Cryptanalysis of Reduced round SKINNY Block Cipher

Sadegh Sadeghi[1], Tahereh Mohammadi[2] and Nasour Bagheri[2,3]

[1] Department of Mathematics, Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran, S.Sadeghi.Khu@gmail.com
[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran, {T.Mohammadi,Nabgheri}@sru.ac.ir
[3] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran, Na.bagheri@gmail.com

Abstract. SKINNY is a family of lightweight tweakable block ciphers designed to have the smallest hardware footprint. In this paper, we present zero-correlation linear approximations and the related-tweakey impossible differential characteristics for different versions of SKINNY .We utilize Mixed Integer Linear Programming (MILP) to search all zero-correlation linear distinguishers for all variants of SKINNY, where the longest distinguisher found reaches 10 rounds. Using a 9-round characteristic, we present 14 and 18-round zero correlation attacks on SKINNY-64-64 and SKINNY-64-128, respectively. Also, for SKINNY-n-n and SKINNY-n-2n, we construct 13 and 15-round related-tweakey impossible differential characteristics, respectively. Utilizing these characteristics, we propose 23-round related-tweakey impossible differential cryptanalysis by applying the key recovery attack for SKINNY-n-2n and 19-round attack for SKINNY-n-n. To the best of our knowledge, the presented zero-correlation characteristics in this paper are the first attempt to investigate the security of SKINNY against this attack and the results on the related-tweakey impossible differential attack are the best reported ones.

Keywords: SKINNY · Zero-correlation linear cryptanalysis · Related-tweakey impossible differential cryptanalysis · MILP

1 Introduction

Because of the growing use of small computing devices such as RFID tags, the new challenge in the past few years has been the application of conventional cryptographic standards to small devices. Several lightweight block ciphers have been proposed to provide security for resource-constrained hardware environment. We can name PRESENT [BKL+07], SIMECK [YZS+15] SIMON, and SPECK [BTCS+15] as some of the lightweight block cipher designs.

The SKINNY [BJK+16] lightweight tweakable block cipher is introduced to compete with NSA recent design SIMON [BTCS+15] in terms of hardware/software performances. Designers of this block cipher have investigated its security against the well known attacks in such contexts as linear and differential cryptanalysis [Mat93, BS91], impossible differential cryptanalysis [BBS99, Knu98], integral attack [DKR97, KW02], and etc. In this paper, we search for zero-correlation distinguishers [BR14] and the related-tweakey impossible differential characteristics [JD03] which have been missing in the security analysis presented by the designers so far.

The impossible differential attack which was independently proposed by Biham et al. [BBS99] and Knudsen [Knu98] is one of the most popular cryptanalytic tools for block

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2018, No. 3, pp. 124–162
DOI:10.13154/tosc.v2018.i3.124-162

ciphers. Impossible differential cryptanalysis starts with finding an input difference which results in an output difference with probability 0. Related-tweakey attacks [Bih94] give a cryptanalyst the possibility to choose an appropriate relation between tweakeys and then predict the encryptions under these tweakeys. Indeed, related-tweakey impossible differential attack [JD03] is a combination of the two aforesaid attacks.

Zero-correlation linear cryptanalysis is a novel cryptanalytic approach proposed by Bogdanov and Rijmen [BR14] in 2012. In contrast to conventional linear cryptanalysis which uses linear approximations with high correlation, zero-correlation linear cryptanalysis is based on linear approximations with a correlation exactly equal to zero for all keys. The main trouble in the original proposal is the data complexity of cryptanalysis, in which almost the whole codebook is required. In a follow-up work, Bogdanov et al. proposed a novel framework to reduce the data needed using multiple independent linear approximations with a correlation of zero simultaneously [BW12]. To remove the independence assumption, a theoretical model was proposed based on the multidimensional linear distinguisher [BLNW12].

Mixed Integer Linear Programming was first introduced by Mouha et al [MWGP11] who used it to minimize the number of active s-boxes in a differential or linear characteristic. After that, Sun et al in [SHW+14a, SHW+14b] extended Mouha et al's work from byte oriented ciphers to bit oriented ciphers. They presented a method for constructing a model that finds the actual linear/differential trail with the specified number of active S-boxes. In their method, when a solution is found, the MILP model is updated in a way that a new constraint is added and the currently found solution is discarded in the next iteration. A binary variable x_i is defined for every input or output bit mask/difference and is set to 0 if the corresponding bit mask/difference is zero and 1 otherwise. At each round, a new binary variable A_j is defined for each S-box and is set to 0 if the input mask/difference of the corresponding S-box is zero and 1 otherwise. Hence, the activity of S-box is demonstrated by A_j. Now, the objective function of the MILP model is set so as to minimize the number of active S-boxes (i.e. $f = \sum_j A_j$). In order to find the minimum number of active S-boxes in a linear or differential trail, only the binary values representing the activity of S-boxes concern us. Therefore, we need to restrict these variables to be binary and can let the others to be any real number, to speed up solving the problem. However, if we aim to find the exact values of all bit-level inputs and outputs, we must restrict all state variables to be binary, which makes the model an integer programming model that is harder to solve than a mixed integer programming model. MILP has been widely used for cryptanalysis of block ciphers recently so that [FWG+16, XZBL16, AAA+15, SBA17, BJK+16, CJF+16] can be mentioned as some examples.

1.1 Related Work.

In [LGS17], the authors could attack 19, 23, and 27 rounds of SKINNY-n-n, SKINNY-n-2n, and SKINNY-n-3n respectively, using related-tweakey impossible differential and rectangle attack. They extended a 14-round related tweakey impossible differential trail (with 4.5 rounds in both forward and backward directions) and 12-round related tweakey impossible differential trail (with 4.5 rounds in backward and 2.5 rounds in forward direction) to attack 23 and 19 rounds of SKINNY-n-n and SKINNY-n-2n, respectively. In our paper,the proposed impossible differential trail consists of 15 rounds which is one round more than the one proposed in [LGS17]. Despite using the longer trail, we present 19 and 23-round attack against SKINNY-n-n and SKINNY-n-2n with less complexity. The main obstacle against reaching to an attack with more rounds in comparison with [LGS17] is the key schedule of skinny, which for attacking 25 rounds of SKINNY, the complexity is more than guessing the whole key.

The authors of [TAY17] utilized the 11-round impossible differential characteristic given in the main paper and presented 18, 20, and 22-round attack applying the key recovery

attack on SKINNY-64-64 (or 128-128), SKINNY-64-128 (or 128-265), and SKINNY-64-192 (or 128-384), respectively. In [ABC+17], the authors used an 11-round related-tweakey impossible differential characteristic to propose a 21-round attack. By assuming some tweakey bits as public key, they could extend the attack to 22 and 23 rounds (extending 6 rounds in forward and 4 rounds in backward direction). Sun et al. [SGL+17] obtained 16 related-tweakey impossible differential characteristics for 12 rounds of SKINNY-64-128 using the constraint programming (CP) method and proposed an 18-round attack on SKINNY-64-128. Untill now, no result on the security of SKINNY against zero-correlation cryptanalysis has been published prior to this work. A brief comparison of these attacks with the results of this paper and the complexities are given in Table 1.

1.2 Our Contribution.

The main purpose of this paper is to search related-tweakey impossible differential and zero-correlation linear characteristics on SKINNY. In this paper, we searched all related-tweakey impossible differential characteristics having only one active bit in the input mask and output mask or tweakeys using Mixed Integer Linear Programming (MILP) while the cell size $s = 4$. The longest related-tweakey impossible characteristics found under the assumption of having a single active bit are 13 and 14-round for SKINNY-64-64 and SKINNY-64-128, respectively. The same characteristics for SKINNY-128-128 and SKINNY-128-256 can be obtained by some slight changes. We also show that in special cases the 14-round SKINNY-64-128 distinguishers can be extended one round by assuming more than one active bits in input, output, and tweakeys. Based on the 15-round obtained distinguisher, we present key recovery attack and propose 23-round related-tweakey impossible differential attack on SKINYY-n-2n. We utilize the 13-round distinguisher to attack 19 rounds of SKINNY-n-n. Also, this paper proposes 9-round and 10-round zero correlation distinguishers on all variants of SKINNY. Based on the aforementioned 9-round zero correlation distinguisher, 14 and 18-round multidimentional zero-correlatin linear cryptanalysis is applied on SKINNY-64-64 and SKINNY-64-128, respectively. Our results are shown in Table 1.

Table 1: Summary of the main results of attacks on SKINNY, where ID, RK-ID, and ZC denote impossible differential, related-key(tweakey) impossible differential, and zero correlation cryptanalysis, respectively.

Vers.	n	Attack	# Rounds	$\log_2$(Time)	$\log_2$(Data)	$\log_2$(Memory)	Ref.
n-2n	64	ID	20	121.08	47.69	74.69	[TAY17]
		RK-ID	23	79†	-	-	[ABC+17]
		RK-ID	23	125.91	62.47	124	[LGS17]
		RK-ID	23	124	62.47	77.47	this paper
		ZC	18	126	62.68	64	this paper
	128	ID	20	245.72	92.1	147.1	[TAY17]
		RK-ID	23	251.47	124.47	248	[LGS17]
		RK-ID	23	243.41	124.41	155.41	this paper
n-n	64	ID	18	57.1	47.52	58.52	[TAY17]
		RK-ID	19	63.03	61.47	56	[LGS17]
		RK-ID	19	62.83	61.30	48.30	this paper
		ZC	14	62	62.58	64	this paper
	128	ID	18	116.94	92.42	115.42	[TAY17]
		RK-ID	19	124.60	122.47	112	[LGS17]
		RK-ID	19	124.43	122.47	97.47	this paper

† : In this attack, 48 bits of the tweakey are considered publicly as tweak. So the upper bound for exhaustive search is 80 bits.

1.3 Outline.

The remainder of this paper is organized as follows. Section 2 provides the required preliminaries, including a brief description of SKINNY. In section 3, related-tweakey impossible differential for different variants of SKINNY are proposed. In section 4, we describe 23-round related-tweakey impossible differential attack on SKINNY-n-2n in details. In section 5, zero-correlation linear characteristics for different variants of SKINNY are proposed and in section 6, the details of 18-round zero-correlation linear cryptanalysis of SKINNY-64-128 is presented. Finally, we conclude the paper in section 7.

2 Preliminaries

In this section, we give a brief description of SKINNY, its round function and key schedule. Then we give a summary of zero-correlation linear cryptanalysis. Finally, the method for using MILP in impossible differential and zero-correlation cryptanalysis is explained.The variables used in this section are introduced in the context.

2.1 A brief description of SKINNY

The lightweight block ciphers of the SKINNY family have 64-bit and 128-bit block versions. In both $n = 64$ and $n = 128$ versions (n is the block size), the internal state is viewed as a 4×4 square array of cells, where each cell can be a nibble (when $n = 64$) or a byte (when $n = 128$). SKINNY is built using the tweakey framework [JNP14] and there are three versions with tweakey sizes $t = n$, $t = 2n$ and $t = 3n$. For simplicity in writing, we show the SKINNY with block size n and tweakey size t with SKINNY-n-t.

Initialization: The cipher takes a plaintext $m = m_0||m_1||\cdots||m_{14}||m_{15}$, while the m_i are s-bit cells (we have $s = 4$ for the 64-bit block SKINNY versions and $s = 8$ for the 128-bit block SKINNY versions). The cipher's internal state is initialized as follows:

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

The Round Function: One encryption round of SKINNY is composed of the following five operations: SubCells(SC), AddConstants(AC), AddRoundTweakey(ART), ShiftRows(SR) and MixColumns(MC) (illustration is in Figure 1(a)).

SubCells: Each cell of the cipher internal state goes through an s-bit S-box. For $s = 4$, this s-box is shown in Table 2.

Table 2: The 4-bit S-box used in SKINNY-64 in hexadecimal form.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_4[x]$	C	6	9	0	1	A	2	B	3	8	5	D	4	E	7	F

AddConstants: In this step the round constants derived from using a 6-bit LFSR are combined with the state.

AddRoundTweakey: The first and the second rows of all tweakey arrays are extracted and bitwise exclusive-ORed to the cipher internal state, respecting the array positioning. Then, the tweakey arrays are updated in 2 steps as shown in Figure 1(b). In the first step, the permutation $P_T = [9, 15, 8, 13, 10, 14, 12, 11, 0, 1, 2, 3, 4, 5, 6, 7]$ is applied on tweakey array. In the second step, every cell of the first and the second rows is individually updated with an LFSR as shown in Table 3.

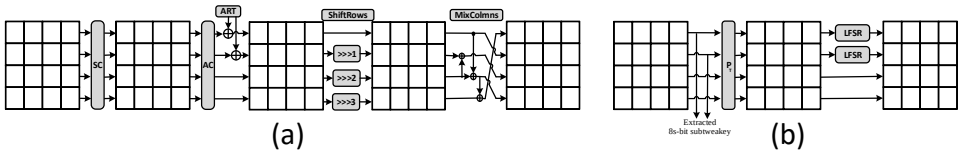

Figure 1: **(a):**SKINNY round function, **(b):**The tweakey schedule of SKINNY

Table 3: The LFSRs used in TK-2 model of SKINNY. The s parameter gives the size of cell in bits.

TK	s	LFSR
TK-2	4	$(x_0\|\|x_1\|\|x_2\|\|x_3) \longrightarrow (x_1\|\|x_2\|\|x_3\|\|x_0 \oplus x_1)$,
	8	$(x_0\|\|x_1\|\|x_2\|\|x_3\|\|x_4\|\|x_5\|\|x_6\|\|x_7) \longrightarrow (x_1\|\|x_2\|\|x_3\|\|x_4\|\|x_5\|\|x_6\|\|x_7\|\|x_0 \oplus x_2)$

Note that, no LFSR is used in TK-1 or single key case. More details about LFSRs in TK-3 model are given in [BJK+16].

ShiftRows: The second, third, and the fourth cell rows are respectively rotated by 1, 2 and 3 positions to the right. This operation can be performed by applying a permutation P on the cells positions of the cipher internal state cell array.

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

MixColumns: Each column of the cipher's internal state array is multiplied by a binary matrix M given below:

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

2.2 Zero-Correlation Linar Cryptanalysis

As described in [SN14], we consider an n-bit block cipher with input variable $x \in \mathbb{F}_2^n$, and f-function $f : \mathbb{F}_2^n \longmapsto \mathbb{F}_2^n$. If we call v and u as the input and output masks, respectively, the linear approximation is defined as follows:

$$x \longmapsto v \cdot x \oplus u \cdot f(x).$$

Its probability can be defined as:

$$p(v; u) = pr(v \cdot x \oplus u \cdot f(x) = 0),$$

and it has the correlation of:

$$C_f(v; u) = 2p(v; u) - 1.$$

We note that the correlation of an approximation will be equal to zero if the probability of approximation is $\frac{1}{2}$.

In zero-correlation linear cryptanalysis, we look for a linear approximation with zero correlation for all keys. There are usually some XORs, F-functions and branches used in each round of any cipher. According to [BR14], there are three rules for these operations:

Lemma 1. ***(XOR operation)*** *Either the three linear selection patterns at an XOR $\oplus$ are equal or the correlation over $\oplus$ is exactly 0.*

Lemma 2. ***(Branching operation)*** *Either the three linear selection patterns at a branching point • sum up to 0 or the correlation over • is exactly 0.*

Lemma 3. ***(Permutation approximation)*** *Over a permutation ϕ, if the input and output selection patterns are neither both zero nor both nonzero, the correlation over ϕ is exactly zero.*

In fact, lemma 1 means that in zero-correlation attack, the inputs and outputs of the XOR operation should be considered equal and if not, the correlation will be zero. Also, in lemma 2, for the branching operation, the input should be equal to the XOR of outputs. Otherwise, the correlation will be zero.

Zero correlation attack

In this subsection, we give a brief explanation about the zero-correlation attack. More details are given in [BW12, SN14, BR14]. Similar to the most of the conventional attacks on block ciphers, zero correlation attack has two stages. In the first stage, the attacker should find a linear approximation with correlation zero for some rounds of the target cipher as a distinguisher. Then in the second stage, he adds some rounds before and after the distinguisher and tries to extract the subkeys of these additional rounds.

