

# Supplementary Material

## Supplementary material for the paper titled ‘On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers’

Ling Sun<sup>1,2</sup>, David Gerault<sup>3</sup>, Wei Wang<sup>1,2</sup> and Meiqin Wang<sup>1,2</sup> (✉)

<sup>1</sup> Key Laboratory of Cryptologic Technology and Information Security,  
Ministry of Education, Shandong University, Jinan, China

<sup>2</sup> School of Cyber Science and Technology, Shandong University, Qingdao, China

<sup>3</sup> Nanyang Technological University, Singapore

lingsun@sdu.edu.cn, david.gerault@ntu.edu.sg, {weiwangsdu, mqwang}@sdu.edu.cn

## A Searching for Deterministic TDs and MDLAs

### A.1 The Construction of CSP models for Basic Operations

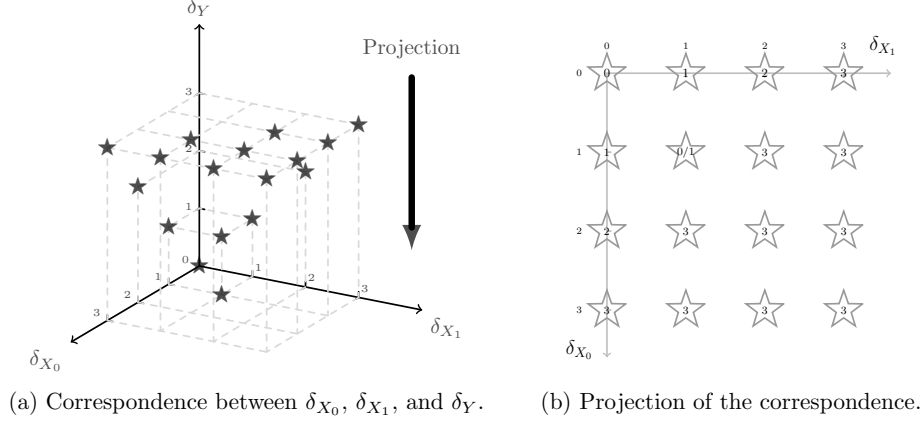
**Model (XOR).** For the XOR operation with two inputs  $X_0, X_1$  and one output  $Y$ , the following constraint specifies the propagation of the differential pattern.

```
if  $\delta_{X_0} + \delta_{X_1} > 2$  then  $\delta_Y = 3$  and  $\zeta_Y = -2$ 
elseif  $\delta_{X_0} + \delta_{X_1} = 1$  then  $\delta_Y = 1$  and  $\zeta_Y = \zeta_{X_0} + \zeta_{X_1}$ 
elseif  $\delta_{X_0} = \delta_{X_1} = 0$  then  $\delta_Y = 0$  and  $\zeta_Y = 0$ 
elseif  $\zeta_{X_0} + \zeta_{X_1} < 0$  then  $\delta_Y = 2$  and  $\zeta_Y = -1$ 
elseif  $\zeta_{X_0} = \zeta_{X_1}$  then  $\delta_Y = 0$  and  $\zeta_Y = 0$ 
else  $\delta_Y = 1$  and  $\zeta_Y = \zeta_{X_0} \oplus \zeta_{X_1}$  endif
```

*Proof.* We take the inclusion method and try to describe the valid values of the 6-tuple  $\langle \delta_{X_0}, \zeta_{X_0}, \delta_{X_1}, \zeta_{X_1}, \delta_Y, \zeta_Y \rangle$ . Since the value of  $\zeta_*$  is useless in most cases, we first focus on appropriate choices for the 3-tuple  $\langle \delta_{X_0}, \delta_{X_1}, \delta_Y \rangle$ . Lemma 2 reveals all possible combinations of  $\langle \delta_{X_0}, \delta_{X_1}, \delta_Y \rangle$ , and we insert these values into the 3-dimensional vector space, which are marked with black stars in Figure 1(a). Then, as illustrated in Figure 1(b), we project all the valid points  $\langle \delta_{X_0}, \delta_{X_1}, \delta_Y \rangle$  onto the plane  $\delta_Y = 0$ , and the value circled within the star is the actual value of  $\delta_Y$ . In the following, the conditional expression is employed to set the value of  $\delta_Y$  regarding  $\delta_{X_0}$  and  $\delta_{X_1}$  case by case first and then determine the value of  $\zeta_Y$ , accordingly.

As in Figure 1(b), we observe that  $\delta_Y$  equals to 3 if  $\delta_{X_0} + \delta_{X_1}$  is no less than 3, and  $\zeta_Y = -2$  holds for sure in this situation. In this way, we get the first conditional statement of the constraint. The following restriction  $\delta_{X_0} + \delta_{X_1} = 1$  tells that one input branch has the zero difference and the other one has a nonzero fixed difference. So, the output difference  $\Delta Y$  is fixed and equals to the difference of the nonzero input branch, i.e.,  $\delta_Y = 1$  and  $\zeta_Y = \zeta_{X_0} + \zeta_{X_1}$ , which constitutes the second conditional statement. Following that, the statement  $\delta_{X_0} = \delta_{X_1} = 0$  is easy to understand as it relates to the case where the two input branches have zero difference, simultaneously. Among the remaining three possible values of  $\langle \delta_{X_0}, \delta_{X_1} \rangle$ , only (0, 2) and (2, 0) fill the condition  $\zeta_{X_0} + \zeta_{X_1} < 0$ . For these two cases, the output pattern must be  $N^*$ , and we have  $\delta_Y = 2$  and  $\zeta_Y = -1$ . The last two

statements deal with the situation where both of the two input branches have nonzero fixed differences. In this setting, the output difference  $\Delta Y$  equals to  $\zeta_{X_0} \oplus \zeta_{X_1}$ , and the value of  $\delta_Y$  varies with the value of  $\zeta_{X_0} \oplus \zeta_{X_1}$ . The equation  $\zeta_{X_0} = \zeta_{X_1}$  indicates  $\delta_Y = 0$  and  $\zeta_Y = 0$ . Otherwise, the relation should be  $\delta_Y = 1$  and  $\zeta_Y = \zeta_{X_0} \oplus \zeta_{X_1}$ .  $\square$

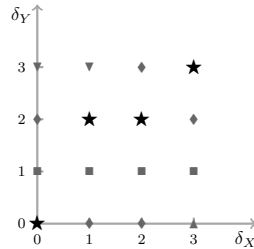


**Figure 1:** Constructing the model for XOR operation.

**Model (S-box).** For the S-box with input  $X$  and output  $Y$ , the following constraint clarifies all possible pattern propagations.

$$\delta_Y \neq 1 \text{ and } \delta_X + \delta_Y \in \{0, 3, 4, 6\} \text{ and } \delta_Y \geq \delta_X \text{ and } \delta_Y - \delta_X \leq 1$$

*Proof.* Note that the valid set of  $\langle \delta_X, \delta_Y \rangle$  is  $\{(0, 0), (1, 2), (2, 2), (3, 3)\}$ . We employ the exclusion method and remove all impossible points from the domain  $\mathcal{D}(\delta_X) \times \mathcal{D}(\delta_Y)$ . As shown in Figure 2, all the candidates for the pair  $\langle \delta_X, \delta_Y \rangle$  are pointed out with black stars. Firstly, note that the condition  $\delta_Y \neq 1$  enables us to reject the points marked with black squares. Then, we narrow the value of  $\delta_X + \delta_Y$  to the set  $\{0, 3, 4, 6\}$ , which further removes the points marked with black lozenges. The third constraint  $\delta_Y \geq \delta_X$  eliminates the point covered by the black triangle. At last, we add the inequality  $\delta_Y - \delta_X \leq 1$ , which excludes the remaining impossible cases labelled with inverted triangles. Hence, impossible propagations violate at least one of the four formulas, while right propagations validate all the four formulas, simultaneously.  $\square$



**Figure 2:** Constructing the model for S-box operation.

## A.2 The Reason to Use CP

We do not claim that the CP is the unique method which can accomplish the search of (RK) TDs and MDLAs. The first reason we employ the CP lies in its conciseness.

Since it supports high-level descriptions, the construction of the model is comparatively transparent. To demonstrate this advantage, we try to reconstruct Model 1 with MILP and SAT methods, respectively.

We take a 4-bit word  $X$  as an example and use two Boolean variables  $\delta_0$  and  $\delta_1$  to represent the differential pattern  $\delta_X$ , six Boolean variables  $\zeta_0, \zeta_1, \dots, \zeta_5$  to replace  $\zeta_X$ . The valid values of variables in different settings are summarised in the following table.

