by i+1, update the model \mathcal{M}_1 , and repeat Step 1-3 until the position of contradiction has been traversed.
- Step 5. Evaluate the number of duplicated elements in S, denoted as N_r , then the number of IDs is $\sum_i N_i N_r$.

For 6-round Midori-64, we only consider the case where the active input differences are equal and the active output differences are equal, when the contradiction position is located between d_4 and a_5 or between d_3 and a_4 , the number of corresponding IDs is 10752 and 10752, respectively. Furthermore, we found that 256 IDs are the same by comparing the specific input and output difference patterns of these IDs, so the total number of IDs for 6-round Midori-64 is 21248. An example of 6-round IDs of Midori-64 is shown in Figure 3.

4.3 CRAFT and SKINNY-64

CRAFT is a 32-round iterative tweakable block cipher proposed at FSE 2019 [3], and it consists of a 64-bit block, a 128-bit key, and a 64-bit tweak. The data structure of CRAFT's internal state is the same as Midori-64. The round function of CRAFT consists of five steps: MixColumn (MC), AddConstants (AC), AddTweakey (ATK), PermuteNibbles (SC), and SubBox (SB).

- MC: Left multiply each column of the state S by a 4×4 matrix M over \mathbb{F}_2^4 :

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

– SC: Each cell of the state is permuted as follows:

 $[S_0, S_1, \dots, S_{15}] \longleftarrow [S_{15}, S_{10}, S_9, S_4, S_3, S_6, S_5, S_8, S_7, S_2, S_1, S_{12}, S_{11}, S_{14}, S_{13}, S_0].$

Since the AddConstants and AddTweakey do not affect the difference propagation, we omit the details of these operations. Moreover, as the only condition we used to model the PermuteNibbles is that it is a permutation, we do not list the truth table of the S-box for the sake of brevity.

The block cipher family SKINNY was presented at CRYPTO 2016 [2], and the SKINNY family consists of 6 different members represented as SKINNY-*n*-*t*, where n = 64, 128 and t = n, 2n, 3n, which respectively represent the sizes of the block and tweakey. In this paper, we only introduce SKINNY-64 since we are only concerned about its security. The data structure of SKINNY-64's internal state is the same as Midori-64. The round function of SKINNY-64 consists of five steps: SubCells (SB), AddConstants (AC), AddRoundTweakey (ART), ShiftRows (SC), and MixColumns (MC).

- SC: The second, third, and fourth rows are rotated by 1, 2 and 3 cell positions to the right, respectively.

- MC: Left multiply each column of the state S by a 4×4 matrix M over \mathbb{F}_2^4 :

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

We also omit the details of AC, ART, and SB operations.

The modeling procedure of CRAFT and SKINNY-64 is very similar to the one for Midori-64. The transition rules of each model to describe SB, SC, and MC for each round are shown in Table 4. Moreover, as presented in Table 6, when the contradiction position is located between c_i and a_{i+1} ($5 \le i \le 7$), there are 13-round IDs of CRAFT. When the contradiction position is located between a_i and b_i ($5 \le i \le 6$), there are 12-round IDs of SKINNY-64. For 13-round CRAFT and 12-round SKINNY-64, we consider all possible input differences and output differences. Excluding the same IDs, the total number of IDs is 12 and 12 for the 13-round CRAFT and the 12-round SKINNY-64, respectively. The example IDs of 13-round CRAFT and 12-round SKINNY-64 are shown in Appendix A.

5 Impossible Differential Cryptanalysis of 11-round Midori-64

After a valid ID has been found, we can extend this ID in both directions to obtain an IDA. Note that the time complexity of an IDA is closely related to the number of keys that need to be guessed. However, the number of keys that need to be guessed is different when using different IDs to mount an IDA. Thus, in order to decrease the complexity, we need to minimize the number of guessed

17

keys. Manually deriving the optimal IDA is currently a common strategy, but this method is limited by the number of IDs.

According to our results in Section 4, we found 6-round, 13-round, and 12round IDs for Midori-64, CRAFT, and SKINNY-64, respectively. Thus, it is more feasible to construct the MILP model to search for the IDA with the minimum number of guessed keys, especially for Midori-64 as we have a large number of IDs. On the other hand, we found 13-round IDs for CRAFT and 12-round IDs for SKINNY-64. The round number of both two ciphers is 32. It seems that IDA is far from threatening their security. However, the round number of Midori-64 is 16, and the IDs can reach 6 rounds. This small gap makes it a more serious threat to the security of Midori-64. Thus, in the following subsection, we model extended rounds of Midori-64 to achieve the best IDA with the minimum number of guessed keys.

5.1 Modeling the Extended Rounds for Midori-64

We first analyze the types of state difference patterns in those additional rounds and choose the appropriate difference propagation rules of SB, SC, and MC accordingly.

Backward difference propagation. Since $(c_1^0[u], c_1^1[u])$ is a Type-1 difference pattern and SC^{-1} is a nibble-wise permutation, we only need to permute the state difference patterns accordingly. Thus, $(b_1^0[u], b_1^1[u])$ belongs to Type-1 difference patterns. Since the output difference of SB^{-1} can take more than three nonzero values when the input difference of SB^{-1} is nonzero, we need to use a Type-2 difference pattern to characterize $(a_1^0[u], a_1^1[u])$ (a Type-1 difference pattern can only characterize three nonzero differences). Thus, $(a_i^0[u], a_i^1[u])$, $(b_i^0[u], b_i^1[u])$, $(c_i^0[u], c_i^1[u]), (d_i^0[u], d_i^1[u])$ also belong to Type-2, where i = 0 and $0 \le u \le 15$.