In the multidimensional case, there exist m independent linear base approximations such that all of their $l = 2^m - 1$ nonzero linear combinations have correlation zero. As shown in [BLNW12], the statistical value T can be computed to find possible key candidates. In order to compute T, for each $i \in \mathbb{F}_2^m$ the attacker allocates a counter $V[i]$ and initializes it to zero. Then for each distinct plaintext, he computes the corresponding data in $\mathbb{F}_2^m$ and increments the counter $V[i]$ of this value by one. Then the attacker computes the statistical T as follows:

$$T = \sum_{i=0}^{2^m-1} \frac{(V[i] - N2^{-m})^2}{N2^{-m}(1-2^{-m})} = \frac{N2^m}{(1-2^{-m})} \sum_{i=0}^{2^m-1} \left(\frac{V[i]}{N} - \frac{1}{2^m}\right)^2.$$

The statistical T follows a χ^2-distribution with mean and variance of $\mu_0 = l\left(\frac{2^n-N}{2^n-1}\right)$ and $\sigma_0^2 = 2l\left(\frac{2^n-N}{2^n-1}\right)^2$ respectively, for the right key guess while it follows a χ^2-distribution with mean and variance of $\mu_1 = l$ and $\sigma_1^2 = 2l$ for the wrong guess key. With error probability type-I as α and error probability type-II as β, if one considers the decision threshold $t = \mu_0 + \sigma_0 z_{1-\alpha} = \mu_1 - \sigma_1 z_{1-\beta}$, then the amount of required distinct known plaintexts (N) is as follows:

$$N = \frac{2^n(z_{1-\alpha} + z_{1-\beta})}{\sqrt{\frac{l}{2}} - z_{1-\beta}},$$

where $z_p = \Phi^{-1}(p)$ for $0 < p < 1$ and Φ is the cumulative function of the standard normal distribution. The number of required pairs of plaintext-ciphertext depends on the number of linear approximations with correlation zero, block length, and error probabilities type-I and II.

2.3 Using MILP in Impossible differential and Zero-correlation cryptanalysis

In [CJF+16], Cui et al. proposed a method for searching impossible differential characteristic and zero-correlation linear distinguisher based on Mixed-Integer Linear Programming (MILP). In this MILP problem, we can set the objective function to the expression which conveys the differential characteristic's probability and the linear constraint phrase is

configured to form the cryptosystem. Thus, with respect to the set cipher system, we can obtain the optimum probability of differential characteristic forming the cipher system corresponding to the answer to the MILP problem. Although, if we cannot obtain the answer to the MILP problem for a specific input or output differential, it shows that the differential characteristic cannot be formed in the specified cipher system for that input and output differential value. Hence, the input and output differentials will be invalid differential characteristics of the given cipher system. Achieving a case where the answer to the MILP problem cannot be obtained leads to searching for MILP-based impossible differential characteristics.

Recently, Sasaki et al. proposed a new impossible differential search tool from the design and cryptanalysis aspects in [ST17] using MILP. They presented an approach for evaluating s-boxes, including 8×8 s-boxes, in impossible differential cryptanalysis which was missing in [CJF+16]. In this paper, we utilize MILP approach and the results of aforementioned papers to search related-tweakey impossible differential and zero-correlation linear characteristics.

3 Searching Related-tweakey Impossible Differential Characteristics of SKINNY

In this section, we present related-tweakey impossible differential for different variants of SKINNY. Because of the special structure of SKINNY and its performance in key recovery, it is not enough to only search for the longest trails. It means that it is possible to recover more rounds with a 12-round characteristic than a 14-round characteristic. The place of the active bit differences of input, output, and tweakey can affect the final recovered rounds. [ABC+17] can be mentioned as an example. Therefore, we tried to search and list all suitable characteristics in this section. It should be mentioned that we list the notations related to each section in the beginning of that section.

3.1 Related-tweakey Impossible Differential Characteristics of SKINNY in TK1 and TK2 model

The following notations are used in the rest of this subsection (also see Figure 2):

$(input)$:	represents input of the first internal state in the first round of impossible differential characteristic.
S_1 :	represents the internal state after SC in the first round of impossible differential characteristic.
tk_1^1, tk_2^1 :	represents the first round tweakey in TK-1 and TK-2 model, respectively.
$(output)$:	represents output of the last internal state in the last round of impossible differential characteristic.
$\Delta[X]$:	represents a nonzero difference in at least one bit of state X.
$\Delta^i[X]$:	represents a nonzero difference in the i-th cell ($i = 0, \ldots, 15$) of state X.
$\Delta^i_j[X]$:	represents a nonzero difference in the j-th bit of the i-th cell $\big(j = 0, 1, 2, 3$ (or $j = 0, \ldots, 7$) and $i = 0, \ldots, 15\big)$ of state X.
$\Delta^i_{\texttt{0xj}}[X]$:	represents difference of the i-th cell $\big(i = 0, \ldots, 15\big)$ of state X is `0xj`.
"0" :	represents zero difference.
"?" :	represents an unknown difference.

In [BJK+16], the miss-in-the-middle approach was used to find 11-round impossible differential characteristic of SKINNY as $\Delta^{12}(input) \overset{11}{\nrightarrow} (\Delta^8(output)$. Then, they utilized

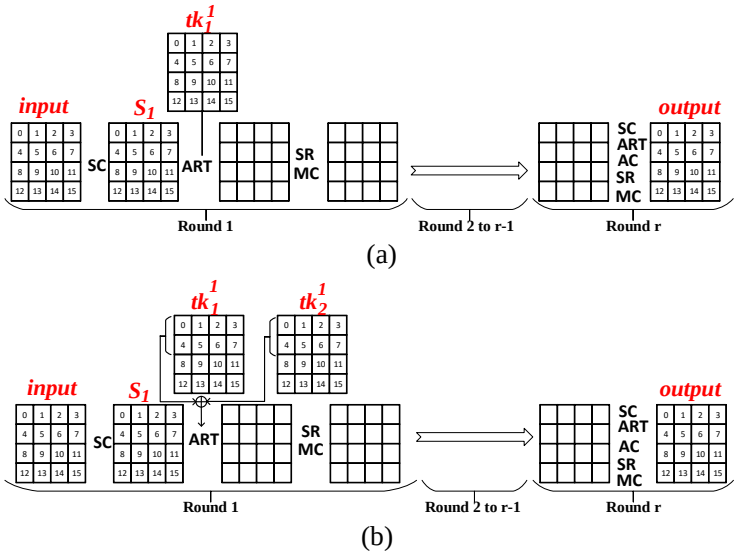

Figure 2: r-round of SKINNY in **(a):** TK1 model **(b):**TK2 model.

it to attack 16-round SKINNY-64-64 (or 128-128) and 18-round SKINNY-64-128. In this paper, in order to find related-tweakey impossible differential characteristics, we use MILP technique to find all related-tweakey impossible differential characteristics based on bit-wise search for SKINNY in TK-1 and TK-2 model. The characteristics in the models are searched considering 1 active bit input or output. However, to reach the best trail in some models due to the structure of trails, we conducted the search under the assumption of having more than 1 bit difference in input or output. Since we can consider the difference in any of *input*, *output*, and tweakey inputs (tk_1^1, tk_2^1), so we have considered the differential models as $\big(\Delta(input), \Delta(tk_1^1), \Delta(output)\big)$ and $\big(\Delta(input), \Delta(tk_1^1), \Delta(tk_2^1), \Delta(output)\big)$, for SKINNY in TK-1 and TK-2 model, respectively. Since in some characteristics, the difference value of *input*, *output*, tk_1^1 or tk_2^1 (in TK-2 model) can be considered zero, we classify the differential trails by the items with zero value. For example, the differential model $\big(\Delta(input), 0, \Delta(tk_2^1), \Delta(output)\big)$ in TK-2 model means that we have only considered the difference in *input*, tk_2^1, and *output* bits and we do not have any difference in tk_1^1. Given that the round-tweakey is combined with internal state after SC, we can consider the difference of the internal state after SC in the first round (S_1) instead of its input in some cases, so we are able to find longer characteristics. A summary of the best-known approximations for SKINNY in both TK-1 and TK-2 model is presented in Table 4. It should be mentioned that we searched the characteristics in case of $s = 4$. However, these characteristics are extendable for $s = 8$ by some slight changes in differences. In addition, in some models, since there are differences in all bits of a cell, we have considered the differential in that cell as truncated.

3.1.1 Searching Related-tweakey Impossible Differential characteristics of SKINNY in TK-1 model.

Differences as $\big(\Delta(input), \Delta(tk_1^1), \Delta(output)\big)$. Considering this case, we found out that the longest related-tweakey impossible differential characteristics reach 12 rounds. We listed all the related-tweakey impossible differential characteristics in Table 5. For example, if we pick $\mathtt{n} = \mathtt{A}$ and choose $(i, j, k) = (12, 8, 8)$, we can derive a 12-round

Table 4: A summary of the known related-tweakey impossible differential characteristics for SKINNY in both TK-1 and TK-2 model.

Cipher	Model Differentials	# Rounds	Ref
SKINNY (In TK-1 model)	$(\Delta(input), \Delta(tk_1^1), \Delta(output))$	12	Table 5
	$(\Delta(S_1), \Delta(tk_1^1), \Delta(output))$	13	Table 7
	$(\Delta(input), \Delta(tk_1^1), 0)$	11	Table 16
	$(\Delta(S_1), \Delta(tk_1^1), 0)$	12	Table 8
	$(0, \Delta(tk_1^1), \Delta(output))$	12	Table 9
	$(0, \Delta(tk_1^1), 0)$	11	Table 20
SKINNY (In TK-2 model)	$(\Delta(input), \Delta(tk_1^1), \Delta(tk_2^1)\Delta(output))$	12	Table 10
	$(\Delta(S_1), \Delta(tk_1^1), \Delta(tk_2^1), \Delta(output))$	14	§ 3.1.2
		15	
	$(\Delta(input), \Delta(tk_1^1), \Delta(tk_2^1), 0)$	12	Table 13
	$(\Delta(input), 0, \Delta(tk_2^1), \Delta(output))$	11	Table 14
	$(\Delta(S_1), 0, \Delta(tk_2^1), \Delta(output))$	12	Table 15
	$(\Delta(input), 0, \Delta(tk_2^1), 0)$	11	Table 16
	$(0, \Delta(tk_1^1), \Delta(tk_2^1), \Delta(output))$	14	Table 17
	$(0, 0, \Delta(tk_2^1), \Delta(output))$	11	Table 18
	$(0, \Delta(tk_1^1), \Delta(tk_2^1), 0)$	13	Table 19
	$(0, 0, \Delta(tk_2^1), 0)$	11	Table 20

Table 5: Related-tweakey impossible differential characteristics $(\Delta^i(input), \Delta^j_{\texttt{0xn}}(tk_1^1), \Delta^l_{\texttt{0xn}}(output))$ for 12-round SKINNY in TK-1 model.

r	n	(i, j, l)
12	any non-zero difference	(1,9,13),(1,10,15),(1,11,11),(2,8,8) (3,9,13),(4,8,8),(4,10,15),(5,9,13) (6,9,13),(6,10,15),(7,10,15),(7,11,11) (12,8,8),(12,9,13),(12,10,15),(12,11,11) (13,9,13),(13,10,15),(14,9,13),(14,10,15) (14,11,11),(15,8,8),(15,9,13),(15,10,15) (15,11,11)

related-tweakey impossible differential characteristic as follows:

$$\underbrace{(0,\ldots,0,\Delta^{12},0,0,0}_{\Delta^{12}(input)};\underbrace{0,\ldots,0,\Delta^8_{\texttt{0xA}},0,0,0,0,0,0,0,0)}_{\Delta^8_{\texttt{0xA}}(tk_1^1)} \overset{12R}{\nrightarrow} \underbrace{(0,\ldots,0,\Delta^8_{\texttt{0xA}},0,0,0,0,0,0,0,0)}_{\Delta^8_{\texttt{0xA}}(output)}.$$

Differences as $(\Delta(S_1), \Delta(tk_1^1), \Delta(output))$. In this model, we consider two cases as follows:

case 1: In this case we noticed that the longest related-tweakey impossible differential characteristic reaches 12 rounds, which is reported in Table 6. For

Table 6: Related-tweakey impossible differential characteristics $(\Delta^i_{\texttt{0xm}}(S_1), \Delta^i_{\texttt{0xm}}(tk_1^1), \Delta^j(output))$ for 12-round SKINNY in TK1 model.

r	m	(i, j)
12	any non-zero difference	(0,8),(1,9),(1,10),(2,8),(2,9),(4,10),(5,8),(5,9),(5,11) (6,9),(6,10),(7,10)

example, Liu et al. [LGS17] used 12-round related-tweakey impossible differen-

tial characteristic $\big(\Delta^1_{\texttt{0xm}}(S_1), \Delta^1_{\texttt{0xm}}(tk^1_1), \Delta^9(output)\big)$, (by choosing $(i,j) = (1,9)$ and considered `0xm` as fixed) and utilized it to attack 19-round SKINNY-n-n.

case 2: We found out that the longest related-tweakey impossible differential characteristics, when there is difference in the input of tk^1_1, input of the internal state after SC at the first round, and also output of the internal state after MC in the last round, reach 13 rounds. We listed these related-tweakey impossible differential characteristics in Table 7.

Table 7: Related-tweakey impossible differential characteristics $\big(\Delta^i_{\texttt{0xm}}(S_1), \Delta^j_{\texttt{0xm}}(tk^1_1), \Delta^l_{\texttt{0xm}}(output)\big)$ for 13-round SKINNY-64-64.

r	m	(i,j,l)
13	any non-zero difference	(0,0,8),(3,3,11),(5,5,9),(7,7,10),(12,3,11),(13,0,8)

As an example, for any non-zero difference `0xm` and by choosing $(i,j,l) = (0,0,8)$, we can obtain a 13-round related-tweakey impossible differential characteristic $\big(\Delta^0_{\texttt{0xm}}(S_1), \Delta^0_{\texttt{0xm}}(tk^1_1), \Delta^8_{\texttt{0xm}}(output)\big)$. The details of this characteristic are depicted in Figure 3. We will consider this characteristic for 19-round attack on SKINNY-n-n in Appendix A.

Differences as $\big(\Delta(S_1), \Delta(tk^1_1), 0\big)$. In this case, we observed that the longest related-tweakey impossible differential characteristic reaches 12 rounds, which is reported in Table 8.

Differences as $\big(0, \Delta(tk^1_1), \Delta(output)\big)$. We list all the related-tweakey impossible differential characteristics for 12-round SKINNY with differences as $\big(0, \Delta(tk^1_1), \Delta(output)\big)$ in Table 9. Sun et al [SGL+17] also obtained these characteristics using CP approach and this table is the same with their results.

3.1.2 Searching Related-tweakey Impossible Differential characteristics of SKINNY in TK-2 model.

In this section, we obtain related-tweakey impossible differential characteristics of SKINNY in the TK-2 model (i.e both tk^1_1 and tk^1_2 are considered).

Differences as $\big(\Delta(input), \Delta(tk^1_1), \Delta(tk^1_2), \Delta(output)\big)$. In this case, the longest related-tweakey impossible differential characteristic obtained consists of 12 rounds. (see Table 10).

As an example, if we consider $(i,j,l) = (12,8,8)$ and $(n,p,q) = (2,2,0)$ we can obtain a 12-round related-tweakey impossible differential characteristic.