CSP	SAT/MILP	CSP	SAT/MILP
$\delta_X = 0$	$\delta_0 \parallel \delta_1 = 00$	$\zeta_X = 0$	$\zeta_0 \parallel \zeta_1 \parallel \dots \parallel \zeta_5 = 000000$
$\delta_X = 1$	$\delta_0 \parallel \delta_1 = 01$	$\zeta_X > 0$	$\zeta_0 \parallel \zeta_1 \parallel \dots \parallel \zeta_5 = 00****$ $\zeta_0 \parallel \zeta_1 \parallel \dots \parallel \zeta_5 \neq 000000$
$\delta_X = 2$	$\delta_0 \parallel \delta_1 = 10$	$\zeta_X = -1$	$\zeta_0 \parallel \zeta_1 \parallel \dots \parallel \zeta_5 = 010000$
$\delta_X = 3$	$\delta_0 \parallel \delta_1 = 11$	$\zeta_X = -2$	$\zeta_0 \parallel \zeta_1 \parallel \dots \parallel \zeta_5 = 100000$

With the correspondence between  $\delta_X$  and  $\zeta_X$ , we derive the valid set  $\mathcal{V} \triangleq \mathcal{V}(\langle \delta_0, \delta_1, \zeta_0, \dots, \zeta_5 \rangle)$  of the 8-tuple  $\langle \delta_0, \delta_1, \zeta_0, \dots, \zeta_5 \rangle$ , that is,

$$\mathcal{V} = \left\{ \begin{array}{l} 0x00, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, \\ 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f, 0x90, 0xe0 \end{array} \right\}.$$

### A.2.1 Two Methods to Generate MILP Models

**Logical condition modelling [SHW<sup>+</sup>14a, SHW<sup>+</sup>14b].** The main idea of this method is to remove improper values one by one. For a specific invalid evaluation, an inequality about all variables is created, which will be disrupted only when the involved variables are instantiated with the invalid evaluation. For instance, in our setting, regarding the impossible point ‘00100110’, a linear combination  $\delta_0 + \delta_1 - \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 - \zeta_4 + \zeta_5$  is created firstly, where the coefficients of the variables whose values equal to 1 are set as ‘-1’, and the coefficients of the remaining variables are set as ‘1’. Note that

$$\min \left\{ \delta_0 + \delta_1 - \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 - \zeta_4 + \zeta_5 \mid \delta_i \in \mathbb{F}_2, \zeta_j \in \mathbb{F}_2 \right\} = -3,$$

and the linear formula gains the minimum value only when  $\delta_0 \parallel \delta_1 \parallel \zeta_0 \parallel \dots \parallel \zeta_5 = 00100110$ . Thus, adding the following inequality to the MILP model eliminates this invalid point

$$\delta_0 + \delta_1 - \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 - \zeta_4 + \zeta_5 \geq -2.$$

Since we have  $|\mathbb{F}_2^8 \setminus \mathcal{V}| = 238$  impossible combinations, in theory, 238 inequalities are expected to establish the model. Simplification for the inequality system is necessary since the redundant formulae decelerate the solving phase of the MILP problem. The greedy algorithm [SHW<sup>+</sup>14b], which intends to derive the “best” inequalities that maximise the number of removed impossible points, does not work in this setting since each formula exactly discards one impossible case. Thus, we take the method called the simplification of the product-of-sum representation in [AST<sup>+</sup>17]. Firstly, we define an 8-bit Boolean function

$$f(\delta_0, \delta_1, \zeta_0, \dots, \zeta_5) = \begin{cases} 0, & \text{if } \delta_0 \parallel \delta_1 \parallel \zeta_0 \parallel \dots \parallel \zeta_5 \in \mathcal{V} \\ 1, & \text{otherwise} \end{cases},$$

and then generate its product-of-sum representation

$$f(\delta_0, \delta_1, \zeta_0, \dots, \zeta_5) = \bigwedge_{\mathbf{a} \in \mathbb{F}_2^8} \left( f(\mathbf{a}) \vee \bigvee_{i=0}^1 (\delta_i \oplus a_i) \vee \bigvee_{i=0}^5 (\zeta_i \oplus a_{i+2}) \right).$$

After invoking Logic Friday<sup>1</sup> software to simplify the representation, a smaller inequality system is decoded from the result. At last, we receive 13 inequalities

$$\begin{cases} \delta_0 - \zeta_i \geq 0 \ (0 \leq i \leq 1) \\ \delta_1 - \zeta_j \geq 0 \ (2 \leq j \leq 5) \\ -\delta_0 - \zeta_k + 1 \geq 0 \ (2 \leq k \leq 5) \\ -\zeta_0 - \zeta_1 + 1 \geq 0 \\ -\delta_0 + \delta_1 + \zeta_1 \geq 0 \\ -\delta_1 + \zeta_0 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 \geq 0 \end{cases},$$

which is one MILP version of Model 1.

**H-representation of the convex hull [SHW<sup>+</sup>14b].** We view the set  $\mathcal{V}$  as a discrete set in  $\mathbb{R}^8$ , and the *convex hull*  $\text{con}(\mathcal{V})$  of  $\mathcal{V}$  is referred to as the smallest convex set containing  $\mathcal{V}$ . The *H-representation* of the convex hull is a finite linear (in)equality system, whose solutions have a one-to-one correspondence with the vectors in  $\text{con}(\mathcal{V})$ . For the convex hull in the lower-dimensional space, we can invoke the `inequality_generator()` function in the `sage.geometry.polyhedron` class of SageMath<sup>2</sup> to compute the H-representation, efficiently. In our case, the H-representation of  $\text{con}(\mathcal{V})$  is comprised of 12 inequalities. However, a careful analysis shows that  $\mathcal{V} \neq \text{con}(\mathcal{V})$  since the H-representation of  $\text{con}(\mathcal{V})$  incorporates 36 points in  $\mathbb{F}_2^8$ . To obtain a compact representation of the set  $\mathcal{V}$ , we reapply the logical condition modelling method and generate one inequality for each of the 18 impossible points in  $\text{con}(\mathcal{V}) \setminus \mathcal{V}$ . The composition of these inequalities with the output of SageMath strictly specify the vectors in  $\mathcal{V}$ . With the greedy algorithm once again, we get an inequality system with 24 inequalities

$$\begin{cases} -\delta_1 - \zeta_1 + 1 \geq 0 \\ \delta_1 - \zeta_0 - \zeta_i \geq 0 \ (2 \leq i \leq 5) \\ -\delta_1 + \zeta_0 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 \geq 0 \\ \delta_0 + \delta_1 + \zeta_0 - \zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 \geq 0 \\ \delta_0 - \delta_1 - \zeta_0 + \zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 + 1 \geq 0 \\ -\delta_0 + \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 - \zeta_5 + 2 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 + \zeta_3 - \zeta_4 + \zeta_5 + 2 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 + \zeta_3 - \zeta_4 - \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 + \zeta_4 + \zeta_5 + 2 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 + \zeta_4 - \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 - \zeta_4 + \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 + \zeta_2 - \zeta_3 - \zeta_4 - \zeta_5 + 4 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 + 2 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 + \zeta_3 + \zeta_4 - \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 + \zeta_3 - \zeta_4 + \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 + \zeta_3 - \zeta_4 - \zeta_5 + 4 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 - \zeta_3 + \zeta_4 + \zeta_5 + 3 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 - \zeta_3 + \zeta_4 - \zeta_5 + 4 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 - \zeta_3 - \zeta_4 + \zeta_5 + 4 \geq 0 \\ -\delta_0 - \delta_1 + \zeta_0 + \zeta_1 - \zeta_2 - \zeta_3 - \zeta_4 - \zeta_5 + 5 \geq 0 \end{cases}.$$

<sup>1</sup><http://sontrak.com/>

<sup>2</sup><http://www.sagemath.org/index.html>

This inequality system is another optional MILP version of Model 1.

### A.2.2 Creating the SAT Model

It was pointed out in [SWW18] that the simplification of the product-of-sum representation is also available to construct the SAT model. Consequently, a SAT version of Model 1 is yielded as follows

$$\begin{cases} \delta_0 \vee \bar{\zeta}_i = 1 & (0 \leq i \leq 1) \\ \delta_1 \vee \bar{\zeta}_j = 1 & (2 \leq j \leq 5) \\ \bar{\delta}_0 \vee \bar{\zeta}_k = 1 & (2 \leq k \leq 5) \\ \bar{\zeta}_0 \vee \bar{\zeta}_1 = 1 \\ \bar{\delta}_0 \vee \delta_1 \vee \zeta_1 = 1 \\ \bar{\delta}_1 \vee \zeta_0 \vee \zeta_2 \vee \zeta_3 \vee \zeta_4 \vee \zeta_5 = 1 \end{cases}.$$

In light of the simple example, the construction of the CP model is fairly direct comparing to MILP and SAT methods. Besides, since the CP model is more readable than MILP and SAT models, it is convenient for us to locate mistakes in the model. As in the above example, given the set  $\mathcal{V}$ , if we do not know  $\text{con}(\mathcal{V}) \neq \mathcal{V}$  in advance and wrongfully regard the H-representation of  $\text{con}(\mathcal{V})$  as the MILP model of  $\mathcal{V}$ , many undesirable evaluations will be involved into the solution set, which definitely weakens the power of the searching algorithm.

## A.3 A Generalisation for the Search of TDs and MDLAs

In the generalised version, we take the pattern  $\mathbf{N} \oplus \mathbf{N}^*$  into account. After redefining the set  $\mathcal{X}'$  of variables and their domains  $\mathcal{D}'$ , we construct the new set  $\mathcal{C}'$  of constraints, accordingly. The new CSP  $\mathcal{P}' = \langle \mathcal{X}', \mathcal{D}', \mathcal{C}' \rangle$  can realise the function of propagating  $\mathbf{N} \oplus \mathbf{N}^*$  to a certain extent.

