Related-Key Impossible-Differential Attack on Reduced-Round SKINNY

Ralph Ankele[1], Subhadeep Banik[2], Avik Chakraborti[3], Eik List[4],
Florian Mendel[5], Siang Meng Sim[2], Gaoli Wang[6]

[1] Royal Holloway University of London, United Kingdom.
ralph.ankele.2015@rhul.ac.uk
[2] Nanyang Technological University, Singapore.
{bsubhadeep@,ssim011@e.}ntu.edu.sg
[3] Indian Statistical Institute, Kolkata, India. avikchkrbrti@gmail.com
[4] Bauhaus-Universität Weimar, Germany. eik.list@uni-weimar.de
[5] Graz University of Technology, Austria. florian.mendel@iaik.tugraz.at
[6] East China Normal University, China. glwang@sei.ecnu.edu.cn

Abstract. At CRYPTO'16, Beierle et al. presented SKINNY, a family of lightweight tweakable block ciphers intended to compete with SIMON. SKINNY can be implemented efficiently in both soft- and hardware, possesses a Substitution-Permutation-Network structure, and supports block sizes of 64 and 128 bits as well as key and tweak sizes of 64, 128, 192, and 256 bits. This paper outlines a related-key impossible-differential attack on 21 and 22 rounds of SKINNY-64/128.

Keywords: Symmetric cryptography · cryptanalysis · tweakable block cipher · impossible differential · lightweight cryptography.

1 Introduction

SKINNY is a family of lightweight tweakable block ciphers recently proposed at CRYPTO 2016 by Beierle et al. [3]. Its goal was to design a cipher that could be implemented highly efficiently on both soft- and hardware platforms, with performance comparable or better than the SIMON and SPECK families of block ciphers [1]. Like the NSA designs SIMON and SPECK, SKINNY supports a wide range of block sizes and tweak/key sizes – however, in contrast to the And-RX and Add-RX based NSA proposals, SKINNY should base on the better understood Substitution-Permutation-Network approach.

SKINNY offers a large security margin within the number of rounds for each member of the SKINNY family. The designers show that the currently best known attacks approach close to half of the number of rounds of the cipher. To motivate third-party cryptanalysis, the designers of SKINNY recently announced a cryptanalysis competition [2] for SKINNY-64/128 and SKINNY-128/128 with the obvious challenge of attacking more rounds than the preliminary analysis, concerning both the single- and related-key models.

Related Work. Liu *et al.* [7] analysed SKINNY in the related-tweakey model, showing impossible-differential and rectangle attacks for 18, 22, 27 rounds of SKINNY-n/n, SKINNY-n/2n and SKINNY-n/3n, respectively. Tolba *et al.* [9] showed impossible-differential attacks for 18, 20, 22 rounds of SKINNY-n/n, SKINNY-n/2n and SKINNY-n/3n, respectively. Moreover, Sadeghi *et al.* [8] studied related-tweakey impossible-differential and zero-correlation linear characteristics. In comparison our proposed 22 round related-tweakey impossible-differential attacks have the lowest time complexity so far.

Contributions and Outline. In this paper, we propose an impossible-differential attack on SKINNY-64/128 reduced to 21 rounds in the related-key model which we then extend to 22 rounds. The attack uses an 11-round impossible differential trail, to which six and four rounds can be added to the beginning and end, respectively, in order to obtain a 21-round attack. Later we show that another round can be appended in the end to give a 22 round attack.
The paper is organized in the following manner: In Section 2, we give a brief introduction to the SKINNY family of block ciphers. In Section 3, we detail the attack on SKINNY and provide time and memory complexities. Finally, Section 4 concludes the paper.

2 Description of SKINNY

Each round of SKINNY consists of the operations SubCells, AddRoundConstants, AddRoundTweakey, ShiftRows, and MixColumns. The round operations are schematically illustrated in Figure 1. A cell represents a 4-bit value in SKINNY-64/* and an 8-bit value in SKINNY-128/*.

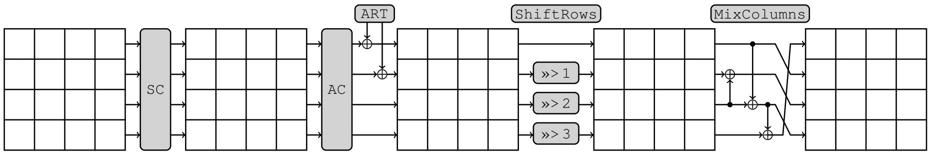

Fig. 1: Round function of SKINNY.

We concentrate on SKINNY-64/128, which has a block size of 64 bits and a tweakey size of 128 bits. The data is arranged nibble-by-nibble in a row-wise fashion in a 4 × 4-matrix.

SubCells (SC) substitutes each nibble x by $S(x)$, which is given below.