Forward difference propagation. Since one nibble of the output difference of MC is unknown (either zero or nonzero) when the corresponding three input nibbles have nonzero and unequal differences, we need to use a Type-2 difference pattern to characterize $(d_7^0[u], d_7^1[u])$. Thus, $(a_i^0[u], a_i^1[u])$, $(b_i^0[u], b_i^1[u])$, $(c_j^0[u], c_j^1[u])$, $(d_j^0[u], d_j^1[u])$ belong to Type-2, where $8 \le i \le 10, 8 \le j \le 9$ and $0 \le u \le 15$.

Once we have determined the types of the state difference patterns in the extended rounds, we can choose appropriate difference propagation rules to characterize SB, SC, and MC as shown in Table 7.

The second step for an IDA is to guess the keys k_{in} and k_{out} involved in those additional rounds. Since the time complexity T_N will increase with the increase of $|k_{in} \cup k_{out}|$, in order to decrease the complexity T_N , we need to minimize $|k_{in} \cup k_{out}|$. Guessing the equivalent key is a common way to reduce the number of guessed keys. In this paper, we swap KA with MC. Then, the equivalent key K' after the swap operation and original key K have the following relations:

$$K = M \cdot K',$$

Direction	$\mathbf{Input} \to \mathbf{Output}$	$\textbf{Input} \rightarrow \textbf{Output}$	Propagation rule
	$\Delta c_1 \xrightarrow{SC^{-1}} \Delta b_1$	Type-1 $\xrightarrow{SC^{-1}}$ Type-1	Case 1 of SC^{-1}
	$\Delta b_1 \xrightarrow{SB^{-1}} \Delta a_1$	Type-1 $\xrightarrow{SB^{-1}}$ Type-2	Case 1 of SB^{-1}
Backward	$\Delta d_i \stackrel{MC^{-1}}{\longrightarrow} \Delta c_i, i = 0$	Type-2 $\xrightarrow{MC^{-1}}$ Type-2	Case 7 of MC^{-1}
	$\Delta c_i \xrightarrow{SC^{-1}} \Delta b_i, i = 0$	Type-2 $\xrightarrow{SC^{-1}}$ Type-2	Case 2 of SC^{-1}
	$\Delta b_i \stackrel{SB^{-1}}{\longrightarrow} \Delta a_i, i = 0$	Type-2 $\xrightarrow{SB^{-1}}$ Type-2	Case 2 of SB^{-1}
	$\Delta c_7 \xrightarrow{MC} \Delta d_7$	Type-1 \xrightarrow{MC} Type-2	Case 6 of MC
Forward	$\Delta a_i \xrightarrow{SB} \Delta b_i, 8 \le i \le 10$	Type-2 \xrightarrow{SB} Type-2	Case 2 of SB
	$\Delta b_i \xrightarrow{SC} \Delta c_i, 8 \le i \le 9$	Type-2 \xrightarrow{SC} Type-2	Case 2 of SC
	$\Delta c_i \xrightarrow{MC} \Delta d_i, 8 \le i \le 9$	Type-2 \xrightarrow{MC} Type-2	Case 7 of MC

Table 7. The difference propagation rules for the extended rounds

$$K' = M^{-1} \cdot K.$$

Correspondingly, the equivalent round key RK'_i and the original round key RK_i (we denote the equivalent seed keys of original seed keys K_0 and K_1 by K'_0 and K'_1 , respectively) have the following relations:

$$(RK_{i}[4j], \dots, RK_{i}[4j+3])^{T} = M \cdot (RK'_{i}[4j], \dots, RK'_{i}[4j+3])^{T},$$
$$(RK'_{i}[4j], \dots, RK'_{i}[4j+3])^{T} = M^{-1} \cdot (RK_{i}[4j], \dots, RK_{i}[4j+3])^{T},$$

where $i \ (0 \le i \le 14)$ is the round number, and $j \ (0 \le j \le 3)$ is the column index of RK_i and RK'_i . Moreover, we have:

$$RK_{i}[4j+t] = \bigoplus_{1 \le s \le 3} RK'_{i}[4j+(t+s) \mod 4],$$
$$RK'_{i}[4j+t] = \bigoplus_{1 \le s \le 3} RK_{i}[4j+(t+s) \mod 4],$$

where $0 \le i \le 14, 0 \le j \le 3, 0 \le t \le 3$.

Equivalent key transform technique. Denote the multiset of all key nibbles that need to be guessed in the IDA as Guessed Key Multiset (GKM). Since the key schedule of Midori-64 is linear, the key nibbles in the GKM \mathbb{K} can be expressed as linear combinations of nibbles in K_0 and K_1 . Furthermore, we can find a corresponding coefficient matrix T, such that all guessed nibbles can be expressed as a matrix-vector multiplication.

Note that, in an IDA, an attacker always extends the distinguisher forward and backward for several rounds. As shown in Figure 3, the distinguisher is extended 1.5 rounds forward and 3.5 rounds backward, and an attacker has to guess the keys involved in these rounds to check the input and output differences of the distinguisher, which usually has few active nibbles. Intuitively, the number of key nibbles involved in an inner round is less than those involved in an outer round. For example, the key nibbles involved in the 9-th round are less than the key nibbles involved in the 10-th round as shown in Figure 3. Since we can swap AK and MC and the MC operation has a strong diffusion effect, guessing the equivalent keys of the 9-th and 10-th rounds is more advantageous for attackers, as the nibbles that need to be computed before the MC operation are less than those after the MC operation. Thus, we have the following theorem (The proof of Theorem 1 is shown in Appendix B).