### A.3.1 Initialising Variables

Similarly to the basic method introduced in the main body of the paper, for each entry  $X_i$  of the inner state  $X$ , we introduce an integer variable  $\delta'_{X_i}$  to stand for its differential pattern. The domain  $\mathcal{D}(\delta'_{X_i})$  of  $\delta'_{X_i}$  is  $\{z \in \mathbb{Z} \mid 0 \leq z \leq 4\}$ . The correspondence between the differential pattern of  $X_i$  and  $\delta'_{X_i}$  is

$$\delta'_{X_i} = \begin{cases} 0, & \text{if } \Delta_{X_i} = \mathbf{Z} \\ 1, & \text{if } \Delta_{X_i} = \mathbf{N} \\ 2, & \text{if } \Delta_{X_i} = \mathbf{N}^* \\ 3, & \text{if } \Delta_{X_i} = \mathbf{N} \oplus \mathbf{N}^* \\ 4, & \text{if } \Delta_{X_i} = \mathbf{U} \end{cases}.$$

The variable  $\zeta'_{X_i}$  is also imported for each  $X_i$ , and the data range of  $\zeta'_{X_i}$  is updated as

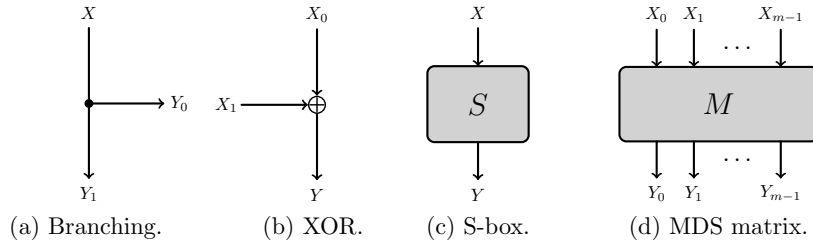
$$\zeta'_{X_i} \in \begin{cases} \{0\}, & \text{if } \delta'_{X_i} = 0 \\ \{1, 2, \dots, 2^s - 1\}, & \text{if } \delta'_{X_i} = 1 \\ \{0\}, & \text{if } \delta'_{X_i} = 2 \\ \{1, 2, \dots, 2^s - 1\}, & \text{if } \delta'_{X_i} = 3 \\ \{-1\}, & \text{if } \delta'_{X_i} = 4 \end{cases}.$$

If  $\Delta X_i$  is known, i.e.,  $\Delta X_i = Z/N$ ,  $\zeta'_{X_i}$  equals the actual value of  $\Delta X_i$ . Moreover, if a partial message of  $\Delta X_i$  is known, that is,  $\Delta X_i = N \oplus N^*$ ,  $\zeta'_{X_i}$  memorises the difference of the nonzero fixed part in  $\Delta X_i$ .

Typically, the XOR of two nonzero varied differential patterns  $N_1^*$  and  $N_2^*$  becomes  $U$ . However, if the equality relation  $N_1^* = N_2^*$  between them is identified in advance, we have  $N_1^* \oplus N_2^* = Z$ , although their actual values are unknown. Based on this observation, the two patterns  $N_1^*$  and  $N_2^*$  are said to belong to the same class if they satisfy the equality relation; otherwise, they come from different classes. Note that the input and output patterns of an S-box belong to different classes since the relation between them is vague. To record the class of the pattern  $N^*$ , we introduce a new variable  $\xi'_{X_i}$  for each  $X_i$ . The data range of  $\xi'_{X_i}$  is related to the value of  $\delta'_{X_i}$

$$\xi'_{X_i} \in \begin{cases} \{0\}, & \text{if } \delta'_{X_i} = 0 \\ \{0\}, & \text{if } \delta'_{X_i} = 1 \\ \{1, 2, \dots\}, & \text{if } \delta'_{X_i} = 2 \\ \{1, 2, \dots\}, & \text{if } \delta'_{X_i} = 3 \\ \{0\}, & \text{if } \delta'_{X_i} = 4 \end{cases}.$$

The value of  $\xi'_{X_i}$  represents the class of the pattern  $N^*$  when  $\delta'_{X_i} = 2/3$ .



**Figure 3:** Basic operations.

The CSP model depicting the link among  $\delta'_{X_i}$ ,  $\zeta'_{X_i}$  and  $\xi'_{X_i}$  is supplied as follows.

**Model** (Relation among  $\delta'_{X_i}$ ,  $\zeta'_{X_i}$  and  $\xi'_{X_i}$ ). *The following constraint will ensure that the values of  $\zeta'_{X_i}$  and  $\xi'_{X_i}$  fall into correct domains.*

*if*  $\delta'_{X_i} = 0$  *then*  $\zeta'_{X_i} = 0$  *and*  $\xi'_{X_i} = 0$   
*elseif*  $\delta'_{X_i} = 1$  *then*  $\zeta'_{X_i} > 0$  *and*  $\xi'_{X_i} = 0$   
*elseif*  $\delta'_{X_i} = 2$  *then*  $\zeta'_{X_i} = 0$  *and*  $\xi'_{X_i} > 0$   
*elseif*  $\delta'_{X_i} = 3$  *then*  $\zeta'_{X_i} > 0$  *and*  $\xi'_{X_i} > 0$   
*else*  $\zeta'_{X_i} = -1$  *and*  $\xi'_{X_i} = 0$  *endif*

### A.3.2 Propagating Differential Patterns

The new CSP models for basic operations are proposed.

**Model** (Branching). *For the Branching operation in Figure 3(a), the following constraint defines the pattern propagation.*

$$\delta'_{Y_0} = \delta'_X \text{ and } \zeta'_{Y_0} = \zeta'_X \text{ and } \xi'_{Y_0} = \xi'_X \text{ and } \delta'_{Y_1} = \delta'_X \text{ and } \zeta'_{Y_1} = \zeta'_X \text{ and } \xi'_{Y_1} = \xi'_X$$

**Model (XOR).** For the XOR operation in Figure 3(b), the following constraint specifies the propagation of the differential pattern.

```

if  $\delta'_{X_0} + \delta'_{X_1} > 4$  then  $\delta'_Y = 4$  and  $\zeta'_Y = -1$  and  $\xi'_Y = 0$ 
elseif  $\delta'_{X_0} = 0$  and  $\delta'_{X_1} = 0$  then  $\delta'_Y = 0$  and  $\zeta'_Y = 0$  and  $\xi'_Y = 0$ 
elseif  $\delta'_{X_0} + \delta'_{X_1} = 1$  then  $\delta'_Y = 1$  and  $\zeta'_Y = \zeta'_{X_0} + \zeta'_{X_1}$  and  $\xi'_Y = 0$ 
elseif  $\delta'_{X_0} + \delta'_{X_1} = 3$  then  $\delta'_Y = 3$  and  $\zeta'_Y = \zeta'_{X_0} + \zeta'_{X_1}$  and  $\xi'_Y = \xi'_{X_0} + \xi'_{X_1}$ 
elseif  $\delta'_{X_0} = 4$  or  $\delta'_{X_1} = 4$  then  $\delta'_Y = 4$  and  $\zeta'_Y = -1$  and  $\xi'_Y = 0$ 
elseif  $\delta'_{X_0} = 2$  and  $\delta'_{X_1} = 2$  then  $\delta'_Y = 4$  and  $\zeta'_Y = -1$  and  $\xi'_Y = 0$ 
elseif  $\delta'_{X_0} + \delta'_{X_1} = 4$  and  $\zeta'_{X_0} = \zeta'_{X_1}$  then
     $\delta'_Y = 2$  and  $\zeta'_Y = 0$  and  $\xi'_Y = \xi'_{X_0} + \xi'_{X_1}$ 
elseif  $\delta'_{X_0} + \delta'_{X_1} = 4$  then  $\delta'_Y = 3$  and  $\zeta'_Y = \zeta'_{X_0} \oplus \zeta'_{X_1}$  and  $\xi'_Y = \xi'_{X_0} + \xi'_{X_1}$ 
elseif  $\delta'_{X_0} = 2$  or  $\delta'_{X_1} = 2$  then  $\delta'_Y = 2$  and  $\zeta'_Y = 0$  and  $\xi'_Y = \xi'_{X_0} + \xi'_{X_1}$ 
elseif  $\zeta'_{X_0} = \zeta'_{X_1}$  then  $\delta'_Y = 0$  and  $\zeta'_Y = 0$  and  $\xi'_Y = 0$ 
else  $\delta'_Y = 1$  and  $\zeta'_Y = \zeta'_{X_0} \oplus \zeta'_{X_1}$  and  $\xi'_Y = 0$  endif

```

**Model (S-box).** For the S-box in Figure 3(c), the following constraint clarifies all the possible pattern propagations.

$$\delta'_Y \neq 1 \text{ and } \delta'_Y \neq 3 \text{ and } \delta'_Y \geq \delta'_X \text{ and } \delta'_Y - \delta'_X \leq 1$$

To exploit the information of  $N^*$ , we should maintain a counter for the value of  $\xi'_{X_i}$  that memories all classes that have been occupied. To escape manually allocating the value of  $\xi'_{X_i}$ , we invoke the function `alldifferent_except_0()` in Minizinc to set the value of  $\xi'_{X_i}$ , automatically. Please find the file titled ‘Program1\_Toy\_Feistel.mzn’ for more details.

### A.3.3 Clarifying the Searching Scopes of the Input and Output Patterns

Also, we do not fix the format of the input pattern and only claim that the input difference is nonzero. Then, the new model can accomplish an exhaustive search in the solving phase. For the searching scope of the output pattern, we also set it to assert a certain word satisfies a predetermined differential pattern.

Please find a toy example regarding the 2-branch Feistel structure in Figure 4 in the file titled ‘Program1\_Toy\_Feistel.mzn’ for an illustration of the generalised model.

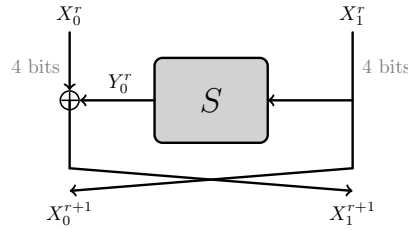


Figure 4: Toy example.