Table 8: Related-tweakey impossible differential characteristics $\big(\Delta^i_{\texttt{0xm}}(S_1), \Delta^j_{\texttt{0xm}}(tk^1_1), 0\big)$ for 12-round SKINNY-64-64.

r	m	(i,j)
12	any non-zero difference	(0,0),(1,1),(2,2),(3,3),(4,4),(5,5),(6,6),(7,7) (12,3),(13,0),(14,1)

Table 9: The related-tweakey impossible differential characteristics $\big(0, \Delta^j_{\texttt{0xn}}(tk^1_1), \Delta^l_{\texttt{0xn}}(output)\big)$ for 12-round SKINNY-64-64.

r	n	(j,l)
12	any non-zero difference	(8,8),(13,9),(15,10),(11,11)

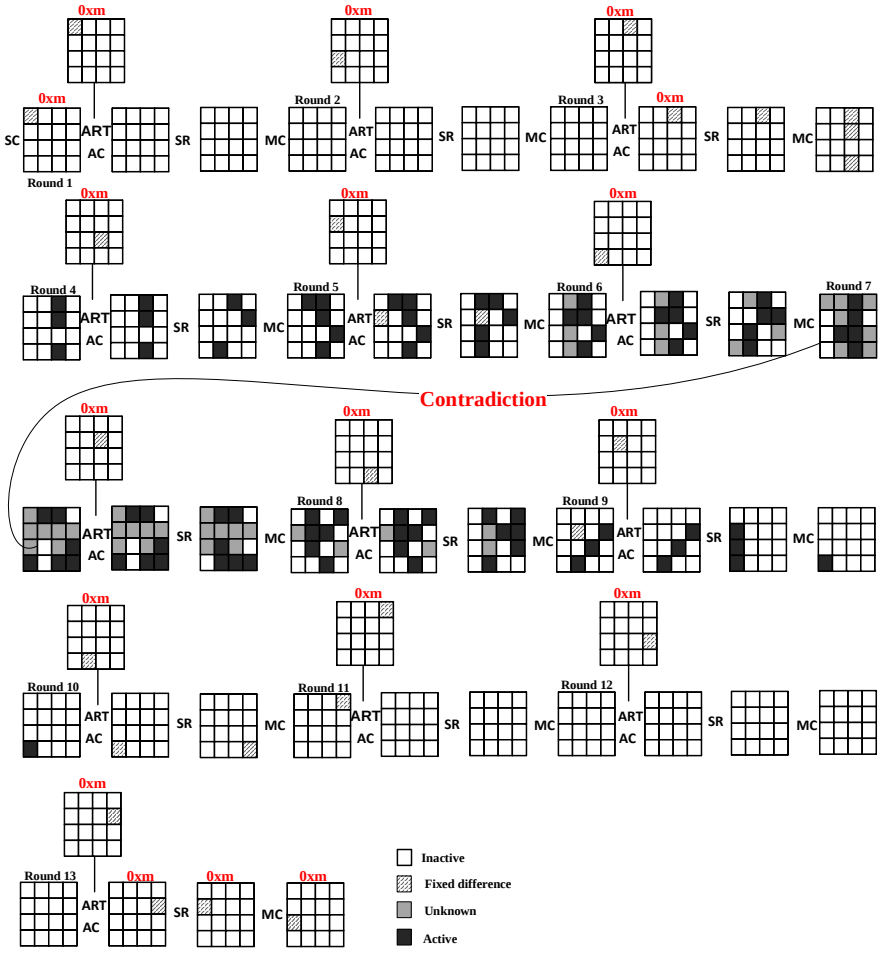

Figure 3: Related-tweakey impossible differential characteristic $\left(\Delta^0_{\mathtt{0xm}}(S_1), \Delta^0_{\mathtt{0xm}}(tk^1_1), \Delta^8_{\mathtt{0xm}}(output)\right)$ for 13-round SKINNY in TK-1 model.

Table 10: Related-tweakey impossible differential characteristics $\big(\Delta^i(input), \Delta^j_n(tk^1_1), \Delta^j_p(tk^1_2), \Delta^l_q(output)\big)$ for 12-round SKINNY-64-128.

r	(i,j,l)	(n,p,q)
12	(1,13,9),(1,15,10),(1,11,11),(2,8,8),(3,13,9) (4,8,8),(4,15,10),(5,13,9),(6,13,9),(6,15,10) (7,15,10),(7,11,11),(12,8,8),(12,11,11),(12,13,9) (12,15,10),(13,13,9),(13,15,10),(14,11,11) (14,13,9),(14,15,10),(15,8,8),(15,13,9) (15,15,10),(15,11,11)	(2,2,0),(2,0,1),(3,1,1) (1,0,2),(1,1,3),(0,2,2)

Differences as $\big(\Delta(S_1), \Delta(tk^1_1), \Delta(tk^1_2), \Delta(output)\big)$. In this difference model, we have considered the input and output differences to include one or more than one active bit, to get better result. Therefore, in this model we consider the following two cases:

case 1: The differential $\big(\Delta^i_{0\mathtt{xm}}(S_1), \Delta^i_{0\mathtt{xn}}(tk^1_1), \Delta^i_{0\mathtt{xp}}(tk^1_2), \Delta^l(output)\big)$ is a 14-round related-tweakey impossible differential characteristic when the following conditions are satisfied:

(1) Choose (i,l) from the sets $\big\{(0,8),(0,9),(1,8),(2,10),(3,10),(4,9),(4,10),(6,8),(6,9),(6,11),(7,9),(7,10)\big\}$.

(2) $\mathtt{m} = \mathtt{n} \oplus \mathtt{p}$.

(3) $LFSR(\mathtt{p}) = \mathtt{n}$.

The possible values of $\mathtt{m}, \mathtt{n}$, and $\mathtt{p}$ that satisfy conditions (2) and (3) are listed in Table 11. This table is constructed for $s = 4$. For $s = 8$, the table can be derived by the same approach.

Table 11: The values of $\mathtt{m}, \mathtt{n}$, and $\mathtt{p}$ for 14-round RK-ID as $\big(\Delta^i_{0\mathtt{xm}}(S_1), \Delta^i_{0\mathtt{xn}}(tk^1_1), \Delta^i_{0\mathtt{xp}}(tk^1_2), \Delta^l(output)\big)$ in TK2 model.

	m	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
⇓	n	E	C	2	8	6	4	A	F	1	3	D	7	9	B	5
	p	F	E	1	C	3	2	D	7	8	9	6	B	4	5	A

For example, Liu et al. [LGS17] used 14-round related-tweakey impossible differential characteristic $\big(\Delta^2_{0\mathtt{xm}}(S_1), \Delta^2_{0\mathtt{xn}}(tk^1_1), \Delta^2_{0\mathtt{xp}}(tk^1_2), \Delta^{10}(output)\big)$, (by choosing $(i,l) = (2,10)$ and considering one of $(\mathtt{m},\mathtt{n},\mathtt{p})$ as fixed) and utilized it to attack 23-round SKINNY-n-2n.

case 2: The differential $\big(\Delta^i_{0\mathtt{xm}}(S_1), \Delta^i_{0\mathtt{xn}}(tk^1_1), \Delta^i_{0\mathtt{xp}}(tk^1_2), \Delta^l_{0\mathtt{xq}}(output)\big)$ is a 15-round related-tweakey impossible differential characteristic when the following conditions are satisfied:

(1) Choose (i,l) from the sets $\big\{(1,8),(3,10),(5,11),(6,9)\big\}$.

(2) $\mathtt{m} = \mathtt{n} \oplus \mathtt{p}$.

(3) $LFSR(\mathtt{p}) = \mathtt{n}$.

(4) $\mathtt{n} \oplus LFSR^7(\mathtt{p}) = \mathtt{q}$.

For $s = 4$, the possible values of $\mathtt{m}, \mathtt{n}, \mathtt{p}$, and $\mathtt{q}$ that satisfy conditions (2), (3), and (4) are listed in Table 12. For $s = 8$ the table can be derived by the same approach.

Table 12: The values of $\mathtt{m}, \mathtt{n}, \mathtt{p}$, and $\mathtt{q}$ for 15-round RK-ID as $\left(\Delta^i_{0\mathtt{xm}}(S_1), \Delta^i_{0\mathtt{xn}}(tk_1^1), \Delta^i_{0\mathtt{xp}}(tk_2^1), \Delta^l_{0\mathtt{xq}}(output)\right)$ in TK2 model.

	m	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
⇓	n	E	C	2	8	6	4	A	F	1	3	D	7	9	B	5
	p	F	E	1	C	3	2	D	7	8	9	6	B	4	5	A
	q	7	F	8	E	9	1	6	B	C	4	3	5	2	A	D

As an example, if we consider $(i, l) = (1, 8)$ and $(\mathtt{m}, \mathtt{n}, \mathtt{p}, \mathtt{q})$ as one of the columns of Table 12, we can obtain the 15-round related-tweakey impossible differential characteristic $\left(\Delta^1_{0\mathtt{xm}}(S_1), \Delta^1_{0\mathtt{xn}}(tk_1^1), \Delta^1_{0\mathtt{xp}}(tk_2^1), \Delta^8_{0\mathtt{xq}}(output)\right)$ which is considered for 23-round attack on SKINNY-n-2n in section 4.3. The details are depicted in Figure 4.

Differences as $(\Delta(input), \Delta(tk_1^1), \Delta(tk_2^1), 0)$**.** In this case, we found that the longest related-tweakey impossible differential characteristics reach 12 rounds, which are reported in Table 13.

Table 13: Related-tweakey impossible differential characteristics $\left(\Delta^i(input), \Delta^j_{0\mathtt{xn}}(tk_1^1), \Delta^j_{0\mathtt{xn}}(tk_2^1), 0\right)$ for 12-round SKINNY.

r	n	(i, j)
12	any non-zero difference	(12,5),(12,6),(13,1),(13,4),(13,6),(14,3),(14,4),(14,5) (0,15),(7,14)

Differences as $(\Delta(input), 0, \Delta(tk_2^1), \Delta(output))$**.** In this case, we realized that the longest related-tweakey impossible differential characteristics reach 11 rounds, which are reported in Table 14.

Table 14: Truncated Related-tweakey impossible differential characteristics $(\Delta^i(input), 0, \Delta^k(tk_2^1), \Delta^l(output))$ for 11-round SKINNY.

r	(i, k, l)
11	(12,8,13),(12,9,13),(12,11,13),(13,9,9),(13,9,14),(13,10,9) (14,10,12),(14,10,15),(15,8,8)

Differences as $(\Delta(S_1), 0, \Delta(tk_2^1), \Delta(output))$**.** In this case, the longest related-tweakey impossible differential characteristics will reach 12 rounds, as listed in Table 15.

Table 15: Truncated Related-tweakey impossible differential characteristics $\left(\Delta^i_{0\mathtt{xm}}(S_1), 0, \Delta^i_{0\mathtt{xm}}(tk_2^1), \Delta^l(output)\right)$ for 12-round SKINNY.

r	m	(i, l)
12	any non-zero difference	(0,8),(1,9),(1,10),(2,8),(2,9),(4,10),(5,8),(5,9) (5,11),(6,9),(6,10),(7,10)

Differences as $(\Delta(input), \Delta(tk_1^1), 0)$ **and** $(\Delta(input), 0, \Delta(tk_2^1), 0)$**.** We considered the differences as $(\Delta(input), \Delta(tk_1^1), 0)$ in TK-1 model or $(\Delta(input), 0, \Delta(tk_2^1), 0)$ in TK-2 model and noticed that the longest related-tweakey impossible differential characteristics reach 11 rounds. These characteristics are listed in Table 16.

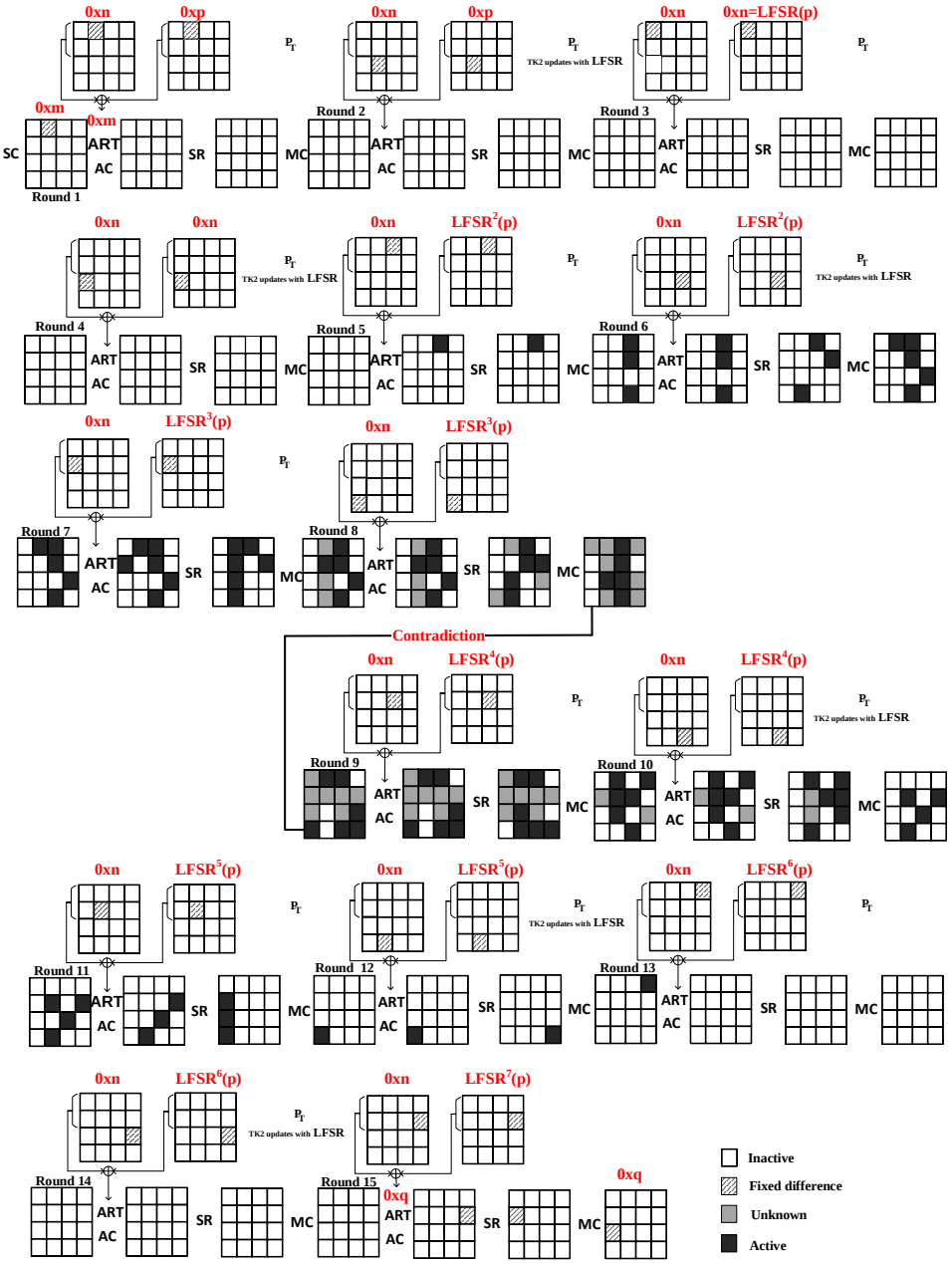

Figure 4: Related-tweakey impossible differential characteristic $\left(\Delta^1_{\texttt{0xm}}(S_1), \Delta^1_{\texttt{0xn}}(tk^1_1), \Delta^1_{\texttt{0xp}}(tk^1_2), \Delta^8_{\texttt{0xq}}(output)\right)$ for 15-round SKINNY in TK-2 model.

Table 16: Truncated Related-tweakey impossible differential characteristics $(\Delta^i(input), \Delta^j(tk_1^1), 0)$(in TK-1 model) or $(\Delta^i(input), 0, \Delta^j(tk_2^1), 0)$ for 11-round SKINNY.

r	(i, j)
11	(0,9),(0,12),(0,14),(1,11),(1,12),(1,13),(1,15),(2,8),(3,13) (3,14),(4,8),(4,12),(4,15),(5,9),(5,13),(5,14),(6,9),(6,12) (6,13),(6,14),(6,15),(7,9),(7,11),(7,12),(7,14),(7,15),(12,8) (12,9),(12,10),(12,11),(12,12),(12,13),(12,14),(12,15),(13,9) (13,10),(13,12),(13,13),(13,14),(13,15),(14,11),(14,12),(14,13) (14,14),(14,15),(15,8),(15,9),(15,10),(15,11),(15,13),(15,14) (15,15)

Table 17: Related-tweakey impossible differential characteristics $(0, \Delta_n^j(tk_1^1), \Delta_p^j(tk_2^1), \Delta_q^l(output))$ for 14-round SKINNY in TK-2 model.

r	l	q	(j, n, p)
14	8	0	(9,2,3)
		3	(9,1,2)
	9	0	(14,2,3)
		3	(14,1,2)
	10	0	(11,2,3)
		3	(11,1,2)
	11	0	(13,2,3)
		3	(13,1,2)

Table 18: Related-tweakey impossible differential characteristics $(0, 0, \Delta(tk_2^1), \Delta(output))$ for 11-round SKINNY in TK-2 model.

r	$\big(0, 0, \Delta^k(tk_2^1), \Delta^l(output)\big)$
11	(k, l)
	(8,8),(10,8),(13,8),(9,9),(10,9),(13,9),(14,9),(9,10) (12,10),(14,10),(15,10),(13,11)
	$\big(0, 0, \Delta_p^k(tk_2^1), \Delta^l(output)\big)$
	(p, k, l)
	(0,11,10),(0,15,11)

Differences as $(0, \Delta(tk_1^1), \Delta(tk_2^1), \Delta(output))$. When we consider the differences as $(0, \Delta(tk_1^1), \Delta(tk_2^1), \Delta(output))$, the longest related-tweakey impossible differential characteristics reach 14 rounds, which are listed in Table 17.