Theorem 1. Only swapping KA with MC for the 9-th and 10-th rounds, that is, guessing equivalent key nibbles of RK'_8 and RK'_9 , and guessing original key nibbles of WK and RK_0 , the corresponding coefficient matrix T can take the minimum rank.

Based on Theorem 1, we set the objective function of the model \mathcal{M}_2 as the minimum number of key nibbles to be guessed, that is,

$$obj(\mathcal{M}_2) = \min\left\{\sum_{0 \le u \le 15} (GK_0[u] + GK_1[u] + GK_2[u] + GK_3[u])\right\},\$$

where $GK_0[u]$, $GK_1[u]$, $GK_2[u]$, and $GK_3[u]$ indicate whether the key nibbles WK[u], $K'_1[u]$ (i.e., RK'_9), $K_0[u]$ (i.e., RK_0), and $K'_0[u]$ (i.e., RK'_8) ($0 \le u \le 15$) need to be guessed, respectively.

Since WK and K'_1 are linearly independent, we can first guess the key nibbles in WK and K'_1 and do not care about the linear relations between them. As WK is used as the whitening key and the last round key, for any position u $(0 \le u \le 15)$, if either $\Delta P[u]$ or $\Delta b_{10}[u]$ is active or unknown, then $GK_0[u] = 1$, else $GK_0[u] = 0$. Similarly, K'_1 is used as the equivalent key of the 9-th round, thus $GK_1[u] = 1$ if $\Delta c_9[u]$ is active or unknown. However, when guessing K_0 and K'_0 , we should be careful since we may be able to calculate several nibbles of K_0 and K'_0 according to the linear relations between WK, K'_1 , K_0 , and K'_0 . Thus, the linear relations between WK, K'_1 , K_0 , and K'_0 should be considered to minimize the number of guessed key nibbles. In the following, we denote

 $\mathbb{C}^{u} = \{(1+u_1) \mod 4 + u_2, (2+u_1) \mod 4 + u_2, (3+u_1) \mod 4 + u_2\},\$

where $0 \le u \le 15, u_1 = u \mod 4, u_2 = u - u_1$.

Relations between K'_0 and WK, K'_1 (**R1**): If $GK_1[u] = 1$, we can deduce that $\Delta d_9[u]$ and $\Delta a_{10}[i]$ $(i \in \mathbb{C}^u)$ are all active or unknown, i.e., $GK_0[i] = 1$ for $i \in \mathbb{C}^u$. Then we can calculate $K'_0[u]$ by $K'_0[u] = \bigoplus_{i \in \mathbb{C}^u} WK[i] \oplus K'_1[u]$, i.e., $K'_0[u]$ does not need to be guessed. Then, a new array of binary variables $v_s[i]$ $(0 \le i \le 15)$ is introduced to temporarily indicate whether the corresponding nibbles of K'_0 need to be guessed, and the reason will be explained later. So, for any position u $(0 \le u \le 15)$, $v_s[u] = 1$ if $\Delta c_8[u]$ is active or unknown and $GK_1[u] = 0$, else $v_s[u] = 0$. The following linear inequalities are sufficient to

describe this rule:

$$\begin{cases} GK_1[u] - c_8^0[u] - c_8^1[u] + v_s[u] \ge 0, \\ -GK_1[u] - v_s[u] + 1 \ge 0, \\ c_8^0[u] + c_8^1[u] - v_s[u] \ge 0. \end{cases}$$

Relations between K_0 and WK, K'_1 (**R2**): If $GK_1[i] = 1$ for all $i \in \mathbb{C}^u$, we can deduce that $\Delta a_{10}[u]$ and $\Delta d_9[i]$ ($i \in \mathbb{C}^u$) are all active or unknown, i.e., $GK_0[u] = 1$. Thus we can calculate $K_0[u]$ by $K_0[u] = \bigoplus_{i \in \mathbb{C}^u} K'_1[i] \oplus WK[u]$, i.e., $K_0[u]$ does not need to be guessed. Similarly, a new array of binary variables $v_t[i]$ ($0 \leq i \leq 15$) is introduced to temporarily indicate whether the corresponding nibbles of K_0 need to be guessed. So, for any position u ($0 \leq u \leq 15$), $v_t[u] = 1$ if there exist at least one $i \in \mathbb{C}^u$ such that $GK_1[i] = 0$ and $\Delta d_0[u]$ is active or unknown, else $v_t[u] = 0$. The following linear inequalities are sufficient to describe this rule:

$$\begin{cases} d_0^0[u] + d_0^1[u] - v_t[u] \ge 0, \\ GK_1[i_2] - d_0^0[u] - d_0^1[u] + v_t[u] \ge 0, \\ GK_1[i_1] - d_0^0[u] - d_0^1[u] + v_t[u] \ge 0, \\ - GK_1[i_0] - GK_1[i_1] - GK_1[i_2] - v_t[u] + 3 \ge 0, \\ GK_1[i_0] - d_0^0[u] - d_0^1[u] + v_t[u] \ge 0. \end{cases}$$