## B RK DL Cryptanalysis of AES-192

### B.1 Artificial Randomness Property

The new distinguisher depends on the artificial randomness property that  $\Delta x_5^S[1, 0]$  takes all 255 nonzero values with equal probability. An intuitive explanation for this assumption

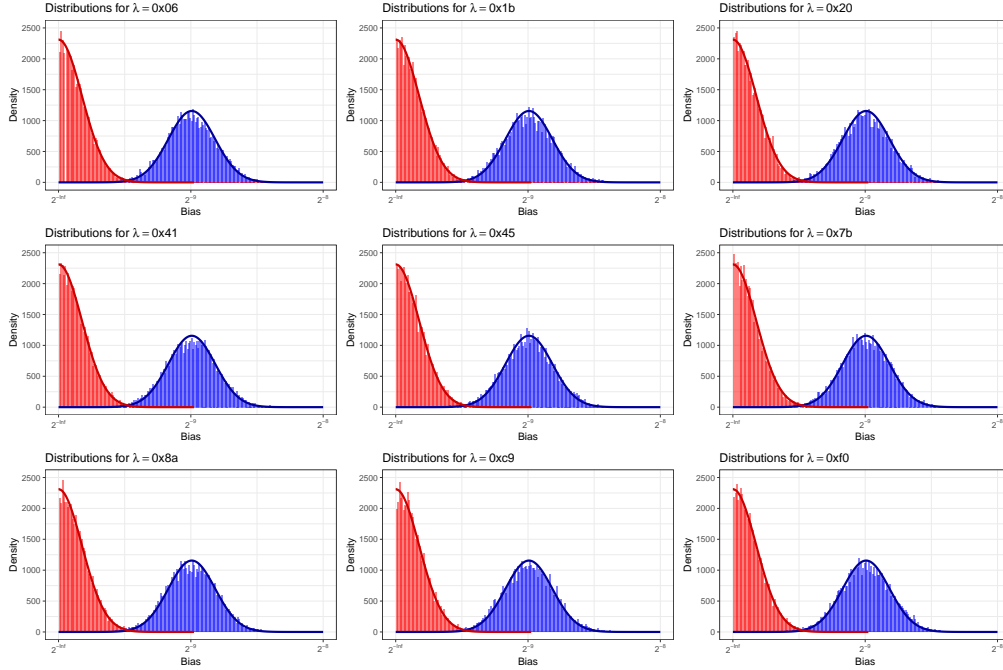
is that for any two consecutive S-boxes in the differential propagation

$$\Delta x_3^I[0,0] = \alpha \xrightarrow{S} \Delta x_3^S[0,0], (0x02 \cdot \Delta x_3^S[0,0]) \xrightarrow{S} \tilde{\delta}, \tilde{\delta} \xrightarrow{S} \Delta x_5^S[1,0],$$

the MixColumns operation in between them ensures that the input value of the second S-box not only relies on the output value of the first S-box but also relies on the values of some other state bytes with zero difference. Thus, the input values of the three S-boxes can be regarded as random variables, and the three differential propagations are independent to some extent. Additionally, the differential distribution table of the S-box for AES exhibits an almost uniform property. With these observations, we conjecture that this assumption is reasonable. Besides, we check the distribution of  $\Delta x_5^S[1,0]$  with random tests<sup>3</sup>. The experimental results show that  $\Delta x_5^S[1,0]$  almost takes 255 nonzero values with equal probability. The intuitive explanation and the experimental verification give support for the assumption.

## B.2 Verification of the Statistical Model

The theoretical distributions are verified with 10000 random keys. For each key, the statistics are approximately estimated with  $2^{21}$  pairs drawn at random. The experimental results under nine randomly selected values of  $\lambda$  are shown in Figure 5. It can be noticed that the test results fit very well with the theoretical distributions.



**Figure 5:** The distributions of the statistic under different settings. The histograms in blue and red are the experimental distributions of the statistic under the right key and wrong key assumptions, respectively. The blue and red curves are the probability density functions of the theoretical distributions under corresponding assumptions.

## B.3 Improved Related-Key DL Attack of AES-192

Please find in Figure 6 for an illustration of the key-recovery attack on AES-192. To obtain the value of  $x_5^W[1,3]$ , we need to know four bytes of  $wk_6$ , which are  $wk_6[0,3]$ ,  $wk_6[1,2]$ ,

<sup>3</sup>Please find the file titled ‘Verify\_Randomness\_Property.cpp’ for more details about the test.



$wk_6[2, 1]$ , and  $wk_6[3, 0]$ , and all the 16 bytes of  $k_7$ . According to the key schedule of AES-192, the eight bytes in the first two columns of  $k_6$ , as well as  $wk_6$ , are known if  $k_7$  is known. So, we only need to guess the two bytes  $wk_6[0, 3]$  and  $wk_6[1, 2]$  of  $wk_6$ . The attack procedure proceeds as follows.

1. Generating  $N$  pairs of plaintexts  $(P, P')$  so that the difference  $P \oplus P'$  satisfies the input difference of the distinguisher.
2. Asking for the encryptions of the plaintexts under  $K$  and  $K'$ , respectively.
3. Guessing the values of  $wk_6[0, 3]$ ,  $wk_6[1, 2]$  and the 16 bytes of  $k_7$ .
  - (a) Initialising a 32-bit counter  $\Sigma$  to zero.
  - (b) Using the guessed subkeys to perform partial decryption for each ciphertext pair so that we can compute the value of  $x_5^W[1, 3]$ .
  - (c) Calculating  $\lambda \cdot \Delta x_5^W[1, 3]$  and incrementing  $\Sigma$  by one if it equals 0.
  - (d) Regarding the guessed key as a candidate if  $|\Sigma/N - 0.5| > \tau$ .

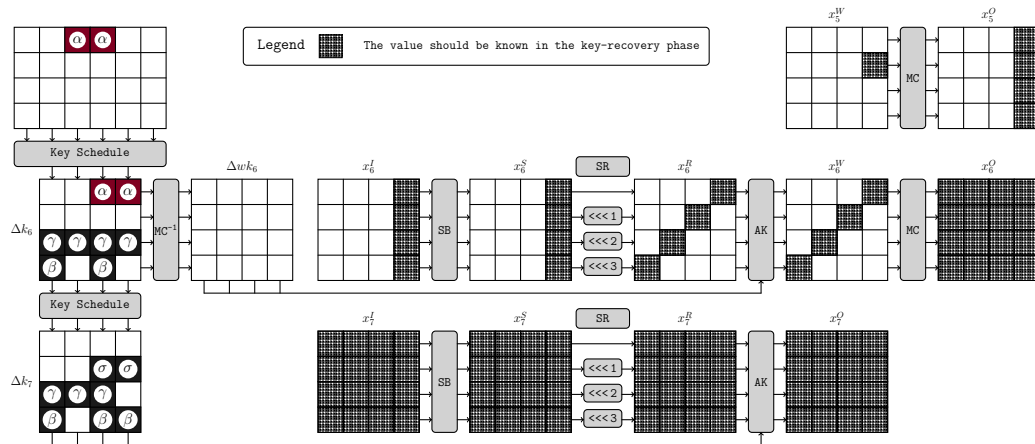
In the above attack, the value of  $\alpha$  can be determined by the attacker. The values of  $\gamma$  and  $\sigma$  are identified if  $\beta$  is known since we have

$$\begin{aligned}\gamma &= S(k_7[3, 0] \oplus k_7[3, 1]) \oplus S(k_7[3, 0] \oplus k_7[3, 1] \oplus \beta), \\ \sigma &= S(k_7[2, 1]) \oplus S(k_7[2, 1] \oplus \gamma).\end{aligned}$$

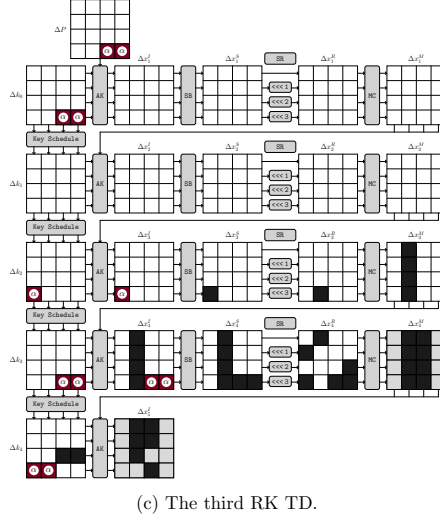
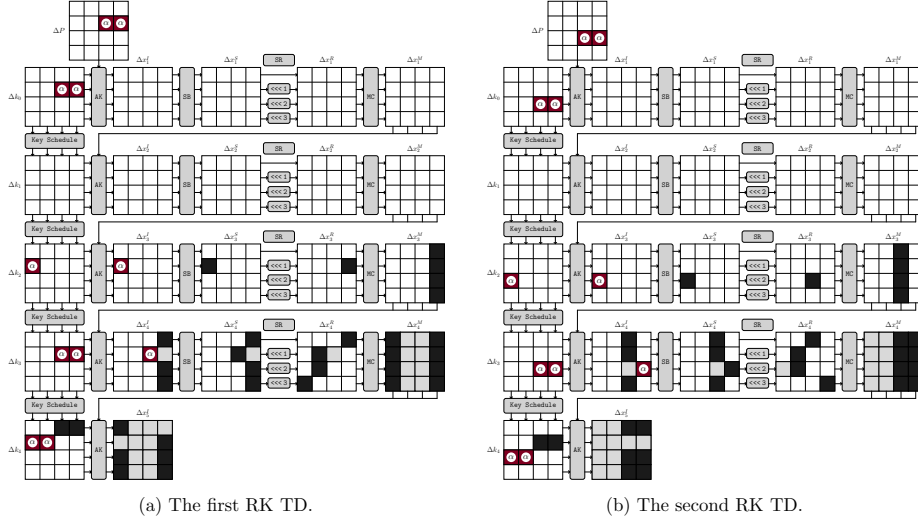
Given the value of  $\alpha$ ,  $\beta$  can take 127 possible values. Therefore, we should repeat the attack stated above for all possible values of  $\beta$ . The total time complexity is  $2^{18 \cdot 8} \times 2N \times 2/7 \times 127 \approx 2^{170.51}$ .

#### B.4 Three 4-Round RK TDs for AES-192

Please find the new RK TDs for AES-192 in Figure 7.