As an example, if we consider $\big(l = 8, q = 0, (j, n, p) = (9, 2, 3)\big)$, we can obtain 14-round related-tweakey impossible differential characteristic as $\big(0, \Delta_2^9(tk_1^1), \Delta_3^9(tk_2^1), \Delta_0^8(output)\big)$.

Differences as $(0, 0, \Delta(tk_2^1), \Delta(output))$. In this case, the longest related-tweakey impossible differential characteristics reaches 11 rounds, as reported in Table 18.

Differences as $(0, \Delta(tk_1^1), \Delta(tk_2^1), 0)$. In this case, we only considered difference in tweakey (both tk_1^1 and tk_2^1) and obtained 13-round related-tweakey impossible differential characteristics, listed in Table 19. Also, the characteristics for 11 and 12 rounds are listed in this table.

For example, Ankele et al. [ABC+17] used 11-round related-tweakey impossible differential characteristic $\big(0, \Delta^8(tk_1^1), \Delta^8(tk_2^1), 0\big)$ and utilized it to attack 23-round SKINNY-64-128.

Table 19: Related-tweakey impossible differential characteristics $\big(0, \Delta(tk_1^1), \Delta(tk_2^1), 0\big)$ for 11, 12 and 13-round SKINNY in TK-2 model.

$\big(0, \Delta^j(tk_1^1), \Delta^k(tk_2^1), 0\big)$	
r	(j,k)
11	(8,8),(9,9),(9,13),(10,10),(11,11),(12,12) (14,12),(12,14),(13,9),(13,13),(13,14) (14,13),(14,14),(14,15),(15,14),(15,15)

$\big(0, \Delta_n^j(tk_1^1), \Delta_p^j(tk_2^1), 0\big)$		
r	j	(n,p)
12	$0, \cdots, 7$	(0,0),(1,1),(2,2),(3,3)
13	$8, \cdots, 15$	(1,2),(2,3),(3,0)

Table 20: Related-tweakey impossible differential characteristics $\big(0, \Delta^j(tk_1^1), 0\big)$ (in TK-1 model) or $\big(0, 0, \Delta^j(tk_2^1), 0\big)$ for 10 and 11-round SKINNY.

r	j
10	0,$\cdots$,15
11	8,$\cdots$,15

Differences as $(0, \Delta(tk_1^1), 0)$ or $(0, 0, \Delta(tk_2^1), 0)$. We considered the differences as $(0, \Delta(tk_1^1), 0)$ in TK-1 model or $(0, 0, \Delta(tk_2^1), 0)$ in TK-2 model and found that the longest related-tweakey impossible differential characteristics reach 11 rounds. These characteristics for 10 and 11 rounds are listed in Table 20.

4 Related Tweakey Impossible Differential Attack on SKINNY

4.1 Notations

The following notations are used in the rest of paper:

P :	represents plaintext.
C :	represents ciphertext.
tk_1^i, tk_2^i :	represents the i-th round tweakey in TK-1 and TK-2 model, respectively.
TKi :	represents the i-th round tweakey. This is equal to the result of exclusive-ORing the first and the second rows of tk_1^i and tk_2^i and $TKi[j]$ represents the j-th cell $(0 \leq j \leq 15)$ of TKi.
X_i :	represents the internal state before SC in round i and $X_i[j]$ represents the j-th cell $(0 \leq j \leq 15)$ of X_i.
Y_i :	represents the internal state before ART in round i and $Y_i[j]$ represents the j-th cell $(0 \leq j \leq 15)$ of Y_i.
Z_i :	represents the internal state before SR in round i and $Z_i[j]$ represents the j-th cell $(0 \leq j \leq 15)$ of Z_i.
W_i :	represents the internal state before MC in round i and $W_i[j]$ represents the j-th cell $(0 \leq j \leq 15)$ of W_i.
col(i):	represents the column $i (1 \leq i \leq 4)$.
$\overline{X}$:	represents the corresponding variable under the related tweakey difference encryption.
$\Delta X_i, \Delta X_i[j]$:	represents the difference at state X_i and cell $X_i[j]$, respectively.

4.2 An Overview of Impossible Differential Cryptanalysis

We start with recalling the framework introduced by Boura et al in [BNPS14]. In this method, the cipher is split to three parts: $E = E_3 \circ E_2 \circ E_1$ in which E_2 consists of an impossible differential $\Delta X \nrightarrow \Delta Y$ and by propagating ΔX and ΔY through E_1^{-1} and E_3 respectively, we obtain Δ_{in} and Δ_{out} with probability 1. Therefore, we can verify the differential $\Delta X \leftarrow \Delta_{in}$ and $\Delta Y \leftarrow \Delta_{out}$ with probability $\frac{1}{2^{c_{in}}}$ and $\frac{1}{2^{c_{out}}}$ respectively. Notice that c_{in} and c_{out} are defined as the number of bit-conditions needed to be verified to obtain ΔX from Δ_{in} and ΔY from Δ_{out}, respectively. We consider k_{in} and k_{out} as the key information involved in E_1 and E_3, respectively.

For a given pair of inputs, the probability of having a difference ΔX and an output difference ΔY under a random key guess is $2^{-(c_{in}+c_{out})}$. The probability for a trial key to be placed in the set of possible keys should be small enough so that the number of pairs N can be chosen appropriately. This probability is calculated as:

$$P = (1 - 2^{-(c_{in}+c_{out})})^N \simeq e^{-N \times 2^{-(c_{in}+c_{out})}}.$$

By adopting the strategy presented in [BNPS14], we consider the smallest value of pairs such that $e^{-N \times 2^{-(c_{in}+c_{out})}} < \frac{1}{2}$, to reduce the exhaustive search by at least one bit.

Now, we need to find N pairs which verify a given differential. From [BNPS14], using the limited birthday problem, the cost of obtaining the N pairs (C_N) is:

$$\max \left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N 2^{n+1-|\Delta|}} \right\}, N 2^{n+1-|\Delta_{in}|-|\Delta_{out}|} \right\}, \tag{1}$$

verifying that $C_N < 2^n$, where n is the size of the block cipher. By considering the cost of one encryption as C_E, the time complexity C_T is given by the following equation:

$$C_T = \left(C_N + \left(N + 2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C_E' + 2^{|K|} P \right) C_E, \tag{2}$$

where C_E' is the ratio of the cost of partial encryption to the full encryption and $2^{|K|}P$ is the cost of the exhaustive search for the key K after the impossible differential attack. It should be noted that in [BNPS14], a generic complexity analysis of impossible differential attacks against block ciphers was presented. Afterwards, in [Der16], Derbez showed that the results of the paper [BNPS14] may be incorrect and sometimes can produce a miscalculation in time complexity. In fact, it is because of the structure of the key schedule which has a non-negligible impact on the time complexity of such attacks and it has to be added to the time complexity C_T. Boura et al. did not consider this case in their investigations. Recently, Boura et al. in [BLNPS18], introduced techniques which complete and improve the method and the given analysis in [BNPS14]. Based on this new paper, the part of the key schedule which connects the sub-keys of the first rounds to the sub-keys of the last rounds can be seen as a black box and the computation above should be taken into account in the estimation of time complexity. The details of this technique can be seen in [BLNPS18]. Note that the formula provided for time complexity in [BNPS14, BLNPS18] is just a lower-bound approximation of the time complexity and for an exact determination of the complexity, one must perform the detailed attack step by step. Therefore, in this paper, we performed the detailed attack step by step to compute the time complexity.

To describe our related-tweakey impossible differential attack on the SKINNY-n-2n and SKINNY-n-n, first, we should introduce the following lemma [ABC+17, LGS17]:

Lemma 4. *The equation $S(x + \Delta_i) + S(x) = \Delta_0$ has one solution x on average for $\Delta_i, \Delta_0 \neq 0$. Similar result holds for the inverse S-Box, S^{-1}.*

Using this lemma, we are going to present the 23-round related-tweakey impossible differential attack on SKINNY-n-2n in the following section.

There are some slight differences between different variants of SKINNY. SKINNY-64-64 and SKINNY-128-128 just differ in the cell size. SKINNY-64-128 and SKINNY-128-256 differ in cell size and the LFSR operation of the key schedule. Since our attacks are based on the same 15-round distinguisher for SKINNY-64-128 and SKINNY-128-256 and the same 13-round distinguisher for SKINNY-64-64 and SKINNY-128-128, we present the details of attacks as a function of the cell size s, where $s = 4$ and $s = 8$ in case of SKINNY-64 and SKINNY-128 respectively. The attack on the 19-round SKINNY-n-n work in a similar manner and is presented in Appendix A.

4.3 23-round Related Tweakey Impossible Differential Attack on SKINNY-n-2n

In this section, the details of our 23-round attack on SKINNY-n-2n will be presented utilizing related-tweakey impossible differential cryptanalysis. We use the 15-round related-tweakey impossible differential trail, which is shown in Figure 4, and extend it by 3 and 5 rounds in backward and forward directions respectively (see Figure 5). In this attack, instead of the tweakey $TK1$, we can obtain the equivalent tweakey ETK by using $ETK = MC(SR(TK1))$ in the first round, so we can start our tweakey recovery attack at Y_1; given that there is no tweakey used before Y_1. The plaintext P can be recovered by applying MC^{-1}, SR^{-1}, AC^{-1}, and SC^{-1} layers on Y_1. In the following section, we first describe the overall strategy of attack and then go through details.

Overall Strategy In this section, we explain the overall strategy of the attack when $s = 4$ based on Figure 5. For $s = 8$, the attack can be followed by the same approach. Figure 5 shows that in the state cells, what kind of information (just difference or just value or both difference and value) is needed to verify the differential path from $\Delta X_{19} \rightarrow \Delta C$ and $\Delta Y_4 \rightarrow \Delta P$. As an example, during the key recovery phase in rounds 19 to 23, those key guesses for which the given ciphertext pair follows the differential trail from $\Delta X_{19} \rightarrow \Delta C$ (shown by gray cells) are collected. We can do this by checking if ΔX_{i+1} will lead to the required difference ΔW_i or not for $19 \leq i \leq 23$ in each round. Starting the procedure from ΔC, to calculate ΔX_i in each round i, it is required to know the difference and the values of state in the active cells of the corresponding ΔY_i's. To compute the required state values of Y_i's in each round i, knowledge of the state values of cells (that might not be active differentially) and also the key values in the next rounds (round $i + 1$ till round 23) are required, on which the Y_i's are dependent.

As shown in Figure 5, the values of constant differentials $TK2[7]$, $TK4[1]$, $TK18[7]$, $TK20[1]$, and $TK22[0]$, in rounds 2, 4, 18, 20, and 22 respectively, affect the 23-round attack. In addition, these differentials are dependent on each other and by choosing any of input differentials of the 15-round impossible characteristic, the others can be defined. These differentials are shown in a table which we call it Tweakey Differentials Table (TDT) (see Table 21).

For example, from the first column of TDT, if we consider the value of difference in $TK4[1]$ equal to `0x1` ($\Delta TK4[1] =$ `0x1`), the value of difference in $TK18[7]$ must be `0x7` to construct a 15-round related-tweakey impossible differential trail (see Table 12). By choosing these differences, the other differences in $TK2[7]$, $TK20[1]$, and $TK22[0]$ should be `0x9`, `0xD`, and `0x8`, respectively. It should be mentioned that all the 255 differences in case of $s = 8$ can be calculated and form the TDT by the same approach.

In this 23-round attack, instead of using just one characteristic (one column of TDT), we use all of these characteristics (all columns of TDT). The general procedure of this attack is to use 15 lists $L_i (i = 1, \cdots, 15)$ for storing pairs. In fact, during the

Table 21: The TDT table.

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}	L_{14}	L_{15}
TDT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$TK2[7]$ 1	9	3	A	6	F	5	C	4	D	7	E	2	B	1	8
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$TK4[1]$ 2	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
$TK18[7]$ 3	7	F	8	E	9	1	6	B	C	4	3	5	2	A	D
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
$TK20[1]$ 4	D	A	7	5	8	F	2	6	B	C	1	3	E	9	4
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
$TK22[0]$ 5	8	1	9	2	A	3	B	C	4	D	5	E	6	F	7

attack procedure, the data related to column i of TDT is saved in L_i list and the attack will be continued based on this list. Using this technique, the adversary will be able to remove more wrong keys than the case of using just one trail with the same initial data. For this purpose, for each pair of plaintext and ciphertext, first, the adversary guesses the value of $7th$ cell of $ETK(ETK[7])$ in the first round to calculate the value of difference $\Delta Y_2[7]$ in the second round. Here, we study two cases. First, when the adversary uses one of the impossible differential characteristics and second when the adversary uses all characteristics.

Case a. When the adversary uses one of the impossible differential characteristics (one column of TDT):

In this case, the adversary should check the equality of $\Delta Y_2[7] = \Delta TK2[7]$ for each pair after calculating $\Delta Y_2[7]$ based on the guessed $ETK[7]$. This will lead to an s-bit filter on the remained pairs. Also, by knowing the value of difference in Y_2, the probability of having those differences in Z_3 will be $p = 2^{-s}$.

Case b. When the adversary uses all impossible differential characteristics (all columns of TDT is considered in the attack):

In this case, after guessing $ETK[7]$ and calculating $\Delta Y_2[7]$ for each pair, the adversary chooses i index based on the first row of TDT such that:

$$TDT[1][i] = \Delta Y_2[7]$$

Then stores this pair on list L_i ($TDT[m][n]$ means the mth row of nth column of TDT). For example, if $\Delta Y_2[7] = \texttt{0xA}$, the corresponding pair will be saved on list L_3 and the same approach applies for storing the remained pairs. Obviously, each pair will be stored in one of the lists and there is no need to filtering in this step. Also, by knowing the value of differences in Y_2, the probability of having the differences in Z_3 will be $p = 1$. Then the adversary can complete the attack based on these lists for each pair. As an example, consider the adversary calculates the value of $\Delta Y_4[1]$ (i.e., the input of impossible differential characteristic) by guessing the related keys in the first rounds for the pairs in ith list (L_i). For each pair the adversary checks if:

$$\Delta Y_4[1] = TDT[2][i].$$

If the equation is not correct, the corresponding pair will be omitted from the list. The same process will be applied for the other lists. Therefore, this step results in a total of an s-bit filter on the remained pairs. This procedure should be continued for the other rounds to determine the value of difference $\Delta X_{19}[8]$ and remove the wrong keys.

In this paper, the 23-round cryptanalysis is described in details based on the second case in the following section:

Data Collection The adversary should construct 2^x structures at Y_1 and consider all the possible values in 4 cells $Y_1[5,7,8,15]$ for each structure, while the remaining cells take a fixed value. By using $2^{x+|\Delta_{in}|} = 2^{x+4s}$ messages, we can generate $2^{x+2|\Delta_{in}|} = 2^{x+8s}$ pairs of messages $(P, \overline{P})$, then ask the encryption oracle to obtain the corresponding ciphertexts $(C, \overline{C})$. Then for each ciphertext pair, we check whether $n - |\Delta_{out}|$ bits are zero or not and discard it if false. Note that in our 23-round attack on SKINNY-n-2n this step is skipped as $n = |\Delta_{out}|$ and in our 19-round attack on SKINNY-n-n, this step is not skipped. The expected number of the remaining pairs is approximately $N = 2^{x+2|\Delta_{in}|-(n-|\Delta_{uot}|)} = 2^{x+8c}$ plaintext pairs. This step requires a total of $2^{x+|\Delta_{in}|+1} = 2^{x+8c+1}$ encryption calls.