Relations between K_0 and K'_0 (**R3**): If $v_s[i] = 1$ for all $i \in \mathbb{C}^u$, we can calculate $K_0[u]$ by $K_0[u] = \bigoplus_{i \in \mathbb{C}^u} K'_0[i]$, i.e., $K_0[u]$ does not need to be guessed. Thus, for any position u ($0 \le u \le 15$), $GK_2[u] = 1$ if $v_t[u] = 1$ and there exist at least one $i \in \mathbb{C}^u$ such that $v_s[i] = 0$, else $GK_2[u] = 0$. The following linear inequalities are sufficient to describe this rule:

$$\begin{cases} v_t[u] - GK_2[u] \ge 0, \\ v_s[i_0] - v_t[u] + GK_2[u] \ge 0, \\ v_s[i_1] - v_t[u] + GK_2[u] \ge 0, \\ - v_s[i_0] - v_s[i_1] - v_s[i_2] - GK_2[u] + 3 \ge 0 \\ v_s[i_2] - v_t[u] + GK_2[u] \ge 0. \end{cases}$$

Relations between K'_0 and K_0 (**R4**): If $GK_2[i] = 1$ for all $i \in \mathbb{C}^u$, we can calculate $K'_0[u]$ by $K'_0[u] = \bigoplus_{i \in \mathbb{C}^u} K_0[i]$, i.e., $K'_0[u]$ does not need to be guessed. Thus, for any position u ($0 \le u \le 15$), $GK_3[u] = 1$ if $v_s[u] = 1$ and there exist at least one $i \in \mathbb{C}^u$ such that $GK_2[i] = 0$, else $GK_3[u] = 0$. The following linear inequalities are sufficient to describe this rule:

$$\begin{cases} v_s[u] - GK_3[u] \ge 0, \\ GK_2[i_0] - v_s[u] + GK_3[u] \ge 0, \\ GK_2[i_1] - v_s[u] + GK_3[u] \ge 0, \\ - GK_2[i_0] - GK_2[i_1] - GK_2[i_2] - GK_3[u] + 3 \ge 0, \\ GK_2[i_2] - v_s[u] + GK_3[u] \ge 0. \end{cases}$$

Put the above constraints together, \mathcal{M}_2 can be constructed, which is composed of the following inequalities.

Constraints of \mathcal{M}_2 :

- 1. Construct the constraints of the difference propagation $(\Delta c_1 \rightarrow \Delta a_0, \Delta c_7 \rightarrow \Delta b_{10})$ according to Table 7.
- 2. Construct the constraints of the objective function according to R1 R4.

When the contradiction position is located between d_3 and a_4 , the objective value of \mathcal{M}_2 is 26. However, when the contradiction position is located between d_4 and a_5 , the objective value of \mathcal{M}_2 is 23. That is, the minimum number of key nibbles to be guessed in the 11-round IDA of Midori-64 is 23.

Discussion. When modeling extended rounds of Midori-64, we need to characterize the number of guessed keys. So, we have to discuss the connection between the equivalent key and the original key to avoid repeated keys. Compared with Midori-64, SKINNY-64's key schedule involves a nibble permutation, and the MixColumn matrix is not a circulant matrix. This makes it difficult to describe the relations between the equivalent key and the original key. In addition, the key schedule of CRAFT involves a 64-bit tweak, and the influence of both the 128-bit key and the 64-bit tweak needs to be considered when constructing the model. In other words, IDA is related to the structural properties of both the cipher itself and the key schedule, and the simple key schedule of Midori-64 makes it possible to describe the number of guessed keys with linear inequalities.

5.2 Impossible Differential Cryptanalysis of 11-round Midori-64

In this subsection, we present a new IDA on 11-round Midori-64. The overview of our attack is illustrated in Figure 4, which consists of the following steps:

- 1. Take a group of 2^{36} plaintexts as a structure that traverses all possible values at the nibble positions of (0, 1, 4, 5, 6, 9, 10, 12, 14), and fixes the remaining nibbles as any constant. A structure consists of approximately $2^{36} \times 2^{35} = 2^{71}$ plaintext pairs. We prepare 2^n structures that differ in the constant values, thus there are 2^{n+36} plaintexts and 2^{n+71} plaintext pairs. Encrypt these 2^{n+36} plaintexts for 11 rounds to obtain the corresponding ciphertexts. For each pair of ciphertexts within the same structure, we reserve the pair that has a zero difference at positions (3, 11) and a nonzero difference at positions (0, 1, 2, 4, 5, 8, 9, 10, 12, 13). Thus, there remain approximately $2^{n+71-2\times 4} =$ 2^{n+63} pairs and we store them in table Ω_0 .
- 2. Guess $\Delta d_9[3]$, since there are 2^4 possible values for $\Delta d_9[3]$, we can compute the corresponding 2^4 values for $\Delta a_{10}[0, 1, 2]$. Given two nonzero differences Δ_{in} and Δ_{out} in \mathbb{F}_2^4 , the equation $\mathsf{SB}(x) \oplus \mathsf{SB}(x \oplus \Delta_{in}) = \Delta_{out}$ has one solution on average (This property also holds for the inverse of SB , i.e., SB^{-1}). So, we can get 1 solution, on average, for $a_{10}[0, 1, 2]$, $b_{10}[0, 1, 2]$ and $d_9[3]$. Then, we can calculate WK[0, 1, 2] by $WK[0, 1, 2] = b_{10}[0, 1, 2] \oplus C[0, 1, 2]$. Create a



Fig. 4. Impossible differential cryptanalysis of 11-round Midori-64

table Ω_1 with 2^{12} key values of WK[0, 1, 2] as indexes, and each item stores $2^{n+63} \times 2^4/2^{12} = 2^{n+55}$, on average, plaintext-ciphertext pairs associated with its corresponding values of $(d_9[3], d'_9[3])$.