**Figure 6:** Key-recovery phase of the 7-round related-key differential-linear attack.



**Figure 7:** Three related-key truncated differentials for AES-192.

## C Applications to SKINNY

### C.1 A Brief Description of SKINNY

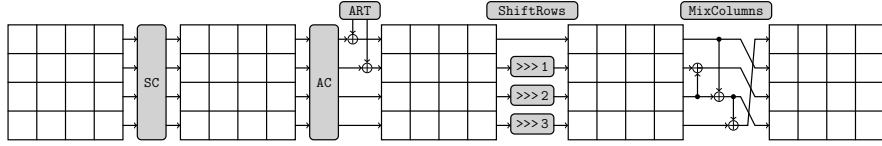
The plaintext  $m$  is divided into 16 cells  $m = m_0 || m_1 || \dots || m_{15}$ , and each  $m_i$  stands for an  $s$ -bit cell, that is  $s = 4$  for SKINNY-64-\* and  $s = 8$  for SKINNY-128-\*. The internal state of the cipher is initialised as

$$IS = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}.$$

As in Figure 8, each round of SKINNY consists of four operations, which are SubCells(SC), AddConstants(AC), AddRoundTweakey(ART), ShiftRows(SR), and MixColumns(MC) operations.

- **SubCells**: an  $s$ -bit S-box is applied to each cell of the internal state.
- **AddConstants**: the round constant derived from a 6-bit linear feedback shift register (LFSR) is combined with the state.
- **AddRoundTweakey**: the topmost two rows of the round tweakey are extracted and are added to the internal state.
- **ShiftRows**: the  $i$ -th row is rotated by  $i$  cells to the right for  $0 \leq i \leq 3$ .
- **MixColumns**: each column of the state is multiplied by the matrix  $M_{\text{SKINNY}}$ ,

$$M_{\text{SKINNY}} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

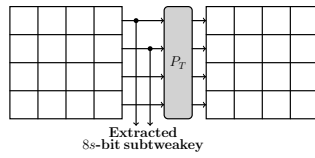


**Figure 8:** Round function of SKINNY.

The tweakey schedule of SKINNY- $n$ - $n$  is shown in Figure 9. The  $n$ -bit tweakey  $TK1$  is loaded row-wise like the internal state. In each round, after extracting the two topmost rows of the tweakey, the tweakey is updated by a cell permutation  $P_T$ ,

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

This permutation ensures that all cells of the tweakey get involved in two consecutive rounds. See [BJK<sup>+</sup>16] for more details about SKINNY.



**Figure 9:** Tweakey schedule of SKINNY- $n$ - $n$ .

## C.2 12.5-Round Impossible Differentials for SKINNY

With Algorithm 1, we have

$$\begin{aligned} \mathcal{E}(Z) &= (4.5, 4.5, 4.5, 4.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5), \\ \mathcal{E}(N) &= (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), \\ \mathcal{E}(N^*) &= (5.5, 5.5, 5.5, 5.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5), \\ \mathcal{D}(Z) &= (6, 6, 6, 6, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6), \\ \mathcal{D}(N) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ \mathcal{D}(N^*) &= (7, 7, 7, 7, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7). \end{aligned}$$

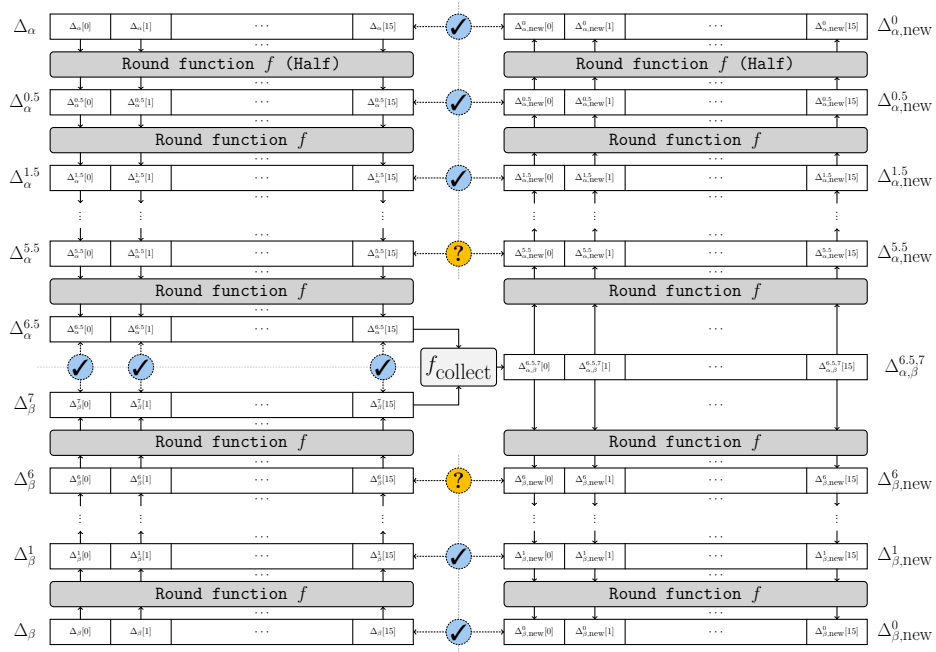
The twelve 12.5-round impossible differentials are listed as follows

ID000: (0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	( $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID001: (0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	(0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID002: (0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	(0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID003: (0, 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	(0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID004: (0, 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	(0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID005: (0, 0, 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0, 0, 0)	$\nrightarrow$	(0, 0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID006: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0)	$\nrightarrow$	(0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, $\beta$ , 0, 0),
ID007: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0)	$\nrightarrow$	(0, 0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, $\beta$ , 0, 0),
ID008: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0)	$\nrightarrow$	( $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID009: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, 0, 0, $\alpha$ , 0, 0, $\alpha$ , 0)	$\nrightarrow$	(0, 0, 0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, $\beta$ , 0, 0),
ID010: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, $\alpha$ , 0)	$\nrightarrow$	( $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),
ID011: (0, 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, $\alpha$ , 0, 0, 0, 0, $\alpha$ , 0)	$\nrightarrow$	(0, $\beta$ , 0, 0, 0, 0, 0, 0, 0, 0, $\beta$ , 0, 0, 0, $\beta$ , 0, 0, 0),

where  $\alpha$  and  $\beta$  can take arbitrary nonzero differences. The illustrations for the trails can be found in the file titled ‘IDs\_SKINNY.pdf’.

### C.3 Provable Security against Impossible Differential

**Theorem 1.** *Under the keyed (uniform) bijective S-box assumption, 13.5-round encryption of SKINNY is secure against impossible differential with arbitrary nonzero input and output differences.*



**Figure 10:** Analysing the existence of 13.5-round IDs with  $R'_1 = 6.5$ .

*Proof.* We apply proof by contradiction. Suppose that there exists a 13.5-round impossible differential  $\alpha \nrightarrow \beta$  for SKINNY. Then, the inconsistency of the trail must be Type II contradiction (see Figure 4(b) in the main body of the paper), and we have  $R_1 + 1 + R_2 = 13.5$ . Denote  $R_1 + 1$  as  $R'_1$ .

Firstly, we show that when  $R'_1 = 6.5$  and  $R_2 = 7$ , all combinations of input and output differences result in possible differentials. With the property of the matrix in the

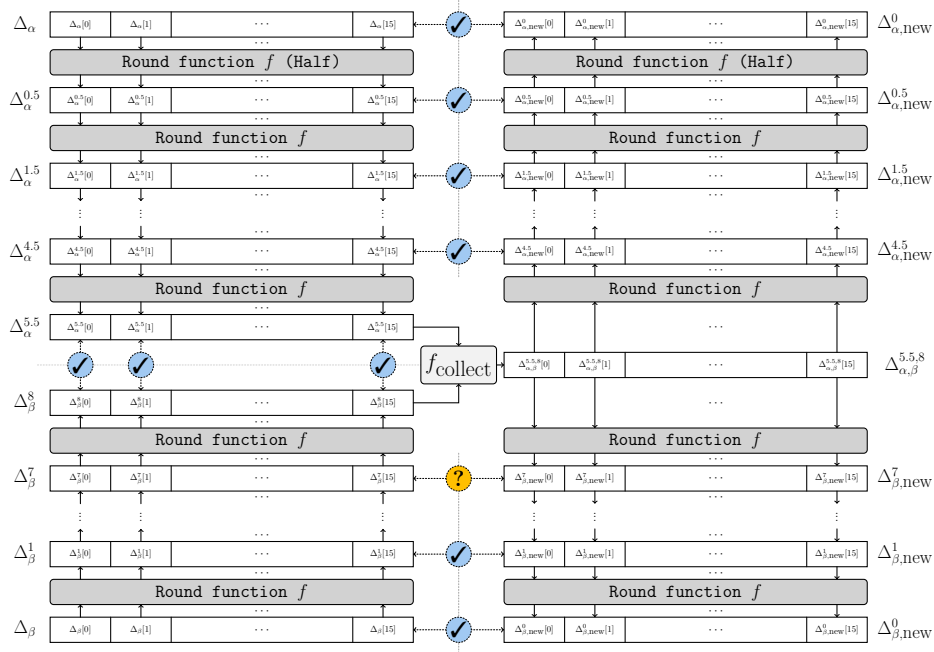


Figure 11: Analysing the existence of 13.5-round IDs with  $R'_1 = 5.5$ .