Tweakey Recovery For each of the N pairs

1. Guess $ETK[7]$ and compute $Y_2[7]$ and then by using Table 21 (TDT), determine i index such $Y_2[7] = TDT[1][i]$ and store the pair in the list L_i and repeat this for the other pairs. Obviously, each pair will be saved in one list and there is no need to filtering in this step. The time complexity of this step is $N.2^s$ and the number of tests left for the next step is $N.2^s$.
2. Satisfying the round 23, by applying the following steps on all N message pairs on all lists, leads to the determination of the number of possible values of $TK23[0:7]$:
   (a) From the knowledge of the value of ciphertext pair, we can compute $\Delta X_{23}[8,12]$, since there is no need to have any tweakey information to compute these cells. Due to the MC operation on the active cells of col(1) of W_{22}, we have $\Delta X_{23}[4] = \Delta X_{23}[8] \oplus \Delta X_{23}[12]$. Given that:
   $$S^{-1}(Y_{23}[4]) \oplus S^{-1}(\underbrace{Y_{23}[4] \oplus \Delta Y_{23}[4]}_{\overline{Y}_{23}[4]}) = \Delta X_{23}[4],$$
   the knowledge of the $\Delta X_{23}[4]$ and $\Delta Y_{23}[4]$ allows the attacker to calculate $Y_{23}[4] = Z_{23}[4] \oplus TK23[4]$ as a solution of the above equation by using lemma 4. Now, the attacker can determine $TK23[4]$.
   (b) We can compute $\Delta X_{23}[14]$ from the knowledge of $Z_{23}[14]$ and $\Delta Z_{23}[14]$. Based on the properties of MC operation on col(3) of W_{22}, the equation $\Delta X_{23}[2] = \Delta X_{23}[6] = \Delta X_{23}[14]$ helps us to know the difference values of $\Delta X_{23}[2], \Delta X_{23}[6]$. Since
   $$S^{-1}(Y_{23}[2]) \oplus S^{-1}(Y_{23}[2] \oplus \Delta Y_{23}[2]) = \Delta X_{23}[2],$$
   $$S^{-1}(Y_{23}[6]) \oplus S^{-1}(Y_{23}[6] \oplus \Delta Y_{23}[6]) = \Delta X_{23}[6],$$
   by knowing the difference values of $\Delta Y_{23}[2], \Delta Y_{23}[6], \Delta X_{23}[2]$ and $\Delta X_{23}[6]$, the lemma 4 guarantees one solution on average and we can obtain $Y_{23}[2,6]$ and thus $TK23[2,6]$ (due to $Y_{23}[2] = z_{23}[2] \oplus TK23[2]$ and $Y_{23}[6] = z_{23}[6] \oplus TK23[6]$).
   (c) We can compute $\Delta X_{23}[15]$ from the knowledge of $Z_{23}[15]$ and $\Delta Z_{23}[15]$. Based on the properties of MC operation on col(4) of W_{22}, $\Delta X_{23}[7]$ will be simply determined using $\Delta X_{23}[7] = \Delta X_{23}[15]$. Since
   $$S^{-1}(Y_{23}[7]) \oplus S^{-1}(Y_{23}[7] \oplus \Delta Y_{23}[7]) = \Delta X_{23}[7],$$
   $Y_{23}[7]$ can be derived by using lemma 4 and the knowledge of the value of $Z_{23}[7]$ helps us to determine $TK23[7]$.

(d) Guess $TK23[0,1,3,5]$. Hence, we can compute Z_{22} and ΔZ_{22} as shown in Figure 5.

At this step, the attacker can uniquely determine $tk_1^1[3]$ and $tk_2^1[3]$ from the knowledge of $ETK[7]$ and $TK23[0]$. This helps her to determine $ETK[15]$ and $TK21[1]$. The time complexity of this step is $N.2^{5s}$ and the number of tests left for the next step is $N.2^{5s}$.

3. Satisfying the round 22, by applying the following steps on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK22[2,3,6,7]$:
   (a) From the knowledge of $Z_{22}[14]$ and $\Delta Z_{22}[14]$, we can compute $\Delta X_{22}[14]$. Based on the properties of MC operation on col(3) of W_{21}, we have $\Delta X_{22}[2] = \Delta X_{22}[14]$. Hence, we can determine $\Delta X_{22}[2]$. Since
   $$S^{-1}(Y_{22}[2]) \oplus S^{-1}(Y_{22}[2] \oplus \Delta Y_{22}[2]) = \Delta X_{22}[2],$$
   by using lemma 4, $Y_{22}[2]$ can be determined. Now, since $Y_{22}[2] = Z_{22}[2] \oplus TK22[2]$, $TK22[2]$ can be determined
   (b) Guess $TK22[3,6,7]$. Hence, we can compute col(3) and col(4) of W_{21} and ΔW_{21} as shown in Figure 5. From this information, we can compute $\Delta X_{21}[8,12]$. Based on the properties of MC operation on col(1) of W_{20}, we have $\Delta X_{21}[8] = \Delta X_{21}[12]$. Checking if $\Delta X_{21}[8]$ and the active cell $\Delta X_{21}[12]$ are equal will lead to an s-bit filter on the remaining tweakeys.

   The time complexity of this step is $N.2^{8s}$ and the number of tests left for the next step is $N.2^{7s}$.

4. Satisfying the round 22, by applying the following steps on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK22[0,1,4,5]$:
   (a) From the knowledge of $Z_{22}[12]$ and $\Delta Z_{22}[12]$ we can compute $\Delta X_{22}[12]$. Based on the properties of MC operation on col(1) of W_{21}, we have $\Delta X_{22}[4] = \Delta X_{22}[12]$ and thus $\Delta X_{22}[4]$ will be simply determined. Since
   $$S^{-1}(Y_{22}[4]) \oplus S^{-1}(Y_{22}[4] \oplus \Delta Y_{22}[4]) = \Delta X_{22}[4],$$
   and we have $Y_{22}[4] = Z_{22}[4] \oplus TK22[4]$, the value of $TK22[4]$ can be determined based on lemma 4.
   (b) From the knowledge of $Z_{22}[13]$ and $\Delta Z_{22}[13]$, we can compute $\Delta X_{22}[13]$. Based on the properties of MC operation on col(2) of W_{21}, we have $\Delta X_{22}[5] = \Delta X_{22}[13]$. Since
   $$S^{-1}(Y_{22}[5]) \oplus S^{-1}(Y_{22}[5] \oplus \Delta Y_{22}[5]) = \Delta X_{22}[5],$$
   and we have $Y_{22}[5] = Z_{22}[5] \oplus TK22[5]$, by using lemma 4, $TK22[5]$ can be determined.
   (c) Guess $TK22[0,1]$. For each pair on list L_i, $\Delta Y_{22}[0]$ can be determined. In fact, $\Delta Y_{22}[0] = \Delta Z_{22}[0] \oplus \Delta TK22[0]$ and $\Delta TK22[0]$ can be determined from TDT as $\Delta TK22[0] = TDT[5][i]$. Compute Z_{21} and ΔZ_{21} as shown in Figure 5.

   The time complexity of this step is $N.2^{9s}$ and the number of tests left for the next step is $N.2^{9s}$.

5. Satisfying the round 21, by applying the following steps on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK21[0,2,5,6]$:

(a) From the knowledge of $Z_{21}[12]$ and $\Delta Z_{21}[12]$ we can compute $\Delta X_{21}[12]$. Based on the properties of MC operation on col(1) of W_{20}, we have $\Delta X_{21}[0] = \Delta X_{21}[12]$. Since $Y_{21}[0] = Z_{21}[0] \oplus TK21[0]$ and we have

$$S^{-1}(Y_{21}[0]) \oplus S^{-1}(Y_{21}[0] \oplus \Delta Y_{21}[0]) = \Delta X_{21}[0],$$

thus, by using lemma 4, $TK21[0]$ can be determined.

(b) We can compute $\Delta X_{21}[13]$ from the knowledge of $Z_{21}[13]$ and $\Delta Z_{21}[13]$. Based on the properties of MC operation on col(2) of W_{20}, we have $\Delta X_{21}[13] = \Delta X_{21}[5]$. Since $Z_{21}[5] = Y_{21}[5] \oplus TK21[5]$, we have

$$S^{-1}(Y_{21}[5]) \oplus S^{-1}(Y_{21}[5] \oplus \Delta Y_{21}[5]) = \Delta X_{21}[5],$$

hence, by using lemma 4, $TK21[5]$ can be determined.

(c) Based on the properties of MC operation on col(3) of W_{20}, we have $\Delta X_{21}[2] = \Delta X_{21}[6] = \Delta X_{21}[14]$. Since $Y_{21}[2] = Z_{21}[2] \oplus TK21[2]$ and $Y_{21}[6] = Z_{21}[6] \oplus TK21[6]$, we have

$$S^{-1}(Y_{21}[2]) \oplus S^{-1}(Y_{21}[2] \oplus \Delta Y_{21}[2]) = \Delta X_{21}[2],$$

$$S^{-1}(Y_{21}[6]) \oplus S^{-1}(Y_{21}[6] \oplus \Delta Y_{21}[6]) = \Delta X_{21}[6],$$

hence, by using lemma 4, we can determine $TK21[2, 6]$. For each list L_i, checking if $\Delta X_{21}[5] = TDT[4][i]$ will generally lead to an s-bit filter on all L_i lists.

At this step, the attacker can uniquely determine $tk_1^1[1, 2, 4, 7]$ and $tk_2^1[1, 2, 4, 7]$ from the knowledge of $TK23[2, 3, 4, 5]$ and $TK21[0, 2, 5, 6]$. This helps her to determine $ETK[1, 2, 5, 6, 8, 9, 14]$ and $TK3[1]$. The time complexity of this step is $N.2^{9s}$ and the number of tests left for the next step is $N.2^{8s}$.

6. Since $ETK[5, 8, 15]$ is known from the previous steps, we can satisfy the round 1 by applying the following steps on all N message pairs on all lists for the remaining tweakeys.

   (a) From the knowledge of $ETK[5, 8, 15]$, $\Delta Y_2[8, 5, 15]$ can be determined. Based on the properties of MC^{-1} operation on col(3) of X_3, we have $\Delta Y_2[5] = \Delta Y_2[15]$ and $\Delta Y_2[15] = \Delta Y_2[8]$. This will lead to two s-bit filters.

   The time complexity of this step is $N.2^{8s}$ and the number of tests left for the next step is $N.2^{6s}$.

7. Since we know $TK21[1]$ from the previous steps, we can satisfy the round 21 by applying the following steps on all N message pairs on all lists for the remaining tweakeys. This will lead to the determination of the number of possible values of $TK21[4, 7]$:

   (a) Guess $TK21[4, 7]$. Compute Z_{20} and ΔZ_{20} as shown in Figure 5. Checking if $X_{20}[10] = X_{20}[14]$ will lead to an s-bit filter on the remaining tweakeys.

   At this step, the attacker can uniquely determine $tk_1^1[0, 5]$ and $tk_2^1[0, 5]$ from the knowledge of $TK23[1, 6]$ and $TK21[4, 7]$.

   The time complexity of this step is $N.2^{8s}$ and the number of tests left for the next step is $N.2^{7s}$.

8. Satisfying the round 20, by applying the following steps on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK20[2, 6]$:

(a) From the knowledge of $Z_{20}[14]$ and $\Delta Z_{20}[14]$ we can compute $\Delta X_{20}[14]$. Based on the properties of MC operation on col(3) of W_{19}, we have $\Delta X_{20}[2] = \Delta X_{20}[14]$. Since $Y_{20}[2] = Z_{20}[2] \oplus TK20[2]$, we have

$$S^{-1}(Y_{20}[2]) \oplus S^{-1}(Y_{20}[2] \oplus \Delta Y_{20}[2]) = \Delta X_{20}[2].$$

Now by using lemma 4, $TK20[2]$ can be simply determined.

(b) Guess $TK20[6]$. Compute $\Delta X_{19}[8]$ and for each L_i list, checking if $\Delta X_{19}[8] = TDT[3][i]$. This will generally lead to an s-bit filter.

At this step, the attacker can uniquely determine $tk_1^1[9, 10]$ and $tk_2^1[9, 10]$ from the knowledge of $TK22[4, 5]$ and $TK20[2, 6]$. The time complexity of this step is $N.2^{8s}$ and the number of tests left for the next step is $N.2^{7s}$.

9. Satisfying the first round, by applying the following step on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $ETK[11]$:

(a) Guess $ETK[11]$. Since $ETK[1, 2, 5-9, 14, 15]$ are known from the previous steps, we can compute Y_2 and ΔY_2 as shown in Figure 5.

At this step, the attacker can uniquely determine $tk_1^1[6]$ and $tk_2^1[6]$ from the knowledge of $TK23[7]$ and $ETK[11]$. The time complexity of this step is $N.2^{8s}$ and the number of tests left for the next step is $N.2^{8s}$.

10. Satisfying the second round, by applying the following step on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK2[1, 2, 6]$:

(a) Guess $TK2[1, 2, 6]$. Knowledge of these cells allows the attacker to compute Y_3 and ΔY_3 as shown in Figure 5. Therefore, from the knowledge of $TK3[1]$, $\Delta Y_4[1]$ can be simply determined. Checking if $\Delta Y_4[1] = TDT[2][i]$, for each pair on L_i list, will certainly lead to an s-bit filter.

At this step, the attacker can uniquely determine $tk_1^1[8, 12, 15]$ and $tk_2^1[8, 12, 15]$ from the knowledge of $TK22[2, 3, 6]$ and $TK2[1, 2, 6]$. The time complexity of this step is $N.2^{11s}$ and the number of tests to verify the impossible distinguisher is $N.2^{10s}$.

Complexity analysis In this attack, the parameters are as follows:

- $|\Delta_{in}| = 4s$, $|\Delta_{out}| = 16s$.
- $c_{in} = 3s$. For more details, there is

  $2s$ bit-conditions in the ΔX_2 propagates to ΔX_3,
  s bit-conditions in the ΔX_4 propagates to ΔY_4.

  So we can verify the differential $\Delta P \to \Delta Y_4$ with probability $\frac{1}{2^{3s}}$.
- $c_{out} = 16s$. For more details, there is

  $4s$ bit-conditions in the ΔX_{23} propagates to ΔX_{22},
  $3s$ bit-conditions in the ΔX_{22} propagates to ΔX_{21},
  $6s$ bit-conditions in the ΔX_{21} propagates to ΔX_{20},
  $3s$ bit-conditions in the ΔX_{20} propagates to ΔX_{19}.

  So we can verify the differential $\Delta C \to \Delta X_{19}$ with probability $\frac{1}{2^{16s}}$.
- $|k_{in} \cup k_{out}| = 29s$. For more details, the key information used in the attack is

  $26s$ for $tk_1^1[i], tk_2^1[i]$, $i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15$,
  $3s$ for $tk_1^1[i] \oplus LFSR^{11}(tk_2^1[i])$, $i = 11, 13, 14$.

Thus, based on the method proposed in [BNPS14], we can calculate the data, time, and memory complexity as follows:

The probability that a given pair of inputs has a difference Δ_{in} and an output difference Δ_{out}, under a random key guess, is $2^{-(c_{in}+c_{out})}$. The probability for a trial key to be placed in the set of possible keys should be small enough so that number of pairs, N, must be chosen appropriately. This probability is calculated as follows:

$$P = (1 - 2^{-(c_{in}+c_{out})})^N = (1 - 2^{-19s})^{2^{x+8s}} \simeq e^{-2^{x-11s}}.$$

By adopting the strategy presented in [BNPS14], we consider the number of pairs such that $e^{-2^{x-11s}} < \frac{1}{2}$, to reduce the exhaustive search by at least one bit. By choosing $x = 45.47$ (resp. $x = 91.40$) in the case of SKINNY-64-128 (resp. SKINNY-128-256), the remaining 29-nibble subkey space is reduced to

$$\begin{aligned} TK_{remain} = 2^{|k_{in} \cup k_{out}|}(1 - 2^{-(c_{in}+c_{out})})^N &\simeq 2^{29s} e^{-2^{x-11s}} \\ &\simeq 2^{29\times 4} e^{-2^{45.47-11\times 4}} \simeq 2^{112} \end{aligned}$$

(resp. $2^{29\times 8} e^{-2^{91.40-11\times 8}} \simeq 2^{216.70}$). By exhaustively searching the $TK_{remain} = 2^{112}$ (resp. $2^{216.70}$) remaining tweakey candidates with 2^{3s} remaining tweakey bits ($TK_{remain} \times 2^{3s}$), which are not used in the attack, we can recover the tweakey candidates.

Data complexity of this attack is

$$\begin{aligned} D = 2^{N \times 2^{n+1-|\Delta_{in}|-|\Delta_{out}|}} &= 2^{2^{x+2|\Delta_{in}|-(n-|\Delta_{out}|)} \times 2^{n+1-|\Delta_{in}|-|\Delta_{out}|}} \\ &= 2^{x+|\Delta_{in}|+1} = 2^{45.47+4\times 4+1} = 2^{62.47} \end{aligned}$$

(resp. $2^{91.40+4\times 8+1} = 2^{124.41}$) chosen plaintexts. Then, the time needed for obtaining N pairs of messages (D), multiplying the number of pairs by the average time needed for trying key candidates out ($N.2^{11s}$) and the time needed for trying the remaining key candidates out and recovering the complete key will determine the complexity. So the the attack requires

$$\begin{aligned} T &= D + N.2^{11s} + TK_{remain} \times 2^{3s} \\ &= 2^{62.47} + 2^{121.47} + 2^{112} \times 2^{12} = 2^{124.21} \end{aligned}$$

(resp. $T = 2^{124.41} + 2^{243.41} + 2^{216.70} \times 2^{24} = 2^{243.61}$) encryptions which is the overall time complexity in case of SKINNY-64-128 (resp. SKINNY-128-256). The memory complexity of the attack is dominated by the memory needed for storing $2^{x+32=77.47}$ (resp. $2^{x+64=155.41}$) pairs, which is $2^{77.47}$ (resp. $2^{155.41}$) in case of SKINNY-64-128 (resp. SKINNY-128-256).

