Supplementary Material

Supplementary material for the paper titled 'A Novel Method for Finding Differential-Linear Distinguishers: Application to Midori64, CRAFT, and Skinny64'

Mei Yan[1], Siwei Chen[1(✉)], Zejun Xiang[1], Shasha Zhang[1], and Xiangyong Zeng[2]

[1] School of Cyber Science and Technology, Key Laboratory of Intelligent Sensing System and Security, Hubei University, Wuhan, China
yanmei@stu.hubu.edu.cn, chensiwei_hubu@163.com,
xiangzejun@hubu.edu.cn, amushasha@163.com

[2] Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan, China
xzeng@hubu.edu.cn

A Formation of Linear Constraints

The differential pattern and difference of a word can be described by $(a_0, a_1, b_0, \cdots, b_3)$ and constrained as

$$\begin{cases} -a_0 - b_1 \geq -1 \\ a_1 - b_3 \geq 0 \\ a_1 - b_2 \geq 0 \\ a_1 - b_1 \geq 0 \\ a_1 - b_0 \geq 0 \\ -a_1 + b_0 + b_1 + b_2 + b_3 \geq 0 \\ a_0, a_1, b_0, b_1, b_2, b_3 \text{ are binary variables.} \end{cases} \quad (1)$$

For XOR operation, the propagation can be described by $(a_0^{X_1}, a_0^{X_2}, a_0^{Y}, b_i^{X_1}, b_i^{X_2}, b_i^{Y})$ for each $i \in \{0, 1, 2, 3\}$ and linearly constrained as

$$\begin{cases} -a_0^{X_1} - a_0^{X_2} - b_i^{X_1} - b_i^{X_2} - b_i^{Y} \geq -2 \\ a_0^{X_1} + a_0^{X_2} - a_0^{Y} \geq 0 \\ -a_0^{X_1} + a_0^{Y} - b_i^{X_1} + b_i^{X_1} + b_i^{Y} \geq 0 \\ b_i^{X_1} + b_i^{X_2} - b_i^{Y} \geq 0 \\ -a_0^{X_2} + a_0^{Y} + b_i^{X_1} - b_i^{X_2} + b_i^{Y} \geq 0 \\ a_0^{X_1}, a_0^{X_2}, a_0^{Y}, b_i^{X_1}, b_i^{X_2}, b_i^{Y} \text{ are binary variables.} \end{cases} \quad (2)$$

For a single S-box of Midori64, the propagation can be described by $(a_0^X, a_1^X, a_0^Y, a_1^Y)$ and $(b_0^X, \cdots, b_3^X, b_0^Y, \cdots, b_3^Y, p_0, p_1)$, and constrained simultaneously as the following two systems of linear inequalities.

$$\begin{cases} -a_0^Y - a_1^Y \geq -1 \\ -a_0^X + a_0^Y \geq 0 \\ -a_1^X + a_0^Y + a_1^Y \geq 0 \\ a_0^X + a_1^X - a_0^Y \geq 0 \\ a_1^X - a_1^Y \geq 0 \\ -a_0^X - a_1^X \geq -1 \\ a_0^X, a_1^X, b_0^Y, b_1^Y \text{ are binary variables.} \end{cases} \tag{3}$$