- 3. For each item in Ω_1 , guess $\Delta d_9[4,5]$. For each of the 2^8 possible values for $\Delta d_9[4,5]$, compute the corresponding 2^8 values for $\Delta a_{10}[4,5,6,7]$. We can get 1 solution, on average, for $a_{10}[4,5,6,7]$, $b_{10}[4,5,6,7]$, and $d_9[4,5]$. Then, we can calculate WK[4,5,6,7] by $WK[4,5,6,7] = b_{10}[4,5,6,7] \oplus C[4,5,6,7]$. Create a table Ω_2 with 2^{16} key values of WK[4,5,6,7] as indexes, and each item stores $2^{n+55} \times 2^8/2^{16} = 2^{n+47}$, on average, plaintext-ciphertext pairs associated with its corresponding values of $(d_9[3,4,5], d_9[3,4,5])$.
- 4. For each item in Ω_2 , guess $\Delta d_9[11]$. For each of the 2^4 possible values for $\Delta d_9[11]$, compute the corresponding 2^4 values for $\Delta a_{10}[8,9,10]$. We can get 1 solution, on average, for $a_{10}[8,9,10]$, $b_{10}[8,9,10]$, and $d_9[11]$. Then, we calculate WK[8,9,10] by $WK[8,9,10] = b_{10}[8,9,10] \oplus C[8,9,10]$. Create a table Ω_3 with 2^{12} key values of WK[8,9,10] as indexes, and each item stores

23

 $2^{n+47} \times 2^4/2^{12} = 2^{n+39}$, on average, plaintext-ciphertext pairs associated with its corresponding values of $(d_9[3, 4, 5, 11], d'_9[3, 4, 5, 11])$.

- 5. For each item in Ω_3 , guess $\Delta d_9[12, 13]$. For each of the 2⁸ possible values for $\Delta d_9[12, 13]$, compute the corresponding 2⁸ values of $\Delta a_{10}[12, 13, 14, 15]$. We can get 1 solution, on average, for $a_{10}[12, 13, 14, 15]$, $b_{10}[12, 13, 14, 15]$, and $d_9[12, 13]$. Then, we can calculate WK[12, 13, 14, 15] by $WK[12, 13, 14, 15] = b_{10}[12, 13, 14, 15] \oplus C[12, 13, 14, 15]$. Create a table Ω_4 with 2¹⁶ key values of WK[12, 13, 14, 15] as indexes, and each item stores $2^{n+39} \times 2^8/2^{16} = 2^{n+31}$, on average, plaintext-ciphertext pairs associated with its corresponding values of $(d_9[3, 4, 5, 11, 12, 13], d'_9[3, 4, 5, 11, 12, 13])$.
- 6. For each item in Ω_4 , guess $\Delta d_8[5, 12]$. For each of the 2⁸ possible values for $\Delta d_8[5, 12]$, compute the corresponding 2⁸ values for $\Delta a_9[4, 6, 7, 13, 14, 15]$. Since Δb_9 can be calculated from Δd_9 , we can calculate $\Delta b_9[4, 6, 7, 13, 14, 15]$ from $\Delta d_9[3, 4, 5, 11, 12, 13]$. We can get 1 solution, on average, for $d_8[5, 12]$, $a_9[4, 6, 7, 13, 14, 15]$, $b_9[4, 6, 7, 13, 14, 15]$, and $c_9[3, 4, 5, 11, 12, 13]$. Then, we can calculate the equivalent round keys $RK'_9[3, 4, 5, 11, 12, 13]$ by $RK'_9[3, 4, 5, 11, 12, 13] = c_9[3, 4, 5, 11, 12, 13] \oplus d_9[3, 4, 5, 11, 12, 13]$. Create a table Ω_5 with 2^{24} key values of $RK'_9[3, 4, 5, 11, 12, 13]$ as indexes, and each item stores $2^{n+31} \times 2^8/2^{24} = 2^{n+15}$, on average, plaintext-ciphertext pairs associated with its corresponding values of $(d_8[5, 12], d'_8[5, 12])$.
- 7. Since $RK'_{9}[5, 12]$, WK[4, 6, 7], and WK[13, 14, 15] are known, we can calculate the equivalent round key nibbles $RK'_{8}[5] = (\bigoplus_{i \in \{4,6,7\}} WK[i]) \bigoplus RK'_{9}[5]$ and $RK'_{8}[12] = (\bigoplus_{i \in \{13,14,15\}} WK[i]) \bigoplus RK'_{9}[12]$. Moreover, $\Delta d_{7}[4, 7]$ can also be calculated. Then, we can filter the plaintext-ciphertext pairs by the condition $\Delta d_{7}[4] = \Delta d_{7}[7]$. In this case, each entry of Ω_{5} remains $2^{n+15}/2^{4} = 2^{n+11}$, on average, plaintext-ciphertext pairs.
- 8. Since WK[0, 1, 4, 5, 6, 9, 10, 12, 14] has been guessed, we use the above key to encrypt the plaintext pairs in Table Ω_5 for 1 round and reserve the plaintext pairs that are only active at $d_0[3, 6, 9]$ after 1 round encryption. After this step, each entry of Ω_5 remains $2^{n+11}/2^{24} = 2^{n-13}$, on average, plaintextciphertext pairs.
- 9. Exhaustively enumerate 2^{12} possibles values of $RK_0[3, 6, 9]$ and encrypt the remaining plaintext pairs in Ω_5 to obtain $\Delta b_1[3, 6, 9]$. The probability of $\Delta b_1[3] = \Delta b_1[6] = \Delta b_1[9]$ is 2^{-8} . Since each entry of Ω_5 contains 2^{n-13} plaintext-ciphertext pairs, the probability that the entry in Ω_5 being empty is $(1 2^{-8})^{2^{n-13}}$. For each empty entry in Ω_5 , iteratively retrieve its index from Ω_5 to Ω_1 . These indexes ($K_0 \oplus K_1[0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15]$, $K_0[3, 6, 9]$, $K'_1[3, 4, 5, 11, 12, 13]$) constitute a valid key candidate of the key nibbles, since all plaintext-ciphertext pairs cannot be encrypted (decrypted) by these keys to obtain the intermediate state difference that matches the ID. After this sieving process, the expected number of candidate keys is $NK = 2^{92} \times (1 2^{-8})^{2^{n-13}}$.
- 10. Exhaustively enumerate the candidate keys returned by the above steps and guess the remaining 9 key nibbles WK[3, 11] and $K'_1[0, 1, 6, 8, 9, 14, 15]$, check if the keys are correct by one encryption.