Remark

In our attack, the time complexity of step 10 is $N.2^{11s}$. Actually, similar to the method used in [LGS17], this complexity can be reduced to $N.2^{9s}$ and it reduces the total complexity as a factor of $2^{-0.2}$ in our attack. For this purpose, step 10 can be performed as follows: For each pair on list $L_i, (i = 1, \cdots, 15)$, the attacker guesses $TK2[2]$. Knowledge of this cell allows her to compute $\Delta Y_3[14]$ and so $\Delta X_4[1]$. From the knowledge of $\Delta Y_4[1](= TDT[1][i])$ and by using lemma 4, $X_4[1]$ can be simply determined. Now, two subtweakey cells $TK2[2, 6]$ can be calculated as given below:
since $S(X_3[1]) \oplus S(X_3[11]) = TK3[1] \oplus X_4[1] \oplus W_3[13]$, by constructing a table to store input values $X_3[1]$ and $X_3[11]$ for the two S-boxes for each possible right hand value, input

values can be retrieved. For each right hand value, we have 2^s possible combinations of $X_3[1]$ and $X_3[11]$. Then we can calculate $TK2[1] = X_3[1] \oplus W_2[9] \oplus W_2[13] \oplus Y_2[1]$ and $TK2[6] = X_3[11] \oplus W_2[11] \oplus Y_2[6]$. So the time complexity of this step can be considered as $N.2^{9s}$ and the number of tests to verify the impossible distinguisher would be $N.2^{10s}$.

5 Zero-Correlation Linear Attack

5.1 Searching Zero-correlation Linear distinguishers of SKINNY

In this section, we use "0" to denote a zero mask, Γ^i to denote a nonzero mask in i-th nibble ($i = 0, \ldots, 15$), and "?" to denote a zero or nonzero mask. Also, we use $\Gamma^i_{in} \overset{r}{\nrightarrow} \Gamma^j_{out}$ to show that the correlation of linear approximation of r-round SKINNY with input mask Γ^i_{in} (i-th nibble of input) to output mask Γ^j_{out} (j-th nibble of output) is zero.

5.1.1 9-round Zero-correlation linear distinguishers for SKINNY

By using miss-in-the-middle approach, we firstly found a 9-round zero-correlation distinguisher as follows:

$$(\Gamma^{15}_{in}) \overset{9}{\nrightarrow} (\Gamma^{12}_{out}).$$

As can be seen in Figure 6, in the encryption direction,we see that for any 4-round non-zero linear characteristic with input mask of (Γ^{15}_{in}), the linear mask of the internal state must be

$$(0, 0, \Gamma^2, \Gamma^3, ?, \Gamma^5, \Gamma^6, 0, \Gamma^8, 0, \Gamma^{10}, 0, \Gamma^{12}, \Gamma^{13}, ?, \Gamma^{15}). \tag{3}$$

Similarly, in the decryption direction for any 5-round non-zero linear characteristic with output mask of Γ^{12}_{out}, the linear mask of the internal state must be

$$(\Gamma^0, ?, ?, ?, \Gamma^4, ?, ?, ?, ?, ?, ?, ?, \Gamma^{12}, \Gamma^{13}, ?, 0). \tag{4}$$

If we combine (3) and (4) with each other, we derive a 9-round zero-correlation linear distinguisher for SKINNY.

We searched for all 9-round zero-correlation characteristics with the miss-in-the-middle technique and we list them all in Table 22. Based on this table, there are 172 different characteristics with single active cells in input and output masks.

Table 22: Zero-correlation linear approximations $\Gamma^i_{in} \overset{r}{\nrightarrow} \Gamma^j_{out}$ for 9-round SKINNY.

$(i,\ j)$
$(0,0), (0,1), \ldots, (0,15), (1,0), (1,1), \ldots, (1,15)$
$(2,0), (2,1), \ldots, (2,15), (3,0), (3,1), \ldots, (3,15)$
$(4,4), (4,5), (4,6), (4,7), (5,4), (5,5), (5,6), (5,7)$
$(6,4), (6,5), (6,6), (6,7), (7,4), (7,5), (7,6), (7,7)$
$(8,4), (8,5), \ldots, (8,11), (8,13), (8,14), (8,15)$
$(9,4), (9,5), \ldots, (8,12), (9,14), (9,15)$
$(10,4), (10,5), \ldots, (10,13), (10,15)$
$(11,4), (11,5), \ldots, (11,14), (12,4), (12,5), \ldots, (12,15)$
$(13,4), (13,5), \ldots, (13,15), (14,4), (14,5), \ldots, (14,15)$
$(15,4), (15,5), \ldots, (15,15)$

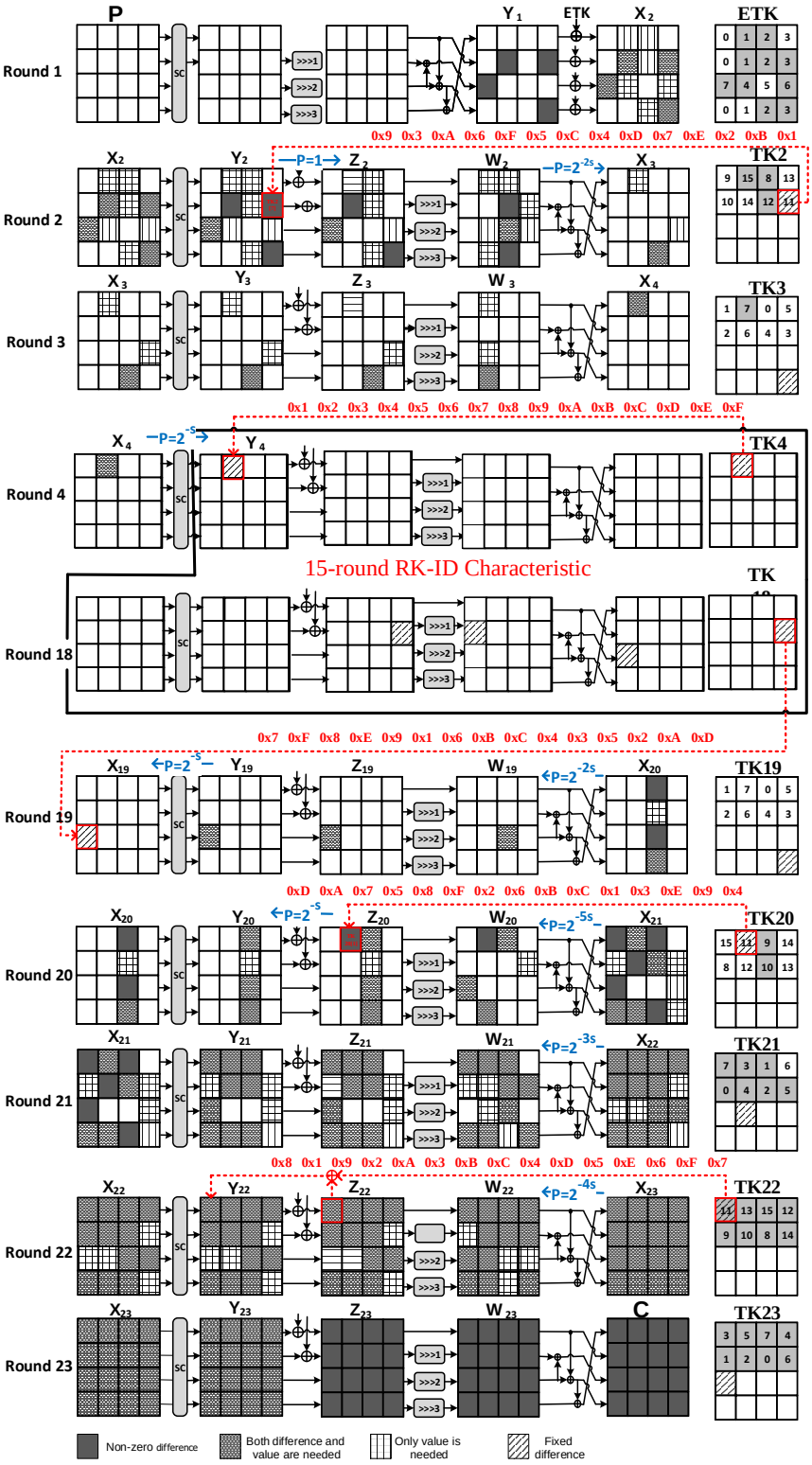

Figure 5: Related-tweakey impossible differential attack on 23-round of SKINNY-n-2n. Differences which are added from tweakey to the state are shown only for the case of $s = 4$.

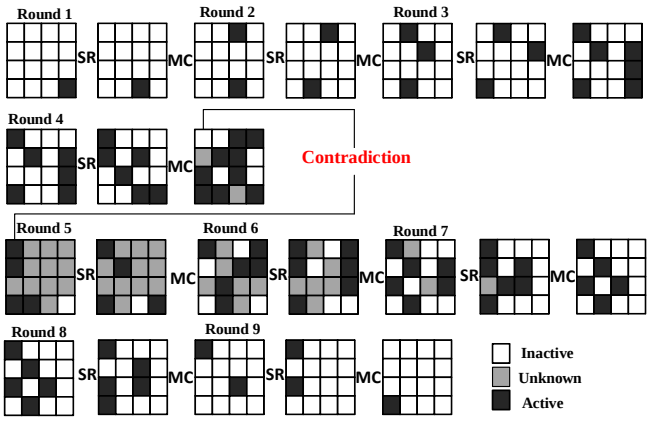

Figure 6: The 9-round distinguisher for SKINNY. SR and MC stand for ShiftRows and MixColumns, respectively. SubCel, AddConstant and AddRoundTweakey are omitted since they are not related here.

Table 23: Zero-correlation linear approximations $\Gamma_{in}^{i} \xrightarrow{r} \Gamma_{out}^{j}$ for 10-round SKINNY.

$(i,\ j)$
$(0,4),(0,5),(0,6),(0,7),(1,4),(1,5),(1,6),(1,7)$
$(2,4),(2,5),(2,6),(2,7),(3,4),(3,5),(3,6),(3,7)$

5.1.2 10-round Zero-correlation linear distinguishers for SKINNY

Using the MILP technique, we found 16 zero-correlation linear characteristics reaching 10 rounds, which are listed in Table 23. After finding the trails by MILP results, we tried to verify them by miss-in-the-middle technique, for which the procedure comes in the following.

For example, one of the 10-round zero-correlation linear characteristics is $(\Gamma_{in}^{0}) \xrightarrow{10} (\Gamma_{out}^{4})$. It should be noted that we tried to obtain this characteristic by using miss-in-the-middle approach and considering r_1 rounds forward and r_2 rounds backward ($r_1 + r_2 = 10$), but we did not reach any contradiction directly by considering different r_1 and r_2 rounds. Hence, to show that the 10-round zero-correlation linear characteristic with this input and output mask exists, we firstly construct a 9-round zero-correlation distinguisher as shown in Figure 7. This distinguisher consists of a forward part (along the encryption direction) and a backward part (along the decryption direction). After encrypting 4 rounds in the forward part and 5 rounds in the backward part, a contradiction will happen in the first cell of the middle state which is shown in Figure 7.

By decrypting (or encrypting) 1 more round in the backward part (or forward part), no contradiction will be found but we used a trick here to reach a contradiction in the 10-round characteristic. There are 3 possible types of cell conditions in each state: active, inactive, and unknown. As we know, the active and inactive cells have deterministic conditions but unknown cells can take any condition so we can assume them to be active or inactive and see whether this assumption can make any change in the condition of the deterministic cells to reach a contradiction. To explain more, the trick is to decrypt one more round after the contradiction place (which here is the state C) in the 9-round trail and derive state B. As we know, states A and B are equivalent but derived from two different directions, i.e., forward and backward. So, we can assume the unknown cells of state B to have the same condition of the corresponding cell in state A. In this step, we should try to assume some of these cells to have the corresponding condition; then, we

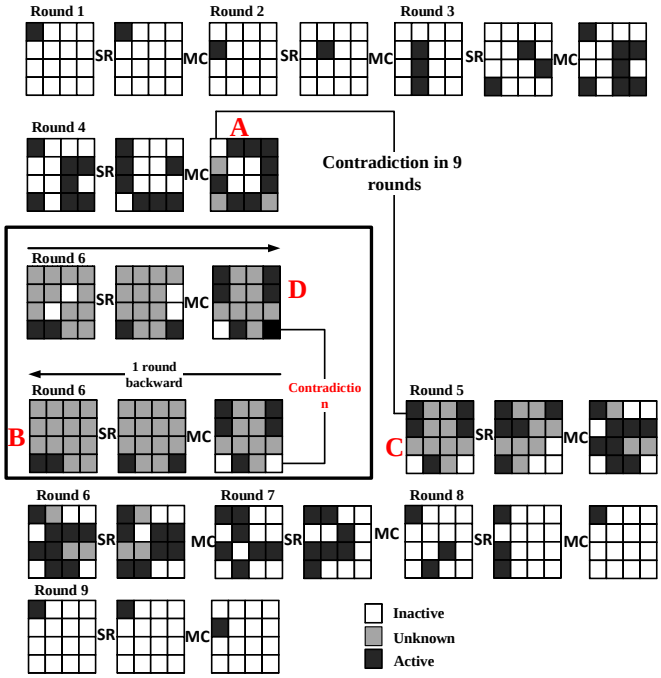

Figure 7: Zero-correlation characteristic for 9-round (extended to 10-round) SKINNY. SR and MC stand for ShiftRows and MixColumns, respectively. Subcells, AddConstant and AddTweakey are omitted since they are not related here.

encrypt one more round under this assumption to check if it will cause any changes in the deterministic cells of the first state of the next round. As it is shown in Figure 7, we assumed the 6-th and the 9-th cells to be inactive and encrypted one more round to derive D. As we can see, in this case, the 15-th cell of the input mask of this new round will change and become active and this is a contradiction. More details are depicted in Figure 7.

6 Zero-Correlation Linear Cryptanalysis of SKINNY

In this section, we investigate the security of SKINNY64-128 by using zero-correlation linear cryptanalysis. Note that we use the 9-round zero-correlation distinguisher described in the section 5.1, since it provides better results in terms of time and memory complexity. We present the key recovery attacks on 18-round SKINNY64-128. The 14-round attack of SKINNY64-64 is presented in Appendix B. In this section, s^i means the internal state of i-th round and $s^i(j)$ means the j-th cell of the state i.

6.1 Zero-correlation linear cryptanalysis of SKINNY-64 with 128-bit tweakey

As shown in Figure 8, we can append 5 rounds after the distinguisher and add 4 rounds before the distinguisher. It means that the 9-round distinguisher starts from the 5-th round and ends at the 13-th round (round number starting from 1). In this way, we can attack 18-round SKINNY64-128.

Attack procedure

In this case, the attack procedure consists of two phases:

Phase one: Collect N pairs of plaintexts and corresponding ciphertexts. Guess 11 nibbles $TK^{18}(0,3,5,6,7)$, $TK^{17}(1,4,6)$, $TK^{16}(2,7)$, $TK^{15}(3)$, do the partial decryption and calculate $s^{13}(12)$ for each pair. Allocate an 8-bit counter$N0[s^1, s^{13}]$ for each of 2^{56} possible values of $(s^1||s^{13})$, where $s^1 = s^1(0,2,3,4,6,7,8,9,10,11,12,13)$ and $s^{13} = s^{13}(12)$, and set them to zero. Then, calculate the number of pairs of plaintext-ciphertext with given values s^1 and s^{13} and store it in $N0[s^1, s^{13}]$. Hence, around 2^{64} plaintext-ciphertext pairs are divided into 2^{56} different states. The expected pairs for each state are about 2^8. So the assumption $N0$ as a 1-byte counter is sufficient.