MixColumns operation, differential patterns of internal states entirely composed of  $N^*$  and  $U$  will turn into the pattern only consisting of  $U$  after two rounds of encryption/decryption. Thus, with the notations in Figure 10, if we can prove that  $(\Delta_\alpha^{5.5}, \Delta_{\alpha,new}^{5.5})$  and  $(\Delta_\beta^6, \Delta_{\beta,new}^6)$  are pairs of compatible patterns,  $\alpha \rightarrow \beta$  must be a 13.5-round possible differential. From the output returned by Algorithm 1, the best status of  $\Delta_{\alpha,\beta}^{6.5,7} = f_{\text{collect}}(\Delta_\alpha^{6.5}, \Delta_\beta^7)$  is  $(N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*, N^*)$ . After one round of decryption,  $\Delta_{\alpha,new}^{5.5}[i]$  must equal to  $U$  for all  $i \geq 4$ , while for the remaining values of  $i$ ,  $\Delta_{\alpha,new}^{5.5}[i]$  may take  $N^*$ . However, it can be noted that  $\Delta_\alpha^{5.5}[i] \neq Z$  when  $0 \leq i \leq 3$ . Thus,  $\Delta_\alpha^{5.5}$  is compatible with  $\Delta_{\alpha,new}^{5.5}$ . Similarly, we can prove that  $\Delta_\beta^6$  is compatible with  $\Delta_{\beta,new}^6$ . Therefore, under the decomposition with  $R'_1 = 6.5$  and  $R_2 = 7$ , we cannot create impossible differentials.

As in Figure 11, if  $R'_1 = 5.5$ ,  $\Delta_{\alpha,\beta}^{R'_1, R_2} = f_{\text{collect}}(\Delta_\alpha^{R'_1}, \Delta_\beta^{R_2})$  must equal  $\Delta_\alpha^{R'_1}$  since  $\mathcal{D}_i(X) \leq 7$  for all  $0 \leq i \leq \ell - 1$  and  $X \in \mathcal{U}^*$ . Hence, the backward propagation of  $\Delta_{\alpha,\beta}^{R'_1, R_2}$  will not result in contradiction with  $\Delta_\alpha^{r_1}$ ,  $0 \leq r_1 \leq R'_1 - 1$ . On the other hand, we have  $\Delta_{\beta,new}^7 = \Delta_\alpha^{6.5}$ , which carries less information than  $\Delta_{\alpha,\beta}^{6.5,7}$ . Thus, in the forward propagating phase of  $\Delta_\alpha^{6.5}$ , we cannot identify inconsistencies between  $\Delta_\beta^{r_2}$  and  $\Delta_{\beta,new}^{r_2}$  for  $0 \leq r_2 \leq 7$ . That is to say, under the decomposition  $(R'_1, R_2) = (5.5, 8)$ , there is no impossible differential.

Recursively, all decompositions with  $R'_1 < 5.5$  ensure the nonexistence of impossible differential. Likewise, for the cases where  $R_2 < 7$ , we can draw the same conclusion. Then, we prove that 13.5-round impossible differential does not exist unless the information of S-box is considered.  $\square$

#### C.4 Provable Security of SKINNY- $n$ - $n$ against RK ID

Denote  $\Delta_{\alpha,\kappa}^r$  the differential pattern after  $r$  rounds of encryption under the input difference  $\alpha$  and tweakey difference  $\kappa$ . For all possible values of  $\kappa$  with the  $t$ -th subblock being the

unique active cell, the maximum number of encryption rounds such that the differential pattern of the  $i$ -th subblock in the internal state follows the pattern  $\mathbf{X}$  is denoted as

$$\mathcal{E}_t(\mathbf{X})[i] = \max_{\alpha \neq 0} \left\{ r \mid \Delta_{\alpha, \kappa}^r[i] = \mathbf{X}, \kappa[t] \neq 0, \kappa[j] = 0 \text{ for all } j \neq t \right\}.$$

Unlike the single-tweakey attack scenario, the tweakey difference of the inner round is affected by the relative position of the current round regarding the starting round. Since we aim to provide the provable security for 13.5-round of encryption, we fix the full round as 13.5. For the output difference  $\beta$  and the tweakey difference  $\kappa$ , let  $\Delta_{\beta, \kappa}^{13.5-r}$  be the differential pattern after  $r$  rounds of decryption. For all possible  $\kappa$ 's with the  $t$ -th subblock being the unique active nibble, the maximum number of decryption rounds such that the differential pattern of the  $i$ -th subblock in the internal state satisfies the pattern  $\mathbf{X}$  is denoted as

$$\mathcal{D}_t(\mathbf{X})[i] = \max_{\beta \neq 0} \left\{ r \mid \Delta_{\beta, \kappa}^{13.5-r}[i] = \mathbf{X}, \kappa[t] \neq 0, \kappa[j] = 0 \text{ for all } j \neq t \right\}.$$

With these notations, we prove the following theorem.

**Theorem 2.** *13.5-round SKINNY- $n$ - $n$  is secure against related-tweakey impossible differential with arbitrary nonzero input and output differences under the following assumptions:*

- the  $S$ -box satisfies keyed (uniform) bijective assumption;
- the difference of tweakey only has one active cell.

*Proof.* After invoking Algorithm 1 for different values of  $t$ , the vectors  $\mathcal{E}_t(\mathbf{X})$  and  $\mathcal{D}_t(\mathbf{X})$  can be determined. Then, the nonexistence of 13.5-round impossible differential is discussed regarding different values of  $t$ . We take  $t = 0$  as an example. The results returned by Algorithm 1 is

$$\begin{aligned} \mathcal{E}_0(\mathbf{Z}) &= (4.5, 0, 0, 4.5, 5.5, 3.5, 0, 5.5, 5.5, 3.5, 4.5, 0, 5.5, 3.5, 0, 4.5), \\ \mathcal{E}_0(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ \mathcal{E}_0(\mathbf{N}^*) &= (4.5, 3.5, 5.5, 4.5, 5.5, 4.5, 6.5, 5.5, 6.5, 5.5, 4.5, 5.5, 3.5, 6.5, 4.5), \\ \mathcal{D}_0(\mathbf{Z}) &= (4, 6, 6, 5, 0, 5, 5, 4, 6, 5, 4, 4, 0, 4, 5, 6), \\ \mathcal{D}_0(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ \mathcal{D}_0(\mathbf{N}^*) &= (7, 6, 6, 5, 6, 5, 5, 4, 6, 5, 4, 6, 6, 5, 5, 6). \end{aligned}$$

If the 13.5-round impossible differential exists and can be decomposed into two truncated differentials with  $R_1 = 6.5$  and  $R_2 = 7$ , the best case for  $\Delta_{\alpha, \beta, \kappa}^{6.5, 6.5} = f_{\text{collect}}(\Delta_{\alpha, \kappa}^{6.5}, \Delta_{\beta, \kappa}^{13.5-7})$  is  $(\mathbf{N}^*, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{N}^*, \mathbf{U}, \mathbf{N}^*, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{U}, \mathbf{N}^*, \mathbf{U})$ . When  $\Delta_{\alpha, \beta, \kappa}^{6.5, 6.5}$  is propagated in the forward and backward directions, the differential pattern  $\Delta_{\alpha, \kappa, \text{new}}^{5.5}$  will not contradict with  $\Delta_{\alpha, \kappa}^{5.5}$ , and  $\Delta_{\beta, \kappa, \text{new}}^{13.5-6}$  will not contradict with  $\Delta_{\beta, \kappa}^{13.5-6}$ . Thus, under the decomposition with  $R_1 = 6.5$  and  $R_2 = 7$ , we cannot create impossible differentials. Likewise, we can show the nonexistence of impossible differential for all possible combinations of  $(R_1, R_2)$ .

After excluding the existence of impossible differential for all values of  $t$ , we prove that if the tweakey only has one active cell, 13.5-round SKINNY- $n$ - $n$  is secure against related-tweakey impossible differential with arbitrary nonzero input and output differences. For reference, we list all values of  $\mathcal{E}_t(\mathbf{X})$  and  $\mathcal{D}_t(\mathbf{X})$  in the following.

$$\begin{aligned} \mathcal{E}_1(\mathbf{Z}) &= (0, 4.5, 3.5, 0, 0, 5.5, 3.5, 3.5, 4.5, 0, 5.5, 3.5, 0, 4.5, 3.5, 3.5), \\ \mathcal{E}_1(\mathbf{N}) &= (0, 0, 4.5, 0, 0, 0, 4.5, 0, 0, 0, 0, 0, 0, 0, 4.5, 0), \\ \mathcal{E}_1(\mathbf{N}^*) &= (5.5, 4.5, 3.5, 3.5, 6.5, 5.5, 5.5, 4.5, 5.5, 4.5, 6.5, 4.5, 6.5, 5.5, 5.5, 3.5), \\ \mathcal{D}_1(\mathbf{Z}) &= (5, 4, 5, 6, 4, 0, 4, 5, 5, 6, 5, 4, 5, 6, 4, 5), \\ \mathcal{D}_1(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ \mathcal{D}_1(\mathbf{N}^*) &= (5, 6, 6, 6, 4, 5, 5, 5, 5, 6, 5, 6, 7, 6, 7, 6). \end{aligned}$$