Complexity: From Step 2 - 9, we have guessed $4 \times 23 = 92$ key bits. Hence, the expected number of candidate keys is $NK = 2^{92} \times (1 - 2^{-8})^{2^{n-13}}$. Table 8 summarizes the time and data complexity of each step. The total time complexity is:

$$2^{n+36} + (2^4 \times 2^{n+63} \times \frac{3}{16})/11 + 2^{12}(2^{n+63} \times \frac{3}{16} + 2^{16}(2^{n+51} \times \frac{3}{16} + 2^{12}(2^{n+47} \times \frac{3}{16} + 2^{16}(2^{n+39} \times 1 + 2^{24}(2^{n+15} + 2^{n+11} + 2^{n-1} \times \frac{2}{4})))))/11 + 2^{128}(1 - 2^{-8})^{2^{n-13}}$$

$$= 2^{n+36} + 2^0(2^{n+67} \times \frac{3}{16} + 2^{12}(2^{n+63} \times \frac{3}{16} + 2^{16}(2^{n+51} \times \frac{3}{16} + 2^{12}(2^{n+47} \times \frac{3}{16} + 2^{16}(2^{n+39} \times 1 + 2^{24}(2^{n+15} + 2^{n+11} + 2^{n-1} \times \frac{2}{4})))))/11 + 2^{128}(1 - 2^{-8})^{2^{n-13}}$$

$$\approx 2^{n+92.59} + 2^{128}(1 - 2^{-8})^{2^{n-13}}.$$

The time complexity depends on the choice of n. We set n = 24.0, the data complexity is 2^{60} chosen plaintexts. The total time complexity is about $2^{116.59}$ 11-round encryptions and the total memory complexity is about $2^{96.76}/16 = 2^{92.76}$ 64-bit blocks.

Table 8. Time and data complexity of 11-round attack on Midori-64

Step	Time Complexity	Memory Complexity
1	$2^{n+36} \times 11$	$2^{n+63} \times (9+14) \times 2$
2	$2^{n+63} \times 2^4 \times \frac{1}{4} \times \frac{3}{4}$	$2^{12} \times 2^{n+55}(2+(9+14)\times 2)$
3	$2^{12} \times 2^{n+55} \times 2^8 \times \frac{1}{4} \times \frac{3}{4}$	$2^{16} \times 2^{n+47} (6 + (9 + 14) \times 2)$
4	$2^{12} \times 2^{16} \times 2^{n+47} \times 2^4 \times \frac{1}{4} \times \frac{3}{4}$	$2^{12} \times 2^{n+39}(8 + (9 + 14) \times 2)$
5	$2^{12} \times 2^{16} \times 2^{12} \times 2^{n+39} \times 2^{8} \times \frac{1}{4} \times \frac{3}{4}$	$2^{16} \times 2^{n+31} (12 + (9 + 14) \times 2)$
6	$2^{12} \times 2^{16} \times 2^{12} \times 2^{16} \times 2^{n+31} \times 2^8 \times 1$	$2^{24} \times 2^{n+15} (4 + (9 + 14) \times 2)$
7	$2^{12} \times 2^{16} \times 2^{12} \times 2^{16} \times 2^{24} \times 2^{n+15} \times 1$	_
8	$2^{12} \times 2^{16} \times 2^{12} \times 2^{16} \times 2^{24} \times 2^{n+11} \times 1$	_
9	$2^{12} \times 2^{16} \times 2^{12} \times 2^{16} \times 2^{24} \times 2^{n-13} \times 2^{12} \times \frac{2}{4}$	—
10	$2^{92} \times (1 - 2^{-8})^{2^{n-13}} \times 2^{36} \times 11$	_

6 Conclusion

Previous techniques for searching IDs can generally be divided into two classes. The first one characterizes the propagation of difference patterns, while the second one characterizes the propagation of differential characteristics which may make the model too large to be solved. Besides, due to the huge search space, both techniques cannot traverse the input and output differences.

In this paper, we proposed a new modeling technique with two-dimensional binary variables to search for IDs, which can be seen as a trade-off between two previous techniques. The advantages of our new technique are:

- 1. Other than only considering unknown and inactive difference patterns, we can distinguish between active, inactive, and unknown using two-dimensional binary variables. Moreover, we can consider three distinct nonzero differences at the input and output of the ID.
- 2. Benefiting from the feature of using 2-bit variables, the contradictions can be characterized by constraints. Thus, we can detect any contradictions between the input and output differences by changing the position of contradictions, which releases us from the exhaustive search for input and output differences.
- 3. Since the contradictions can be captured by constraints, this enables us to model the extended rounds within the same MILP model, which makes it possible to search for the best ID with respect to the number of key nibbles to be guessed.
- 4. Our method can quickly obtain the number of IDs. For a fixed position of contradiction, we can derive an ID from each feasible solution of the MILP model. So, we can obtain all IDs by traversing all possible positions of contradiction.