$$\begin{cases} b_2^X - b_1^Y - b_2^Y - b_3^Y - p_0 + 3p_1 \geq 0 \\ -2b_0^X - b_1^X - b_3^X + 2b_1^Y + 6b_2^Y + 2b_3^Y + 5p_0 - 6p_1 \geq -4 \\ b_1^X - 2b_2^X + b_3^X + b_1^Y - 2b_2^Y + b_3^Y - 3p_0 + 2p_1 \geq -2 \\ 4b_0^X + 3b_1^X + 6b_2^X - b_3^X - 8b_0^Y - 2b_1^Y - 2b_2^Y - 6b_3^Y + 5p_0 + 7p_1 \geq -1 \\ +4b_0^X + 5b_1^X + 3b_2^X + 5b_3^X + 3b_0^Y - 4b_1^Y - 3b_2^Y - 4b_3^Y + 2p_0 + p_1 \geq 0 \\ b_1^X - 2b_2^X + b_3^X + 3b_0^Y + 3b_1^Y - b_2^Y + 3b_3^Y - 2p_0 - 2p_1 \geq -2 \\ 8b_0^X - b_1^X - 2b_2^X + 2b_3^X - 5b_0^Y - 4b_1^Y - 7b_2^Y - b_3^Y + 9p_0 + 8p_1 \geq -3 \\ -7b_0^X + 4b_1^X + b_2^X - 2b_3^X + b_0^Y - 8b_1^Y - b_2^Y + 4b6Y_3 + 6p_0 + p_1 \geq -9 \\ -3b_0^X - 4b_1^X + b_2^X - 2b_3^X + b_0^Y + 6b_1^Y + 4b_2^Y - 5b_3^Y + 7p_0 - 5p_1 \geq -9 \\ -b_1^X - b_2^X - b_3^X + b_0^Y - b_1^Y + b_2^Y - b_3^Y - 2p_0 + 2p_1 \geq -3 \\ -b_0^X - 3b_1^X - 2b_2^X + 3b_3^X - 4b_0^Y + 3b_1^Y - b_2^Y - 2b_3^Y + 2p_0 + 2p_1 \geq -6 \\ b_1^X - 2b_2^X - b_3^X + 2b_0^Y - b_1^Y - 2b_2^Y - 3b_3^Y + 3p_0 + 3p_1 \geq -3 \\ 2b_0^X + b_1^X + 3b_2^X + b_3^X - 3b_0^Y - 2_1^Y + b_2^Y - 2b_3^Y + p_0 + 2p_1 \geq 0 \\ b_0^X - b_1^X + b_2^X - b_3^X + b_0^Y + 2b_1^Y - b_2^Y + 2b_3^Y - 2p_0 \geq -2 \\ b_1^X - b_2^X + b_3^X - p_0 \geq -1 \\ -b_0^X - b_1^X - 2b_2^X - 3b_3^X - 2b_0^Y - 2b_1^Y + 3b_2^Y + 4b_3^Y + 2p_0 - p_1 \geq -7 \\ -4b_0^X - 5b_1^X - 2b_2^X + b_3^X + 6b_0^Y - 2b_1^Y + 5b_2^Y + 4b_3^Y + 3p_0 - 7p_1 \geq -11 \\ 3b_0^X - 2b_1^X + 2b_2^X + b_3^X - b_0^Y - 2b_1^Y - 2b_2^Y + b_3^Y + 2p_1 \geq -2 \\ b_0^X + 2b_1^X + 3b_2^X + 2b_3^X - 2b_0^Y + b_1^Y + b_3^Y + 2p_0 - 2p_1 \geq 0 \\ -b_1^X - 2b_2^X - 2b_3^X + b_0^Y + 3b_1^Y + 2b_2^Y + p_0 - 3p_1 \geq -5 \\ -b_1^X - b_3^X - b_1^Y - b_3^Y - p_0 \geq -4 \\ -b_0^X + b_1^X - b_3^X - b_0^Y - b_2^Y + b_3^Y \geq -3 \\ -2b_0^X - 2b_1^X - b_2^X + 3b_3^X - b_0^Y + 3b_1^Y - 2b_2^Y - b_3^Y + p_1 \geq -5 \\ -b_0^X - b_3^X + b_0^Y + b_1^Y + b_2^Y - p_1 \geq -2 \\ b_0^X + b_2^X - b_3^X + b_1^Y - b_2^Y - p_0 \geq -2 \\ b_i^X, b_i^Y, p_0, p_1 \text{ are binary variables.} \end{cases} \tag{4}$$

B The descriptions of Midori64, CRAFT and Skinny64

B.1 Midori64

The state of Midori64 is a 4×4 matrix, arranged as follows:

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

where s_i represents the cell with a bit size of 4.