$\mathcal{E}_2(\mathbb{Z})$	=	(4.5, 3.5, 3.5, 0, 5.5, 4.5, 4.5, 3.5, 4.5, 3.5, 5.5, 0, 5.5, 3.5, 4.5, 0),
$\mathcal{E}_2(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4.5, 0, 0, 0, 0),
$\mathcal{E}_2(\mathbb{N}^*)$	=	(5.5, 4.5, 4.5, 4.5, 6.5, 5.5, 5.5, 5.5, 6.5, 6.5, 5.5, 5.5, 6.5, 4.5, 5.5, 4.5),
$\mathcal{D}_2(\mathbb{Z})$	=	(5, 6, 5, 4, 4, 5, 4, 0, 5, 4, 5, 6, 4, 5, 5, 6),
$\mathcal{D}_2(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_2(\mathbb{N}^*)$	=	(6, 6, 5, 5, 5, 5, 4, 5, 5, 6, 5, 6, 6, 6, 7, 6),
$\mathcal{E}_3(\mathbb{Z})$	=	(3.5, 3.5, 0, 4.5, 4.5, 3.5, 3.5, 5.5, 5.5, 5.5, 4.5, 4.5, 3.5, 3.5, 0, 5.5),
$\mathcal{E}_3(\mathbb{N})$	=	(0, 0, 0, 0, 0, 4.5, 0, 0, 0, 0, 0, 0, 0, 4.5, 0, 0),
$\mathcal{E}_3(\mathbb{N}^*)$	=	(4.5, 3.5, 4.5, 5.5, 5.5, 3.5, 5.5, 6.5, 5.5, 5.5, 4.5, 6.5, 5.5, 3.5, 4.5, 6.5),
$\mathcal{D}_3(\mathbb{Z})$	=	(6, 6, 5, 4, 5, 5, 4, 0, 5, 4, 4, 6, 4, 5, 6, 5),
$\mathcal{D}_3(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_3(\mathbb{N}^*)$	=	(6, 6, 5, 7, 5, 5, 4, 6, 5, 4, 6, 6, 6, 7, 6, 6),
$\mathcal{E}_4(\mathbb{Z})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4.5, 0, 0, 0, 0, 0),
$\mathcal{E}_4(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0),
$\mathcal{E}_4(\mathbb{N}^*)$	=	(4.5, 4.5, 5.5, 5.5, 5.5, 5.5, 6.5, 5.5, 5.5, 6.5, 5.5, 6.5, 4.5, 4.5, 5.5, 5.5),
$\mathcal{D}_4(\mathbb{Z})$	=	(6, 5, 5, 5, 5, 4, 4, 4, 4, 5, 6, 5, 5, 5, 6, 4),
$\mathcal{D}_4(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_4(\mathbb{N}^*)$	=	(7, 6, 6, 6, 6, 5, 5, 5, 5, 5, 6, 6, 6, 7, 6, 7),
$\mathcal{E}_5(\mathbb{Z})$	=	(4.5, 4.5, 0, 0, 5.5, 5.5, 3.5, 0, 0, 5.5, 3.5, 4.5, 4.5, 5.5, 3.5, 0),
$\mathcal{E}_5(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{E}_5(\mathbb{N}^*)$	=	(5.5, 4.5, 3.5, 5.5, 6.5, 5.5, 4.5, 6.5, 3.5, 6.5, 4.5, 5.5, 5.5, 5.5, 3.5, 6.5),
$\mathcal{D}_5(\mathbb{Z})$	=	(5, 4, 6, 6, 4, 0, 5, 5, 4, 5, 5, 4, 4, 5, 4, 4),
$\mathcal{D}_5(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_5(\mathbb{N}^*)$	=	(7, 7, 6, 6, 6, 6, 5, 5, 6, 6, 5, 6, 6, 6, 6, 7),
$\mathcal{E}_6(\mathbb{Z})$	=	(0, 4.5, 3.5, 3.5, 3.5, 5.5, 4.5, 3.5, 4.5, 4.5, 5.5, 5.5, 0, 5.5, 3.5, 3.5),
$\mathcal{E}_6(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 4.5, 0, 0, 0, 0, 0, 0, 0, 4.5),
$\mathcal{E}_6(\mathbb{N}^*)$	=	(4.5, 5.5, 4.5, 3.5, 5.5, 6.5, 5.5, 3.5, 4.5, 6.5, 5.5, 5.5, 4.5, 6.5, 5.5, 3.5),
$\mathcal{D}_6(\mathbb{Z})$	=	(5, 5, 6, 5, 4, 4, 5, 4, 6, 5, 4, 5, 6, 6, 5, 6),
$\mathcal{D}_6(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_6(\mathbb{N}^*)$	=	(6, 7, 7, 6, 5, 6, 6, 5, 6, 6, 6, 7, 6, 7, 5, 6),
$\mathcal{E}_7(\mathbb{Z})$	=	(0, 0, 4.5, 4.5, 3.5, 0, 5.5, 5.5, 3.5, 4.5, 0, 5.5, 3.5, 0, 4.5, 5.5),
$\mathcal{E}_7(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{E}_7(\mathbb{N}^*)$	=	(3.5, 5.5, 5.5, 4.5, 3.5, 6.5, 5.5, 5.5, 4.5, 4.5, 5.5, 3.5, 6.5, 5.5, 5.5, 5.5),
$\mathcal{D}_7(\mathbb{Z})$	=	(6, 5, 4, 6, 5, 4, 0, 5, 4, 4, 6, 5, 5, 6, 5, 4),
$\mathcal{D}_7(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_7(\mathbb{N}^*)$	=	(6, 7, 7, 6, 5, 6, 6, 5, 6, 6, 6, 7, 5, 6, 6, 5),
$\mathcal{E}_8(\mathbb{Z})$	=	(3.5, 0, 0, 3.5, 4.5, 0, 0, 4.5, 4.5, 0, 3.5, 0, 4.5, 0, 0, 3.5),
$\mathcal{E}_8(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{E}_8(\mathbb{N}^*)$	=	(3.5, 0, 4.5, 3.5, 4.5, 3.5, 5.5, 4.5, 5.5, 4.5, 4.5, 3.5, 4.5, 0, 5.5, 3.5),
$\mathcal{D}_8(\mathbb{Z})$	=	(5, 7, 7, 6, 4, 6, 6, 5, 7, 6, 5, 5, 4, 5, 6, 7),
$\mathcal{D}_8(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_8(\mathbb{N}^*)$	=	(8, 7, 7, 6, 7, 6, 6, 5, 7, 6, 5, 7, 7, 6, 6, 7),
$\mathcal{E}_9(\mathbb{Z})$	=	(0, 3.5, 0, 0, 0, 4.5, 0, 0, 3.5, 0, 4.5, 0, 0, 3.5, 0, 0),
$\mathcal{E}_9(\mathbb{N})$	=	(0, 0, 3.5, 0, 0, 0, 3.5, 0, 0, 0, 0, 0, 0, 0, 3.5, 0),
$\mathcal{E}_9(\mathbb{N}^*)$	=	(4.5, 3.5, 0, 0, 5.5, 4.5, 4.5, 3.5, 4.5, 3.5, 5.5, 3.5, 5.5, 4.5, 4.5, 0),
$\mathcal{D}_9(\mathbb{Z})$	=	(6, 5, 6, 7, 5, 4, 5, 6, 6, 7, 6, 5, 6, 7, 5, 6),
$\mathcal{D}_9(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_9(\mathbb{N}^*)$	=	(6, 7, 7, 7, 5, 6, 6, 6, 6, 7, 6, 7, 8, 7, 8, 7),
$\mathcal{E}_{10}(\mathbb{Z})$	=	(3.5, 0, 0, 0, 4.5, 3.5, 3.5, 0, 3.5, 0, 4.5, 0, 4.5, 0, 3.5, 0),
$\mathcal{E}_{10}(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3.5, 0, 0, 0, 0),
$\mathcal{E}_{10}(\mathbb{N}^*)$	=	(4.5, 3.5, 3.5, 3.5, 5.5, 4.5, 4.5, 4.5, 5.5, 5.5, 4.5, 4.5, 5.5, 3.5, 4.5, 3.5),
$\mathcal{D}_{10}(\mathbb{Z})$	=	(6, 7, 6, 5, 5, 6, 5, 4, 6, 5, 6, 7, 5, 6, 6, 7),
$\mathcal{D}_{10}(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_{10}(\mathbb{N}^*)$	=	(7, 7, 6, 6, 6, 6, 5, 6, 6, 7, 6, 7, 7, 7, 8, 7),
$\mathcal{E}_{11}(\mathbb{Z})$	=	(0, 0, 0, 3.5, 3.5, 0, 0, 4.5, 4.5, 4.5, 3.5, 3.5, 0, 0, 0, 4.5),
$\mathcal{E}_{11}(\mathbb{N})$	=	(0, 0, 0, 0, 0, 3.5, 0, 0, 0, 0, 0, 0, 0, 3.5, 0, 0),
$\mathcal{E}_{11}(\mathbb{N}^*)$	=	(3.5, 0, 3.5, 4.5, 4.5, 0, 4.5, 5.5, 4.5, 4.5, 3.5, 5.5, 4.5, 0, 3.5, 5.5),
$\mathcal{D}_{11}(\mathbb{Z})$	=	(7, 7, 6, 5, 6, 6, 5, 4, 6, 5, 5, 7, 5, 6, 7, 6),
$\mathcal{D}_{11}(\mathbb{N})$	=	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
$\mathcal{D}_{11}(\mathbb{N}^*)$	=	(7, 7, 6, 8, 6, 6, 5, 7, 6, 5, 7, 7, 7, 8, 7, 7),