The number of rounds and the complexity are two important factors for an ID. The ID with the longest round can be obtained by searching all valid IDs, but this method is limited by the huge search space. Sun et al. solved this problem by transforming the exhaustive search into an inherent feature of the searching model [16]. However, our model (i.e., two different types of variables to describe the state differences in the single-key scenario) made it easier to deal with this problem as we could directly model both linear and nonlinear layers rather than characterize the propagation of difference patterns through three basic operations (branch, XOR, and S-box). With our new technique, we successfully obtained 6-, 13-, and 12-round IDs for Midori-64, CRAFT, and SKINNY-64 within a few seconds, respectively, which is faster than Hu et al.'s method [9].

Searching for an ID with the best time complexity is a long-term problem. Compared with the method of Sun et al. [16], our technique could search for IDs with optimal time complexity for a particular cipher. Specifically, we combined both the ID-search and the key-recovery by modeling extended rounds into the same MILP model and setting the objective function as the minimum number of keys that need to be guessed. Using the new ID of 6-round Midori-64 we obtained, we presented the current best IDA on 11-round Midori-64 in terms of time complexity.

Acknowledgements. We would like to thank the anonymous reviewers for their helpful comments. This work was supported by Wuhan Science and Technology Bureau (NO. 2022010801020328), and the National Natural Science Foundation of China (NO. 61802119).

A The Example IDs of 13-round CRAFT and 12-round SKINNY-64



Fig. 5. Impossible differential of 13-round CRAFT



Fig. 6. Impossible differential of 12-round SKINNY-64

B The proof of Theorem 1

Proof. We denote the key nibbles guessing way of Theorem 1 as Strategy-1 (S1) and assume that the number of keys to be guessed is r_1 . Without loss of generality, we assume that there is a different key nibbles guessing way (denoted as Strategy-2 (S2)) that only converts the equivalent key nibbles RK'_8 in the S1 into original key nibbles RK_8 , i.e., the equivalent key nibbles RK'_9 , original key nibbles WK, RK_0 , and RK_8 need to be guessed in the S2. Let the GKM of S2 be \mathbb{K}' , the number of keys to be guessed is r_2 . In the following, we prove that

 $r_1 \leq r_2$.

Since some key nibbles of RK'_8 and RK_8 can be calculated according to the linear relations between WK, K'_1 , K_0 , and K'_0 , thus they do not need to be guessed. We denote the key nibbles of RK'_8 in \mathbb{K} and RK_8 in \mathbb{K}' that can be calculated as Calculable Key Nibbles (CKN), and we denote the number of

CKNs in S1 and S2 as $|CKN_1|$ and $|CKN_2|$, respectively. Thus, the linear relations between WK, K'_1 , K_0 , and K'_0 should be considered to calculate r_1 and r_2 . In the following, we denote

$$\mathcal{C}^{i} = \{(i+1) \bmod 4, (i+2) \bmod 4, (i+3) \bmod 4\},\$$

where $0 \leq i \leq 3$. Then

$$K_0'[i] = \bigoplus_{j \in \mathcal{C}^i} K_0[j].$$
⁽²⁾

$$K_0[i] = \bigoplus_{j \in \mathcal{C}^i} K'_0[j].$$
(3)

We take the key nibbles in the first column of $RK'_8(K'_0[0], \ldots, K'_0[3])$ and $RK_8(K_0[0], \ldots, K_0[3])$ as an example to discuss the calculation process of r_1 and r_2 .

Before considering the CKN of RK'_8 and RK_8 :

- 1. If there are 0 nibbles of the first column of RK'_8 need to be guessed in **S1**, there are 0 nibbles in the first column of RK_8 need to be guessed in **S2**.
- 2. If there is 1 nibble of the first column of RK'_8 that needs to be guessed in **S1**, without loss of generality, we assume that $K'_0[0]$, which satisfies Equation 2, needs to be guessed in **S1**. Then there are 3 nibbles in the first column of RK_8 that need to be guessed in **S2**.
- 3. If there are at least 2 nibbles of the first column of RK'_8 that need to be guessed in **S1**, without loss of generality, we assume that $K'_0[0], \ldots, K'_0[n-1]$ $(2 \le n \le 4)$, which satisfy Equation 2, need to be guessed in **S1**. Then there are 4 nibbles in the first column of RK_8 need to be guessed in **S2**.