For the round function $F = \texttt{AK} \circ \texttt{MC} \circ \texttt{SC} \circ \texttt{SB}$. The details about four components of F are clarified as below:

- **SubCell** (SB): Each cell in the state goes through the same S-box, the truth table of this S-box is listed as Table 1.

Table 1. The truth table of S-box used in Midori64 and CRAFT.

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

- **ShuffleCell** (SC): The cells in the state are permuted as follows:

$$\mathcal{P} = [0, 10, 5, 15, 14, 4, 11, 1, 9, 3, 12, 6, 7, 13, 2, 8].$$

- **MixColumn** (MC): The state matrix is multiplied by a binary almost MDS matrix M, as shown below:

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

- **AddRoundKey** (AK): The rth round-key is XOR-ed with the state.

At the last round, the state only goes through the S-box and is XOR-ed with the last round-key.

B.2 CRAFT

The 64-bit internal state of CRAFT is viewed as a 4×4 square array of 16 4-bit cells as follows:

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

where s_i denote the ith cell.

The round function of CRAFT is $F = \texttt{SB} \circ \texttt{PN} \circ \texttt{ATK}_i \circ \texttt{ARC}_i \circ \texttt{MC}$, where $i \in \{0, 1, \cdots, 30\}$. The five transformations involved are explained below.

- **MixColumn** (MC): Each column of the state is multiplied by the following binary matrix:

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

- **AddConstants**$_i$ (ARC$_i$): The state cells s_4 and s_5 are XOR-ed with two 4-bit constants a_i and b_i in the ith round, respectively.
- **AddTweakey**$_i$ (ATK$_i$): The state is XOR-ed with the round tweakey.
- **PermuteNibbles** (PN): An involutory permutation $\mathcal{P}$ is applied to the cell positions in the state.

$$\mathcal{P} = [15, 12, 13, 14, 10, 9, 8, 11, 6, 5, 4, 7, 1, 2, 3, 0].$$

- **SubBox** (SB): Each cell of the state goes through the same S-box. The truth table of the S-box is given in Table 1.

Additional, the last round excludes the SB operation.

B.3 Skinny64

Similar to CRAFT, the internal state of Skinny64 is splited to 16 cells with a bit size of 4, and arranged as a square array as

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

where s_i denote the ith cell.

The components of the round function of Skinny64 are explained briefly as follows.

- **SubCell** (SC): Each cell in the state goes through the same S-box. The truth table of Skinny64's S-box is listed as the following table.

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

- **AddConstants** (AC): The state s_0 and s_4 are XOR-ed with two constants c_0 and c_1, respectively.
- **AddRoundTweakey** (ART): The first two rows of the state array are XOR-ed with the round tweakey.
- **ShiftRows** (SR): The ith row of the state array is right rotated by i positions for $i \in \{0, 1, 2, 3\}$.
- **MixColumn** (MC): Each column of the state array is multiplied by the following binary matrix M:

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

C The DL Distinguishers of Midori64, CRAFT, and Skinny64

In this section, we will illustrate the DL distinguishers of Midori64, CRAFT and Skinny64, where the notation like 'a_x' is the hexadecimal representation of the 4-bit vector $(1, 0, 1, 0)$. The white (resp. black) word indicates its differential pattern is Z (resp. U^*). The orange (resp. green) word means the pattern is N before (resp. after) going through the S-box. Besides, the yellow (resp. gray) word in the last matrix represents the non-zero (resp. zero) output mask.

C.1 The DL distinguishers of Midori64

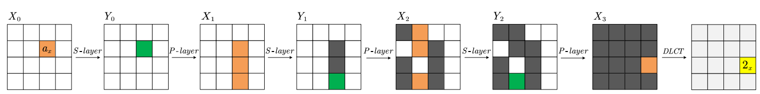

Fig. 1. The 4-round DL distinguisher of Midori64.