$$\begin{aligned}
\mathcal{E}_{12}(\mathbf{Z}) &= (0, 0, 3.5, 0, 3.5, 0, 4.5, 3.5, 3.5, 3.5, 0, 4.5, 3.5, 0, 4.5, 0), \\
\mathcal{E}_{12}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3.5, 0, 0, 0, 0), \\
\mathcal{E}_{12}(\mathbf{N}^*) &= (3.5, 3.5, 4.5, 4.5, 4.5, 4.5, 5.5, 4.5, 4.5, 5.5, 4.5, 5.5, 3.5, 3.5, 4.5, 4.5), \\
\mathcal{D}_{12}(\mathbf{Z}) &= (7, 6, 6, 6, 6, 5, 5, 5, 5, 6, 7, 6, 6, 6, 7, 5), \\
\mathcal{D}_{12}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{D}_{12}(\mathbf{N}^*) &= (8, 7, 7, 7, 7, 6, 6, 6, 6, 6, 7, 7, 7, 8, 7, 8), \\
\\
\mathcal{E}_{13}(\mathbf{Z}) &= (3.5, 3.5, 0, 0, 4.5, 4.5, 0, 0, 0, 4.5, 0, 3.5, 3.5, 4.5, 0, 0), \\
\mathcal{E}_{13}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{E}_{13}(\mathbf{N}^*) &= (4.5, 3.5, 0, 4.5, 5.5, 4.5, 3.5, 5.5, 0, 5.5, 3.5, 4.5, 4.5, 4.5, 0, 5.5), \\
\mathcal{D}_{13}(\mathbf{Z}) &= (6, 5, 7, 7, 5, 4, 6, 6, 5, 6, 6, 5, 5, 6, 5, 5), \\
\mathcal{D}_{13}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{D}_{13}(\mathbf{N}^*) &= (8, 8, 7, 7, 7, 7, 6, 6, 7, 7, 6, 7, 7, 7, 7, 8), \\
\\
\mathcal{E}_{14}(\mathbf{Z}) &= (0, 3.5, 0, 0, 0, 4.5, 3.5, 0, 3.5, 3.5, 4.5, 4.5, 0, 4.5, 0, 0), \\
\mathcal{E}_{14}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 3.5, 0, 0, 0, 0, 0, 0, 3.5), \\
\mathcal{E}_{14}(\mathbf{N}^*) &= (3.5, 4.5, 3.5, 0, 4.5, 5.5, 4.5, 0, 3.5, 5.5, 4.5, 4.5, 3.5, 5.5, 4.5, 0), \\
\mathcal{D}_{14}(\mathbf{Z}) &= (6, 6, 7, 6, 5, 5, 6, 5, 7, 6, 5, 6, 7, 7, 6, 6), \\
\mathcal{D}_{14}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{D}_{14}(\mathbf{N}^*) &= (7, 8, 8, 7, 6, 7, 7, 6, 7, 7, 7, 8, 7, 8, 6, 7), \\
\\
\mathcal{E}_{15}(\mathbf{Z}) &= (0, 0, 3.5, 3.5, 0, 0, 4.5, 4.5, 0, 3.5, 0, 4.5, 0, 0, 3.5, 4.5), \\
\mathcal{E}_{15}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{E}_{15}(\mathbf{N}^*) &= (0, 4.5, 4.5, 3.5, 0, 5.5, 4.5, 4.5, 4.5, 3.5, 3.5, 4.5, 0, 5.5, 4.5, 4.5), \\
\mathcal{D}_{15}(\mathbf{Z}) &= (7, 6, 5, 7, 6, 5, 4, 6, 5, 5, 7, 6, 6, 7, 6, 5), \\
\mathcal{D}_{15}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{D}_{15}(\mathbf{N}^*) &= (7, 8, 8, 7, 6, 7, 7, 6, 7, 7, 7, 8, 6, 7, 7, 6).
\end{aligned}$$

□

## C.5 11.5-round Zero-Correlation Linear Approximations for SKINNY

After invoking Algorithm 1, we obtain the values of the vectors,

$$\begin{aligned}
\mathcal{E}(\mathbf{Z}) &= (4.5, 4.5, 4.5, 4.5, 4.5, 4.5, 4.5, 4.5, 5.5, 5.5, 5.5, 5.5, 3.5, 3.5, 3.5, 3.5), \\
\mathcal{E}(\mathbf{N}) &= (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), \\
\mathcal{E}(\mathbf{N}^*) &= (5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 5.5, 6.5, 6.5, 6.5, 6.5, 4.5, 4.5, 4.5, 4.5), \\
\mathcal{D}(\mathbf{Z}) &= (5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 6, 6, 6, 6), \\
\mathcal{D}(\mathbf{N}) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\
\mathcal{D}(\mathbf{N}^*) &= (6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 7, 7, 7, 7).
\end{aligned}$$

The sixteen 11.5-round zero-correlation linear approximations are listed as follows,

$$\begin{aligned}
\text{ZC000: } (\alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0), \\
\text{ZC001: } (\alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0), \\
\text{ZC002: } (\alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, \beta, 0, 0, \beta, 0, 0, 0, \beta, 0, 0), \\
\text{ZC003: } (\alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta), \\
\text{ZC004: } (0, \alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0) &\rightharpoonup (0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0), \\
\text{ZC005: } (0, \alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0) &\rightharpoonup (0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0), \\
\text{ZC006: } (0, \alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0), \\
\text{ZC007: } (0, \alpha, 0, 0, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta), \\
\text{ZC008: } (0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha) &\rightharpoonup (0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0), \\
\text{ZC009: } (0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha) &\rightharpoonup (0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0), \\
\text{ZC010: } (0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha) &\rightharpoonup (0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0), \\
\text{ZC011: } (0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha) &\rightharpoonup (0, 0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta), \\
\text{ZC012: } (0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0), \\
\text{ZC013: } (0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0), \\
\text{ZC014: } (0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta, 0), \\
\text{ZC015: } (0, 0, 0, \alpha, 0, 0, 0, 0, 0, \alpha, 0, 0, \alpha, 0, 0) &\rightharpoonup (0, 0, 0, 0, 0, 0, 0, \beta, 0, 0, 0, \beta, 0, 0, 0, \beta),
\end{aligned}$$

where  $\alpha$  and  $\beta$  can be arbitrary nonzero masks. Please find in the file titled ‘ZCLAs\_SKINNY.pdf’ for the illustrations of the linear approximations.



## D Applications to Midori64

### D.1 A Brief Description of Midori64

Midori [BBI<sup>+</sup>15] is a family lightweight block ciphers with SPN structure. It has two versions, and both of them accept 128-bit keys but have different block sizes  $n$ . For Midori64, the block size is  $n = 64$ .

The internal state of Midori64 is represented as a  $4 \times 4$  array of nibbles

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

The round function is iterated over 16 times. Each round, except for the last one, consists of an S-layer (SubCell), a P-layer (ShuffleCell and MixColumn), and a key-addition layer (KeyAdd). Each layer updates the internal state  $S$  as follows.

- SubCell: a 4-bit S-box is applied to each cell of the state  $S$  in parallel.
- ShuffleCell: each cell of the state is permuted as

$$(s_0, s_1, \dots, s_{15}) \leftarrow (s_0, s_{10}, s_5, s_{15}, s_{14}, s_4, s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8).$$

- MixColumn:  $M_{\text{Midori}}$  is applied to every column of the state, where

$$M_{\text{Midori}} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

- KeyAdd: the  $i$ -th round key  $RK_i$  is XORed to the state.

An illustration for the round function can be found in Figure 12. See [BBI<sup>+</sup>15] for more details about the encryption algorithm.

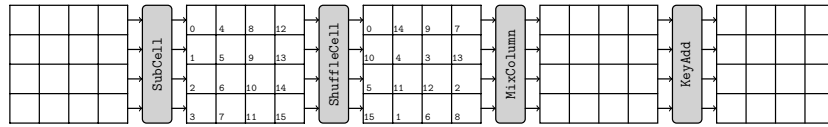


Figure 12: Round function of Midori64.

### D.2 Output of Algorithm 1

With Algorithm 1, we have

$$\begin{aligned} \mathcal{E}(Z) &= (2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5), \\ \mathcal{E}(N) &= (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), \\ \mathcal{E}(N^*) &= (3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5), \\ \mathcal{D}(Z) &= (3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3), \\ \mathcal{D}(N) &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ \mathcal{D}(N^*) &= (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4). \end{aligned}$$



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

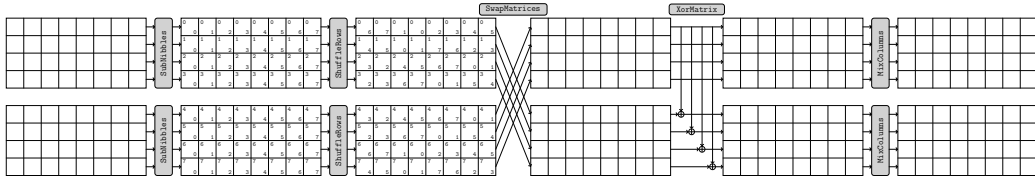




[illegible]

## E Applications to Minalpher-P

An illustration for the round function is given in Figure 13. Please refer to [STA<sup>+</sup>14] for more details about the encryption algorithm.



**Figure 13:** Round function of Minalpher-P.

### E.1 8.5-Round Impossible Differentials of Minalpher-P

The 160 8.5-round distinguishers with the minimum Hamming weight is listed as follows. The illustrations for the ten distinguishers in bold print can be found in the file titled ‘IDs\_Minalpher.pdf’.

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]



[illegible]



[illegible]



[illegible]

## References

- [AST<sup>+</sup>17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) S-boxes to optimize probability of differential characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017.
- [BBI<sup>+</sup>15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *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*, pages 411–436, 2015.
- [BJK<sup>+</sup>16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 123–153, 2016.

- [SAS<sup>+</sup>17] Aein Rezaei Shahmirzadi, Seyyed Arash Azimi, Mahmoud Salmasizadeh, Javad Mohajeri, and Mohammad Reza Aref. 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*, pages 99–104, 2017.
- [SHW<sup>+</sup>14a] Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. *Cryptology ePrint Archive*, Report 2014/747, 2014. <https://eprint.iacr.org/2014/747>.
- [SHW<sup>+</sup>14b] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, pages 158–178, 2014.
- [STA<sup>+</sup>14] Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher v1. *CAESAR Round*, 1, 2014.
- [SWW18] Ling Sun, Wei Wang, and Meiqin Wang. More accurate differential properties of LED64 and Midori64. *IACR Trans. Symmetric Cryptol.*, 2018(3):93–123, 2018.