After considering the CKN of RK'_8 and RK_8 :

- 1. If there are 4 nibbles in the first column of RK_8 that need to be guessed in **S2**, and $|\mathsf{CKN}_2| = m \ (m \leq 4)$, without loss of generality, we assume that $K_0[0], \ldots, K_0[m-1]$, which satisfy Equation 3, are CKNs , and after considering the linear relations between WK, K'_1 , K_0 , and K'_0 , $K_0[m], \ldots, K_0[3]$ are keys that still need to be guessed in **S2**. Then, we need to guess at most 4-mnibbles in the first column of RK'_8 in **S1**, since if $K_0[i] \ (0 \leq i \leq 3)$ is CKN , we only need to guess at most any 2 nibbles in $\{K'_0[j] \mid j \in C^i\}$. For example, when m = 3, without loss of generality, we assume that $K_0[0], K_0[1], K_0[2]$, which satisfy Equation 3, are CKNs , that is, $\bigoplus_{j \in C^0} K'_0[j], \bigoplus_{j \in C^1} K'_0[j]$, and $\bigoplus_{j \in C^2} K'_0[j]$ are known, so we only need to guess at most 4 - m = 1 nibble in $\{K'_0[j] \mid j = 0, 1, 2, 3\}$ in **S1**.
- 2. If there are n $(1 \le n \le 3)$ nibbles in the first column of RK_8 need to be guessed in **S2**, and $|CKN_2| = m$ $(m \le n)$, without loss of generality, we assume that $K_0[0], \ldots, K_0[m-1]$, which satisfy Equation 3, are CKNs, and after considering the linear relations between WK, K'_1 , K_0 , and K'_0 , $K_0[m], \ldots, K_0[n-1]$ are keys that still need to be guessed in **S2**. Then,

we need to guess at most 1 nibble in the first column of RK'_8 in **S1**. In particular, when n = m = 3, without loss of generality, we assume that $K_0[0], K_0[1], K_0[2]$, which satisfy Equation 3, are CKNs. Then, we need to guess 0 nibbles in the first column of RK'_8 in **S1**, since we can calculate $K'_0[3]$ by $K'_0[3] = \bigoplus_{j \in C^3} K_0[j]$.

3. If there are 0 nibbles in the first column of RK_8 need to be guessed in **S2**, and $|CKN_2| = 0$. Then, we need to guess 0 nibbles in the first column of RK'_8 in **S1**.

Therefore, after considering the CKN of RK'_8 and RK_8 , the number of key nibbles that need to be guessed in the first column of RK'_8 in **S1** must be less than or equal to the number of key nibbles that need to be guessed in the first column of RK_8 in **S2**. Similarly, we can get the same conclusion when considering other columns of RK_8 and RK'_8 . Thus,

 $r_1 \leq r_2$.

References

- Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015), https://doi.org/10.1007/978-3-662-48800-3_17
- Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology -CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016), https://doi.org/10.1007/ 978-3-662-53008-5_5
- Beierle, C., Leander, G., Moradi, A., Rasoolzadeh, S.: CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. IACR Trans. Symmetric Cryptol. 2019(1), 5–45 (2019), https://doi.org/10.13154/tosc.v2019. i1.5-45
- Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) Advances in Cryptology EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 12–23. Springer (1999), https://doi.org/10.1007/3-540-48910-X_2
- Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1990), https://doi.org/10.1007/3-540-38424-3_1

- 30 F. Author et al.
- Biryukov, A.: Miss-in-the-middle attack. In: van Tilborg, H.C.A. (ed.) Encyclopedia of Cryptography and Security. Springer (2005), https://doi.org/10.1007/ 0-387-23483-7_256
- Chen, Z., Wang, X.Y.: Impossible differential cryptanalysis of Midori. IACR Cryptol. ePrint Arch. p. 535 (2016), http://eprint.iacr.org/2016/535, withdrawn.
- Cui, T., Jia, K., Fu, K., Chen, S., Wang, M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptol. ePrint Arch. p. 689 (2016), http://eprint.iacr.org/2016/689
- Hu, K., Peyrin, T., Wang, M.: Finding all impossible differentials when considering the DDT. Cryptology ePrint Archive, Paper 2022/1034 (2022), https://eprint. iacr.org/2022/1034
- Kim, J., Hong, S., Sung, J., Lee, C., Lee, S.: Impossible differential cryptanalysis for block cipher structures. In: Johansson, T., Maitra, S. (eds.) Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2904, pp. 82–96. Springer (2003), https://doi.org/10.1007/ 978-3-540-24582-7_6
- 11. Knudsen, L.: DEAL-a 128-bit block cipher. complexity **258**(2), 216 (1998)
- Li, M., Guo, J., Cui, J., et al.: Truncated impossible differential cryptanalysis of Midori-64. (in Chinese). Journal of Software 30(8), 2337–2348 (2019)
- Luo, Y., Lai, X., Wu, Z., Gong, G.: A unified method for finding impossible differentials of block cipher structures. Inf. Sci. 263, 211-220 (2014), https: //doi.org/10.1016/j.ins.2013.08.051
- 14. Sasaki, Y., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: Coron, J., Nielsen, J.B. (eds.) Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10212, pp. 185–215 (2017), https: //doi.org/10.1007/978-3-319-56617-7_7
- Shahmirzadi, A.R., Azimi, S.A., Salmasizadeh, M., Mohajeri, J., Aref, M.R.: Impossible differential cryptanalysis of reduced-round Midori64 block cipher. In: 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology, ISCISC 2017, Shiraz, Iran, September 6-7, 2017. pp. 99–104. IEEE (2017), https://doi.org/10.1109/ISCISC.2017.8488362
- Sun, L., Gérault, D., Wang, W., Wang, M.: On the usage of deterministic (relatedkey) truncated differentials and multidimensional linear approximations for SPN ciphers. IACR Trans. Symmetric Cryptol. 2020(3), 262–287 (2020), https://doi. org/10.13154/tosc.v2020.i3.262-287
- Wu, S., Wang, M.: Automatic search of truncated impossible differentials for wordoriented block ciphers. In: Galbraith, S.D., Nandi, M. (eds.) Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7668, pp. 283–302. Springer (2012), https://doi.org/10.1007/ 978-3-642-34931-7_17