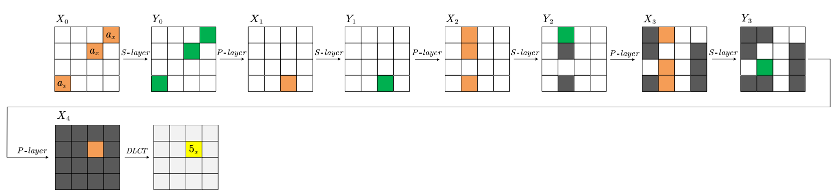

Fig. 2. The 5-round DL distinguisher of Midori64.

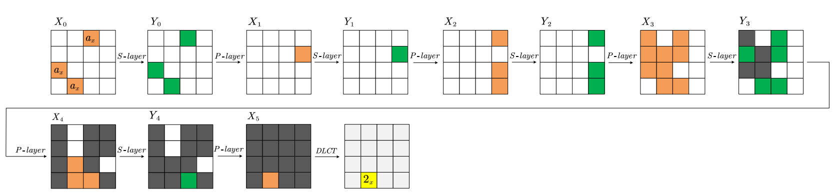

Fig. 3. The 6-round DL distinguisher of Midori64.

C.2 The DL distinguishers of CRAFT

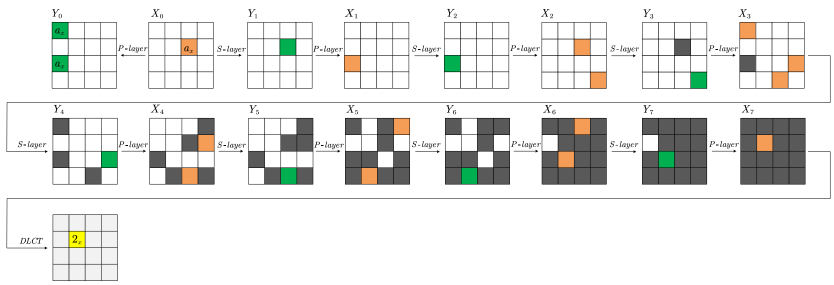

Fig. 4. The 8-round DL distinguisher of CRAFT.

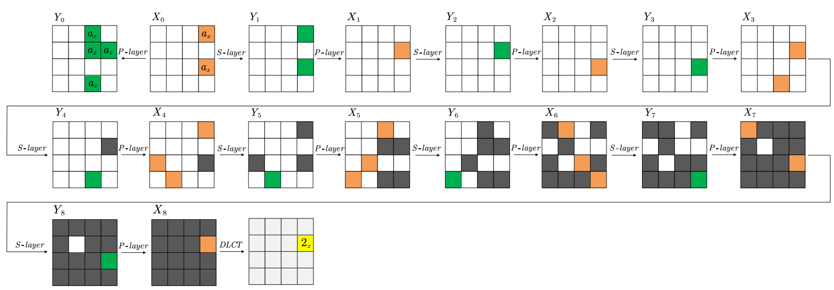

Fig. 5. The 9-round DL distinguisher of CRAFT.

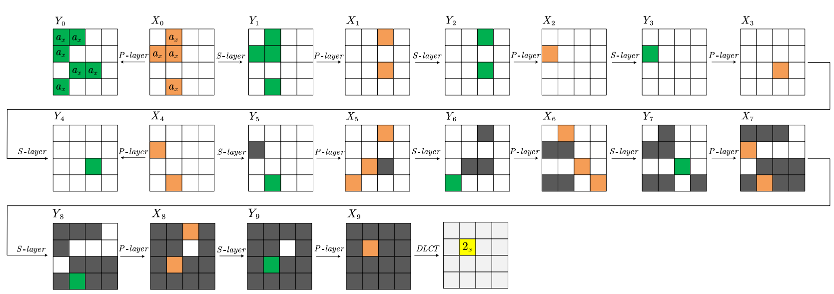

Fig. 6. The 10-round DL distinguisher of CRAFT.