Phase two:

1. Guess the 6 nibbles $TK^1(0,2,3,4,6,7)$. Then, allocate a counter $N1[s^2, s^{13}]$ for each of 2^{36} possible values of $(s^2||s^{13})$, where $s^2 = s^2(0,3,4,6,9,10,11,12)$, and set them to zero. For all 2^{44} possible values of s^1, encrypt s^1 one round to obtain s^2 and update the value $N1[s^2, s^{13}] = N1[s^2, s^{13}] + N0[s^1, s^{13}]$ for all 2^4 values of s^{13}.
2. Guess 4 nibbles $TK^2(0,3,4,6)$. Then, allocate a counter $N2[s^3, s^{13}]$ for each of 2^{24} possible values of $(s^3||s^{13})$, where $s^3 = s^3(3,4,9,11,12)$, and set them to zero. For all 2^{32} possible values of s^2, encrypt s^2 one round to obtain s^3 and update the value $N2[s^3, s^{13}] = N2[s^3, s^{13}] + N1[s^2, s^{13}]$ for all 2^4 values of s^{13}.
3. Guess 2 nibbles $TK^3(3,4)$. Then, allocate a counter $N3[s^4, s^{13}]$ for each of 2^{12} possible values of$(s^4||s^{13})$, where $s^4 = s^4(3,9)$, and set them to zero. For all 2^{20} possible values of s^3, encrypt s^3 one round to obtain s^4 and update the value $N3[s^4, s^{13}] = N3[s^4, s^{13}] + N2[s^3, s^{13}]$ for all 2^4 values of s^{13}.
4. Guess the nibble $TK^4(3)$. Then, allocate a counter $N4[s^5, s^{13}]$ for each of 2^8 possible values of $(s^5||s^{13})$, where $s^5 = s^5(4)$, and set them to zero. For all 2^8 possible values of s^4, encrypt s^4 one round to obtain s^5 and update the value $N4[s^5, s^{13}] = N4[s^5, s^{13}] + N3[s^4, s^{13}]$ for all 2^4 values of s^{13}. The counter $N4[s^5, s^{13}]$ is then taken as the desired counter $V[z]$, where z is the 1-byte data value $s^5||s^{13}$.
5. Compute the statistical value

$$T = \frac{N * 2^4}{1 - 2^{-4}} \sum_{S^{13}=4}^{2^4-1} \sum_{S^5=0}^{2^4-1} \left(\frac{N_4[S^5, S^{13}]}{N} - \frac{1}{2^4} \right)^2 .$$

   If $T < t$, then the guessed key is taken a possible candidate.
6. Do exhaustive search for all keys that correspond to the guessed subkey bits.

Attack complexity

The memory complexity of the attack is 2^{56} bytes which is dominated by step 2. The time complexity of phase one is equal to $N \times 2^{44}$. The time complexity of the steps between 1 and 4 depends on the number of accesses to the memory. The time complexity for each round can be derived as follows.

Step 1: $2^{(44+24)} \times 2^{48} \times 2^4 = 2^{120}$ memory accesses needed, since we should guess 24 bits for TK^1 (plus 44 bits guessed in phase one), encrypt s^1 one round for2^{48} values, and update $N1$ for 2^4 times.

Step 2: $2^{(68+16)} \times 2^{32} \times 2^4 = 2^{120}$ memory accesses needed, since we should guess 16 bits for TK^2, and for 2^{32} values encrypt s^2 one round and update $N2$ for 2^4 times.

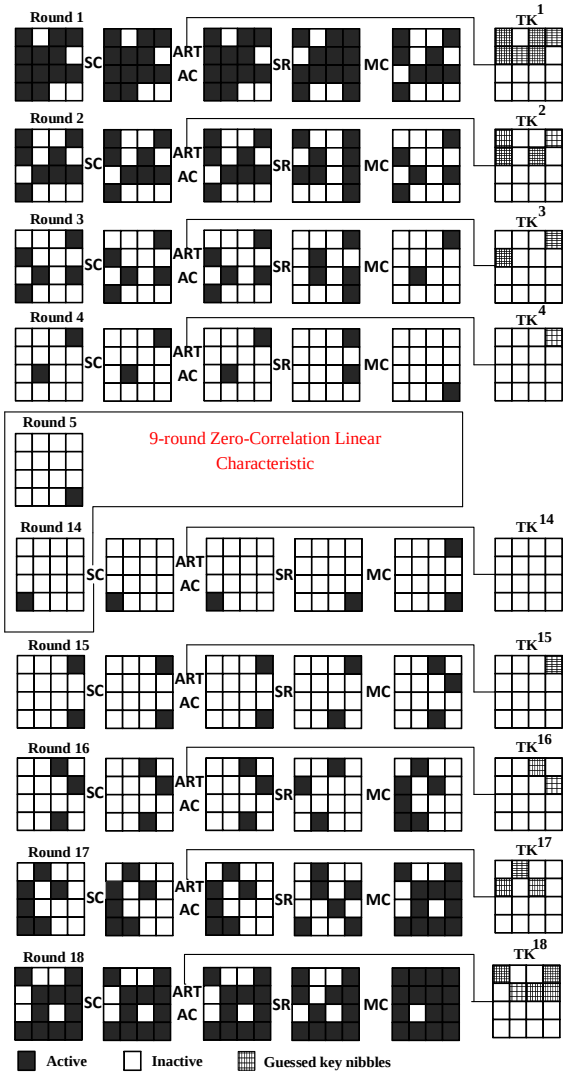

Figure 8: 18-round key recovery with zero-correlation linear attack for SKINNY-64 with 128-bit tweakey

Table 24: Time and data complexity for different values of α and β for SKINNY-64 with 128-bit tweakey

α	β	P_S	Time complexity	Data complexity
$2^{-7.3}$	2^{-1}	0.99	2^{127}	$2^{62.95}$
$2^{-3.3}$	2^{-1}	0.89	2^{127}	$2^{62.37}$
$2^{-3.3}$	2^{-3}	0.89	2^{125}	$2^{63.30}$
$2^{-2.7}$	2^{-1}	0.84	2^{127}	$2^{62.15}$
$2^{-2.7}$	2^{-2}	0.84	2^{126}	$2^{63.47}$
$2^{-1.7}$	2^{-1}	0.69	2^{127}	$2^{61.34}$
$2^{-1.7}$	2^{-5}	0.69	2^{123}	$2^{63.58}$
$2^{-1.7}$	2^{-2}	0.69	2^{126}	$2^{62.58}$

Step 3: $2^{(84+8)} \times 2^{20} \times 2^4 = 2^{116}$ memory accesses needed, since we should guess 8 bits for TK^3, encrypt s^3 one round for 2^{20} values and update $N3$ for 2^4 times.

Step 4: $2^{(92+4)} \times 2^8 \times 2^4 = 2^{108}$ memory accesses needed, since we should guess 4 bits for TK^4, encrypt s^4 one round for 2^8 values and update N3 for 2^4 times.

Step 5: $2^{96} \times \times 2^8 = 2^{105}$ times of reading the 1-byte memory.

Step 6: needs $2^{128} \times \beta$ full encryption.

The total complexity of time and data is available in Table 24 .

7 Conclusion

In this work, we presented the related-tweakey impossible differential and zero-correlation linear characteristics on different variants of SKINNY block cipher. For SKINNY-n-n and SKINNY-n-2n, we searched all of the related-tweakey impossible differential characteristics using MILP technique. Moreover, we found 13-round and 15-round related-tweakey impossible differential characteristics for SKINNY-n-n and SKINNY-n-2n, respectively. Utilizing these characteristics, we proposed 19-round related-tweakey impossible differential attack on SKINNY-n-n and 23-round attack on SKINNY-n-2n. We also constructed 9 and 10-round zero correlation linear distinguishers and attacked 14 and 18 round of SKINNY-64-64 and SKINNY-64-128 respectively, by extending the 9-round trail. Based on the MILP results, we claim that the given characteristics are the longest under the assumption of having a single active bit in input and output masks (and tweakeys in related-tweakey cases).

References

[AAA+15] Mohamed Ahmed Abdelraheem, Javad Alizadeh, Hoda A Alkhzaimi, Mohammad Reza Aref, Nasour Bagheri, and Praveen Gauravaram. Improved linear cryptanalysis of reduced-round simon-32 and simon-48. In *International Conference in Cryptology in India*, pages 153–179. Springer, 2015.

[ABC+17] Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, and Gaoli Wang. Related-key impossible-differential attack on reduced-round skinny. In *International Conference on Applied Cryptography and Network Security*, pages 208–228. Springer, 2017.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 12–23. Springer, 1999.

[Bih94] Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.

[BJK+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 *Annual Cryptology Conference*, pages 123–153. Springer, 2016.

[BKL+07] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *CHES*, volume 4727, pages 450–466. Springer, 2007.

[BLNPS18] Christina Boura, Virginie Lallemand, María Naya-Plasencia, and Valentin Suder. Making the impossible possible. *Journal of Cryptology*, 31(1):101–133, 2018.

[BLNW12] Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. Integral and multidimensional linear distinguishers with correlation zero. In *AsiaCrypt*, volume 7658, pages 244–261. Springer, 2012.

[BNPS14] Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon. *ASIACRYPT (1)*, 8873:179–199, 2014.

[BR14] Andrey Bogdanov and Vincent Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Designs, codes and cryptography*, 70(3):369–383, 2014.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

[BTCS+15] Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. The simon and speck lightweight block ciphers. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

[BW12] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In *Fast Software Encryption*, pages 29–48. Springer, 2012.

[CJF+16] Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. Technical report, Cryptology ePrint Archive, Report 2016/689, 2016.

[Der16] Patrick Derbez. Note on impossible differential attacks. In *International Conference on Fast Software Encryption*, pages 416–427. Springer, 2016.

[DKR97] Joan Daemen, Lars R Knudsen, and Vincent Rijmen. The block cipher square. In *Fse*, volume 97, pages 149–165. Springer, 1997.

[FWG+16] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In *International Conference on Fast Software Encryption*, pages 268–288. Springer, 2016.

[JD03] Goce Jakimoski and Yvo Desmedt. Related-key differential cryptanalysis of 192-bit key aes variants. In *International Workshop on Selected Areas in Cryptography*, pages 208–221. Springer, 2003.

[JNP14] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Tweaks and keys for block ciphers: the tweakey framework. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 274–288. Springer, 2014.

[Knu98] Lars Knudsen. Deal-a 128-bit block cipher. *complexity*, 258(2):216, 1998.

[KW02] Lars Knudsen and David Wagner. Integral cryptanalysis. In *International Workshop on Fast Software Encryption*, pages 112–127. Springer, 2002.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security analysis of skinny under related-tweakey settings (long paper). *IACR Transactions on Symmetric Cryptology*, 2017(3):37–72, 2017.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 386–397. Springer, 1993.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *International Conference on Information Security and Cryptology*, pages 57–76. Springer, 2011.

[SBA17] Sadegh Sadeghi, Nasour Bagheri, and Mohamed Ahmed Abdelraheem. Cryptanalysis of reduced qtl block cipher. *Microprocessors and Microsystems*, 2017.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of aes, skinny, and others with constraint programming. *IACR Transactions on Symmetric Cryptology*, 2017(1):281–306, 2017.

[SHW+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. Technical report, Cryptology ePrint Archive, Report 2014/747, 2014.

[SHW+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 *International Conference on the Theory and Application of Cryptology and Information Security*, pages 158–178. Springer, 2014.

[SN14] Hadi Soleimany and Kaisa Nyberg. Zero-correlation linear cryptanalysis of reduced-round lblock. *Designs, codes and cryptography*, 73(2):683–698, 2014.

[ST17] Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 185–215. Springer, 2017.

[TAY17] Mohamed Tolba, Ahmed Abdelkhalek, and Amr M Youssef. Impossible differential cryptanalysis of reduced-round skinny. In *International Conference on Cryptology in Africa*, pages 117–134. Springer, 2017.

[XZBL16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 648–678. Springer, 2016.

[YZS+15] Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D Aagaard, and Guang Gong. The simeck family of lightweight block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 307–329. Springer, 2015.

A 19-round Related Tweakey Impossible Differential Attack on SKINNY-n-n

In this section, we present the details of a 19-round attack on SKINNY-n-n utilizing the 13-round related-tweakey impossible differential trail shown in Figure 3 and extend it by 3 rounds in both backward and forward directions. The same as the previous section, instead of the tweakey $TK1$, we can obtain the equivalent tweakey ETK in the first round, start the tweakey recovery attack at Y_1, and recover the plaintext P by applying MC^{-1}, SR^{-1}, AC^{-1} and SC^{-1} layers on Y_1.

Data Collection The adversary should construct 2^x structures at Y_1 and consider all the possible values in 4 cells $Y_1[1, 4, 11, 14]$ for each structure, while the remaining cells take a fixed value. By using 2^{x+4s} messages, we can generate 2^{x+8s} pairs of messages $(P, \overline{P})$, then ask the encryption oracle to obtain the corresponding ciphertexts $(C, \overline{C})$. We have 7 s-bit filters after peeling off the last MC layer from the ciphertext to W_{19}. Therefore, we have $N = 2^{x+8s-7s=x+s}$ remaining pairs to do the attack and decrypt them partially over SR^{-1} and compute Z_{19}.

Tweakey Recovery For each of the N pairs

1. Satisfying the round 19, by applying the following steps on all N message pairs, leads to the determination of the number of possible values of $TK19[0, 2, 5, 6]$:
   (a) From the knowledge of $Z_{19}[12]$ and $\Delta Z_{19}[12]$ we can compute $\Delta X_{19}[12]$. Based on the properties of MC operation on col(1) of W_{18}, we have $\Delta X_{19}[0] = \Delta X_{19}[12]$. Since $Y_{19}[0] = Z_{19}[0] \oplus TK19[0]$ and we have

$$S^{-1}(Y_{19}[0]) \oplus S^{-1}(Y_{19}[0] \oplus \Delta Y_{19}[0]) = \Delta X_{19}[0],$$

   thus by using lemma 4, $TK19[0]$ can be determined.

(b) We can compute $\Delta X_{19}[13]$ from the knowledge of $Z_{19}[13]$ and $\Delta Z_{19}[13]$. Based on the properties of MC operation on col(2) of W_{18}, we have $\Delta X_{19}[13] = \Delta X_{19}[5]$. Since $Y_{19}[5] = Z_{19}[5] \oplus TK19[5]$ and we have

$$S^{-1}(Y_{19}[5]) \oplus S^{-1}(Y_{19}[5] \oplus \Delta Y_{19}[5]) = \Delta X_{19}[5],$$

by using lemma 4, $TK19[5]$ can be determined.

(c) Based on the properties of MC operation on col(3) of W_{18}, ,we have $\Delta X_{19}[2] = \Delta X_{19}[6] = \Delta X_{19}[14]$. Since $Z_{19}[2] = Y_{19}[2] \oplus TK19[2]$ and $Z_{19}[6] = Y_{19}[6] \oplus TK19[6]$ we have

$$S^{-1}(Y_{19}[2]) \oplus S^{-1}(Y_{19}[2] \oplus \Delta Y_{19}[2]) = \Delta X_{19}[2],$$

$$S^{-1}(Y_{19}[6]) \oplus S^{-1}(Y_{19}[6] \oplus \Delta Y_{19}[6]) = \Delta X_{19}[6],$$

by using lemma 4 we can determine $TK19[2, 6]$.

(d) From the knowledge of $\Delta Z_{19}[8, 12]$ and $Z_{19}[8, 12]$, we can determine $\Delta X_{19}[8, 12]$. Based on the properties of MC operation on col(1) of W_{18}, we have $\Delta X_{19}[8] = \Delta X_{19}[12]$; checking the correctness of this equality, will lead to an s-bit filter. Also, checking if $\Delta X_{19}[5] = \texttt{0xi}$, stores the pair on the list L_i. Obviously, each pair will be saved on one list and there is no need to filtering in this step.

The time complexity of this step is N and the number of tests left for the next step is $N.2^{-s}$.

2. Since we know $ETK[0, 1, 4, 5, 11, 13]$, from the knowledge of $TK19[0, 2, 5, 6]$, we can satisfy the round 1 by applying the following steps on all N message pairs on all lists for the remaining tweakeys. This will lead to the determination of the number of possible values of $ETK1[7, 9, 14]$:

(a) Since we know $ETK[11]$, $\Delta Y_2[11]$ can be computed. Based on the properties of MC^{-1} operation on col(2) of X_3, we have $\Delta Y_2[11] = \Delta Y_2[14]$. Since $X_2[14] = Y_1[14] \oplus ETK[14]$ and we have

$$S(X_2[14]) \oplus S(X_2[14] \oplus \Delta X_2[14]) = \Delta Y_2[14],$$

hence lemma 4 helps us to determine $ETK[14]$.