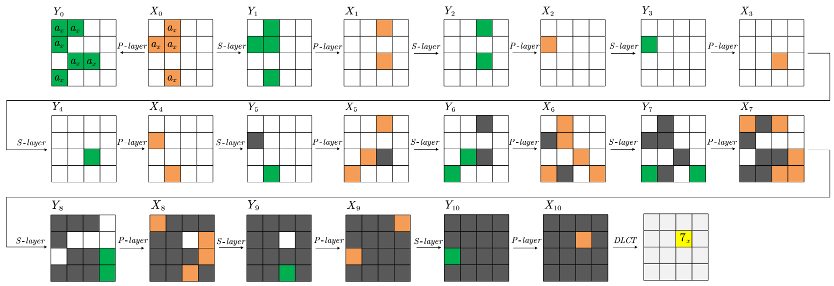

Fig. 7. The 11-round DL distinguisher of CRAFT.

C.3 The DL distinguishers of **Skinny64**

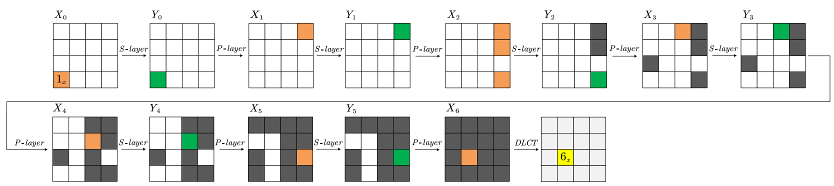

Fig. 8. The 7-round DL distinguisher of Skinny64.

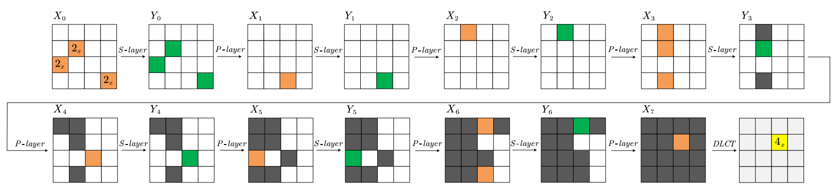

Fig. 9. The 8-round DL distinguisher of Skinny64.

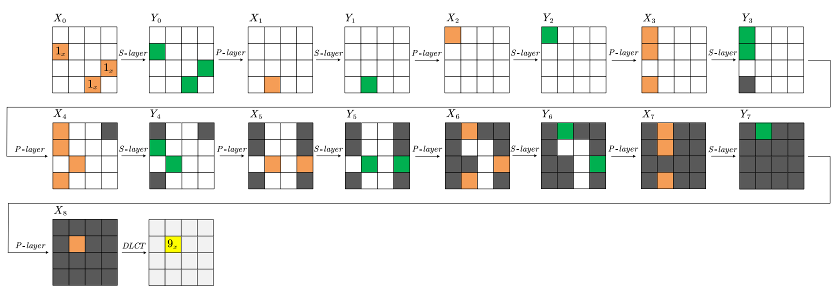

Fig. 10. The 9-round DL distinguisher of Skinny64.

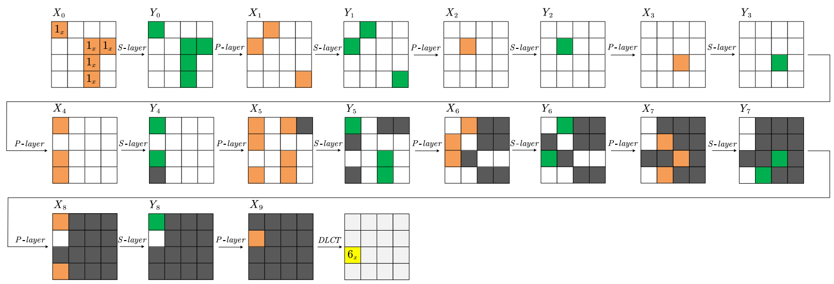

Fig. 11. The 10-round DL distinguisher of Skinny64.