(b) Guess $ETK[10]$. Compute Y_2 and ΔY_2 as shown in Figure 9. Checking if $\Delta Y_2[1] = \texttt{0xi}$, for each pair on L_i list, will lead to a total of an s-bit filter. Also, checking if $\Delta Y_2[4] = \Delta Y_2[11]$, will lead to another s-bit filter.

The time complexity of this step is N and the number of tests left for the next step is $N.2^{-2s}$.

3. Since we know $TK19[1, 4]$ from the knowledge of $ETK[8, 14]$, we can satisfy the round 19 by applying the following steps on all N message pairs on all lists for the remaining tweakeys. This will lead to the determination of the number of possible values of $TK19[7]$:

(a) Guess $TK19[7]$. Compute Z_{18} and ΔZ_{18} as shown in Figure 9. Checking if $\Delta X_{18}[10] = \Delta X_{18}[14]$, will lead to an s-bit filter.

The time complexity of this step is $N.2^{-s}$ and the number of tests left for the next step is $N.2^{-2s}$.

4. Satisfying the round 18, by applying the following steps on all N message pairs on all lists for the remaining tweakeys,leads to the determination of the number of possible values of $TK18[2, 6]$:

(a) From the knowledge of $Z_{18}[14]$ and $\Delta Z_{18}[14]$ we can compute $\Delta X_{18}[14]$. Based on properties of MC operation on col(3) of W_{17}, we have $\Delta X_{18}[2] = \Delta X_{18}[14]$. Since $Y_{18}[2] = Z_{18}[2] \oplus TK18[2]$ we have

$$S^{-1}(Y_{18}[2]) \oplus S^{-1}(Y_{18}[2] \oplus \Delta Y_{18}[2]) = \Delta X_{18}[2].$$

Now by using lemma 4, $TK18[2]$ can be simply determined.

(b) Guess $TK18[6]$. Compute $\Delta X_{17}[8]$ and for each pair on L_i list, checking if $\Delta X_{17}[8] = \texttt{0xi}$, this will lead to a total of an s-bit filter.

The time complexity of this step is $N.2^{-s}$ and the number of tests left for the next step is $N.2^{-2s}$.

5. Satisfying the round 2, by applying the following steps on all N message pairs on all lists for the remaining tweakeys, leads to the determination of the number of possible values of $TK2[0, 1, 5]$:

(a) Guess $TK2[0, 1, 5]$. Knowledge of these cells allows the attacker to compute Y_3 and ΔY_3 as shown in Figure 9. Therefore, from the knowledge of $TK3[0](= TK19[2])$, $\Delta Y_4[0]$ can be simply determined. For each pair on L_i list, checking if $\Delta Y_4[0] = \texttt{0xi}$, this will lead us to a total of an s-bit filter.

The time complexity of this step is $N.2^s$ and the number of tests to verify the impossible distinguisher is N.

Complexity analysis In this attack $c_{in} = |\Delta_{in}| = 4s$, $c_{out} = 8s$, $|\Delta_{out}| = 9s$ and $|k_{in} \cup k_{out}| = 13s$. Thus, according to the formulas derived in the previous section, we can calculate the data, time and memory complexity as follows:

We consider the number of pairs such that $(1 - 2^{-12s})^{2^{x+s}} = e^{-2^{x-11s}} < \frac{1}{2}$, to reduce the exhaustive search by at least one bit. By choosing $x = 44.30$ (resp. $x = 89.47$), in case of SKINNY-64-64 (resp. SKINNY-128-128) the remaining 13-nibble subkey space is reduced to $2^{13\times4}e^{-2^{44.30-11\times4}} = 2^{50.22}$ (resp. $2^{13\times8}e^{-2^{89.47-11\times8}} = 2^{100}$). By exhaustively searching the $2^{50.22}$ (resp. 2^{100}) remaining key candidates with 2^{3s} remaining tweakey bits, which are not used in the attack, we can recover the tweakey candidates. Data complexity of this attack is $D = 2^{x+|\Delta_{in}|+1} = 2^{44.30+4\times4+1} = 2^{61.30}$ (resp. $2^{89.47+4\times8+1} = 2^{122.47}$) chosen plaintexts. Then, the attack requires $T = 2^{61.30} + 2^{52.3} + 2^{50.22} \times 2^{12} = 2^{62.83}$ (resp. $T = 2^{122.47} + 2^{105.47} + 2^{100} \times 2^{24} = 2^{124.43}$) encryption which is the overall time complexity in case of SKINNY-64-64 (resp. SKINNY-128-128). The memory complexity of the attack is dominated by the memory needed for storing $2^{x+4=48.30}$ (resp. $2^{x+8=97.47}$) pairs which is $2^{48.30}$ (resp. $2^{97.47}$) pairs after the ciphertext filtration to exclude the wrong keys in case of SKINNY-64-64 (resp. SKINNY-128-128).

B Zero-correlation linear cryptanalysis of SKINNY-64 with 64-bit tweakey

As shown in Figure 10, by a key recovery attack we can add 2 rounds before the distinguisher and 3 rounds after the distinguisher. It means that the 9-round distinguisher starts from the round 3 and ends at the round 11 (round number starting from 1). In this way, we can attack 14-round SKINNY64-64. The description of this attack is given below:

Attack procedure

1. Collect N pairs of plaintexts and the corresponding ciphertexts.

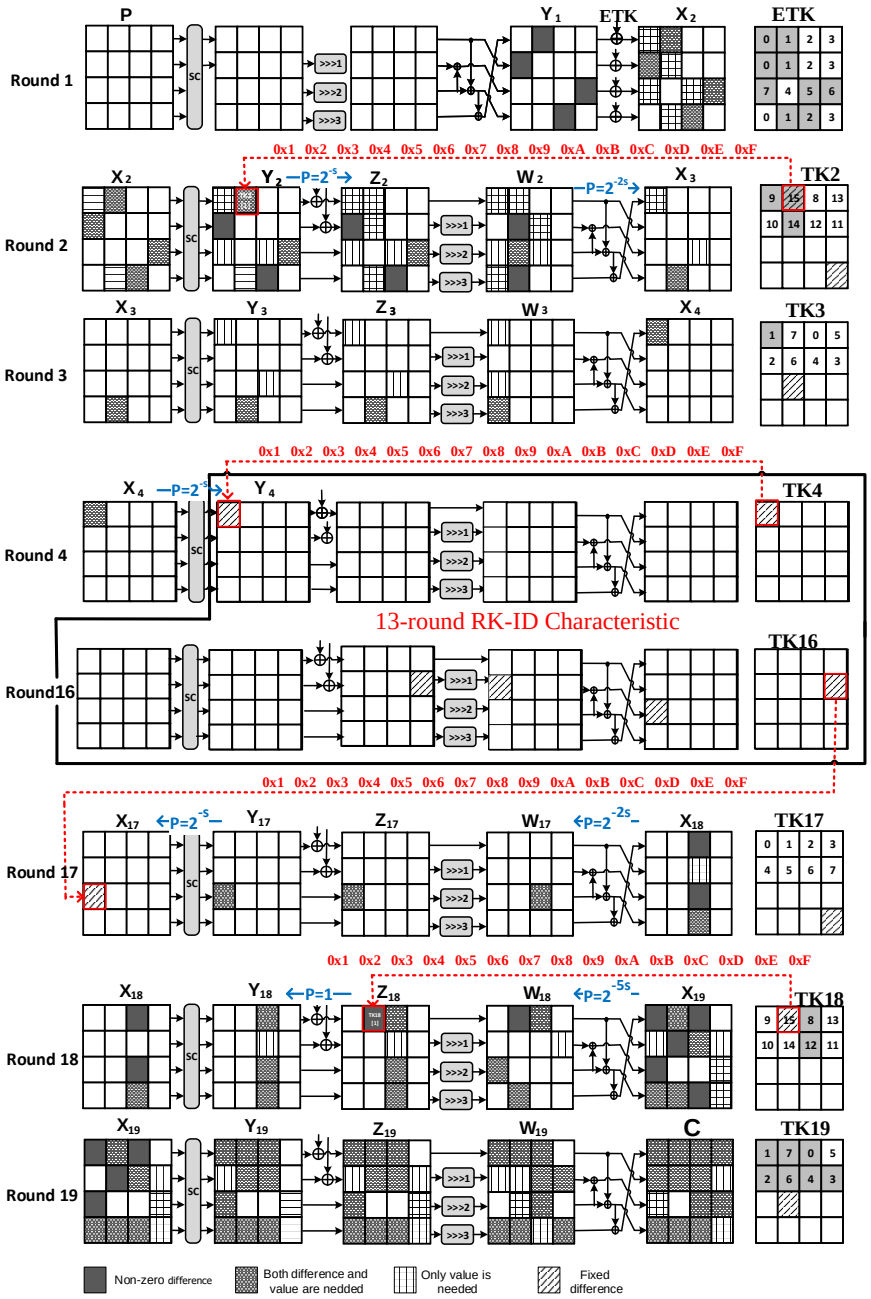

Figure 9: Related-tweakey impossible differential attack on 19-round of SKINNY-n-n. Differences which are added from tweakey to the state are shown only for the case of $s = 4$.

2. Allocate a 3-byte counter $N0[s^1, s^{14}]$ for each of 2^{44} possible values of $(s^1||s^{14})$, where $s^1 = s^1(3,4,9,11,12)$ and $s^{14} = s^{14}(2,5,7,9,13,14)$, and set them to zero. Then, calculate the number of pairs of plaintext-ciphertext with given values s^1 and s^{14} and store it in $N0[s^1, s^{14}]$. Hence, around 2^{44} plaintext-ciphertext pairs are divided into 2^{44} different states. The expected pairs for each state are about 2^{20}. So the assumption $N0$ as a 3-byte counter is sufficient.

3. Guess the 2 nibbles $TK^1(3,4)$. Then, allocate a counter $N1[s^2, s^{14}]$ for each of 2^{32} possible values of $(s^2||s^{14})$, where $s^2 = s^2(3,9)$, and set them to zero. For all 2^{20} possible values of s^1, encrypt s^1 one round to obtain s^2 and update the value $N1[s^2, s^{14}] = N1[s^2, s^{14}] + N0[s^1, s^{14}]$ for all 2^{24} values of s^{14}.

4. Guess the nibble $TK^2(3)$. Then, allocate a counter $N2[s^3, s^{14}]$ for each of 2^{28} possible values of $(s^3||s^{14})$, where $s^3 = s^3(4)$, and set them to zero. For all 2^8 possible values of s^2, encrypt s^2 one round to obtain s^3 and update the value $N2[s^3, s^{14}] = N1[s^3, s^{14}] + N1[s^2, s^{14}]$ for all 2^{24} values of s^{14}.

5. Guess the 2 nibbles $TK^{14}(3,4)$. Then, allocate a counter $N3[s^3, s^{13}]$ for each of 2^{12} possible values of $(s^3||s^{13})$, where $s^{13} = s^{13}(3,4,15)$, and set them to zero. For all 2^{24} possible values of s^{14}, decrypt s^{14} one round to obtain $s^{13}$and update the value $N3[s^3, s^{13}] = N3[s^3, s^{13}] + N2[s^3, s^{14}]$ for all 2^4 values of s^3.

6. Guess the nibble $TK^{13}(0)$. Then, allocate a counter$N4[s^3, s^{12}]$ for each of 2^{12} possible values of$(s^3||s^{12})$, where $s^{12} = s^{12}(0,12)$, and set them to zero. For all 2^{12} possible values of s^{13}, decrypt s^{13} two rounds to obtain s^{11} and update the value $N4[s^3, s^{11}] = N4[s^3, s^{11}] + N3[s^3, s^{13}]$ for all 2^4 values of s^3. The counter $N4[s^3, s^{11}]$ is then taken as the desired counter $V[z]$, where z is the 3-byte data value $s^3||s^{11}$.

7. Compute the statistical value

$$T = \frac{N * 2^4}{1 - 2^{-4}} \sum_{S^{11}=4}^{2^4-1} \sum_{S^3=0}^{2^4-1} \left(\frac{N_4[S^3, S^{11}]}{N} - \frac{1}{2^4} \right)^2 .$$

   If $T < t$, the guessed key is taken as a possible candidate.

8. Do exhaustive search for all keys that correspond to the guessed subkey bits.

Attack complexity

The memory complexity of the attack is $2^{44} \times 3$ bytes, which is dominated by step 2. The time complexity of step 1 and 2 is equal to the number of needed pairs of plaintext-ciphertext N. The time complexity of steps between 3 and 7 depends on the number of accesses to the memory. The time complexity for each round can be derived as follows.

Step 3: $2^8 \times 2^{20} \times 2^{24} = 2^{52}$ memory accesses needed, since we have to guess 8 bits for TK^1, and for 2^{20} values encrypt s^1 one round and update $N1$ for 2^{24} times.

Step 4: $2^{(8+4)} \times 2^8 \times 2^{24} = 2^{44}$ memory accesses needed, since we have to guess 4 bits for TK^2, and for 2^8 values encrypt s^2 one round and update $N2$ for 2^{24} times.

Step 5: $2^{(12+8)} \times 2^{24} \times 2^4 = 2^{48}$ memory accesses needed, since we have to guess 8 bits for TK^{14}, and for 2^{24} values decrypt s^{14} one round and update $N3$ for 2^4 times.

Step 6: $2^{(20+4)} \times 2^{12} \times 2^4 = 2^{40}$ memory accesses needed, since we have to guess 4 bits for TK^{13}, and for 2^{12} values decrypts^{13} two rounds and update $N3$for 2^4 times.

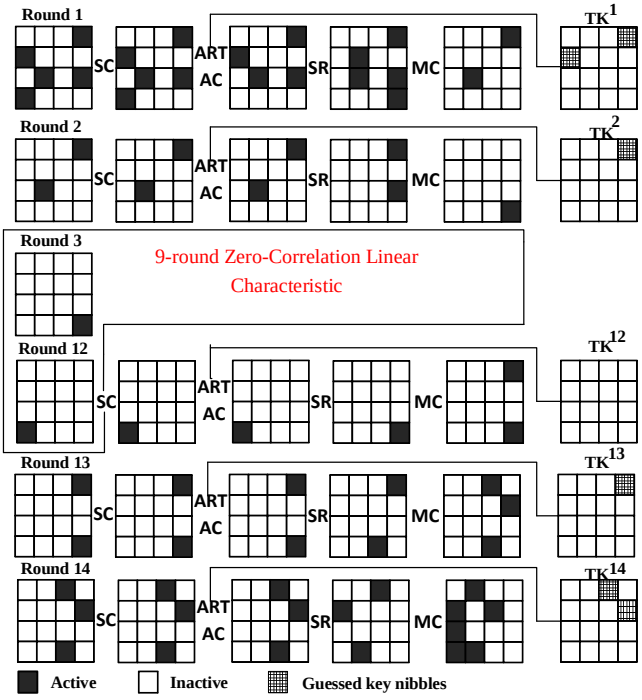

Figure 10: 14-round key recovery with zero-correlation linear attack for SKINNY-64 with 64-bit tweakey and SKINNY-128 with 128-bit tweakey

Step 7: $2^{24} \times 3 \times 2^8 = 2^{33}$ times of reading the 3-byte memory.

In addition, step 8 needs $2^{64}.\beta$ full encryptions, β is the probability of surviving a wrong subkey. The time complexity of step 8 and the success probability are determined based on the error probability type I(α), error probability type II(β), and the number of required plaintexts-ciphertext pairs.

There is a trade-off between the time complexity and the data complexity of the attack, as presented in Table 25.

Table 25: Time and data complexity for different values of α and β for SKINNY-64 with 64-bit tweakey

α	β	P_S	Time complexity	Data complexity
$2^{-7.3}$	2^{-1}	0.99	2^{63}	$2^{62.95}$
$2^{-3.3}$	2^{-1}	0.89	2^{63}	$2^{62.37}$
$2^{-3.3}$	2^{-3}	0.89	2^{61}	$2^{63.30}$
$2^{-2.7}$	2^{-1}	0.84	2^{63}	$2^{62.15}$
$2^{-2.7}$	2^{-2}	0.84	2^{62}	$2^{63.77}$
$2^{-1.7}$	2^{-1}	0.69	2^{63}	$2^{61.34}$
$2^{-1.7}$	2^{-5}	0.69	2^{59}	$2^{63.58}$
$2^{-1.7}$	2^{-2}	0.69	2^{62}	$2^{62.58}$