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

AddRoundConstants (AC) adds LFSR-based round constants to the Cells 0, 4, and 8 of the state.

AddRoundTweakey (ART) adds the round tweakey to the first two state rows.

ShiftRows (SR) rotates the i^{th} row, for $0 \leq i \leq 3$, by i positions to the right.

MixColumns (MC) multiplies each column of the state by a matrix M:

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

SKINNY-64/128 recommends 36 executions of the round function to get the final ciphertext.

Tweakey Schedule. The tweakey schedule of SKINNY, as illustrated in Figure 2, follows the TWEAKEY framework [5]. As a major contrast to previous TWEAKEY designs Deoxys-BC and Joltik-BC, SKINNY employs a significantly more lightweight strategy. In each round, only the both topmost rows of each tweakey word are extracted, and XORed to the state. An additional round-dependent constant is also XORed to the state to prevent attacks from symmetry, such as slide attacks, and complicate subspace cryptanalysis.

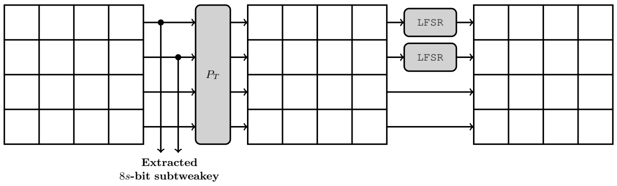

Fig. 2: Tweakey schedule of SKINNY.

The 128-bit tweakey is arranged in two 64-bit tweakey words, represented by 4×4 matrices TK_1 and TK_2. As mentioned, the arrangement is row-wise and nibble-by-nibble. In each round, the tweakey words are updated by a cell permutation P_T that ensures that the two bottom rows of a tweakey word in a certain round are exchanged with the two top rows in the tweakey word in the subsequent round. The permutation is given as:

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

The permutation P_T has a period of 16, as visualized in Fig. 6 in the appendix. Moreover, each individual cell in the two topmost rows of the tweakey word TK_2

is transformed by a 4-bit LFSR to thwart iterative differentials; TK_1 employs no LFSR transformation. The LFSR based transformation L is given by

$$L(x_3, x_2, x_1, x_0) := (x_2, x_1, x_0, x_3 \oplus x_2),$$

where x_3, x_2, x_1, x_0 represent the individual bits (x_0 represents the LSB of the cell) of every tweakey nibble. To avoid confusion, the update equation for the tweak cells can be written explicitly as:

$$TK_1^{r+1}[i] = \begin{cases} TK_1^r[P[i]] & \text{for } 0 \leq i \leq 15, \end{cases}$$
$$TK_2^{r+1}[i] = \begin{cases} L(TK_2^r[P[i]]) & \text{if } 0 \leq i \leq 7, \\ TK_2^r[P[i]] & \text{otherwise.} \end{cases}$$

where $TK_a^r[i]$ represents the i^{th} nibble of TK_a $(a = 1, 2)$ in round r. Note that the r^{th}-round tweakey is given by $K^r = TK_1^r[i] \oplus TK_2^r[i]$, for $0 \leq i \leq 7$.

3 Impossible Differential attack in the Related Key mode

Impossible differential attacks were independently introduced by Biham *et al.* [4] and Knudsen [6]. They are widely used as an important cryptanalytic technique. The attack starts with finding an input difference that can never result in an output difference, which makes up an impossible differential. By adding rounds before and/or after the impossible differential, one can collect pairs with certain plaintext and ciphertext differences. If there exists a pair that meets the input and output values of the impossible differential under some subkey, these subkeys must be wrong. In this way, we can filter as many wrong keys as possible and exhaustively search the rest of the keys.

Notations. Before proceeding let us state a few notations that we will use in the attack description:

K^r represents the r^{th} round key. This is equal to $TK_1^r \oplus TK_2^r$, the first and second tweakey blocks. Similarly $k^r[i] = tk_1^r[i] \oplus tk_2^r[i]$ represents the individual i^{th} tweakey nibble in round r.

A^r represents the internal state before SC in round r, and $A^r[i]$ represents the i^{th} nibble of A^r.

B^r represents the internal state after SC in round r, and $B^r[i]$ represents the i^{th} nibble of B^r.

C^r represents the internal state after AT in round r, and $C^r[i]$ represents the i^{th} nibble of C^r.

D^r represents the internal state after SR in round r, and $D^r[i]$ represents the i^{th} nibble of D^r.

E^r represents the internal state after MC in round r, and $E^r[i]$ represents the i^{th} nibble of E^r. Incidentally, we have $E^r = A^{r+1}$.

L^t represents the t-times composition of LFSR function L.

$\overline{X}$ represents the corresponding variable X under the related-key difference encryption flow.

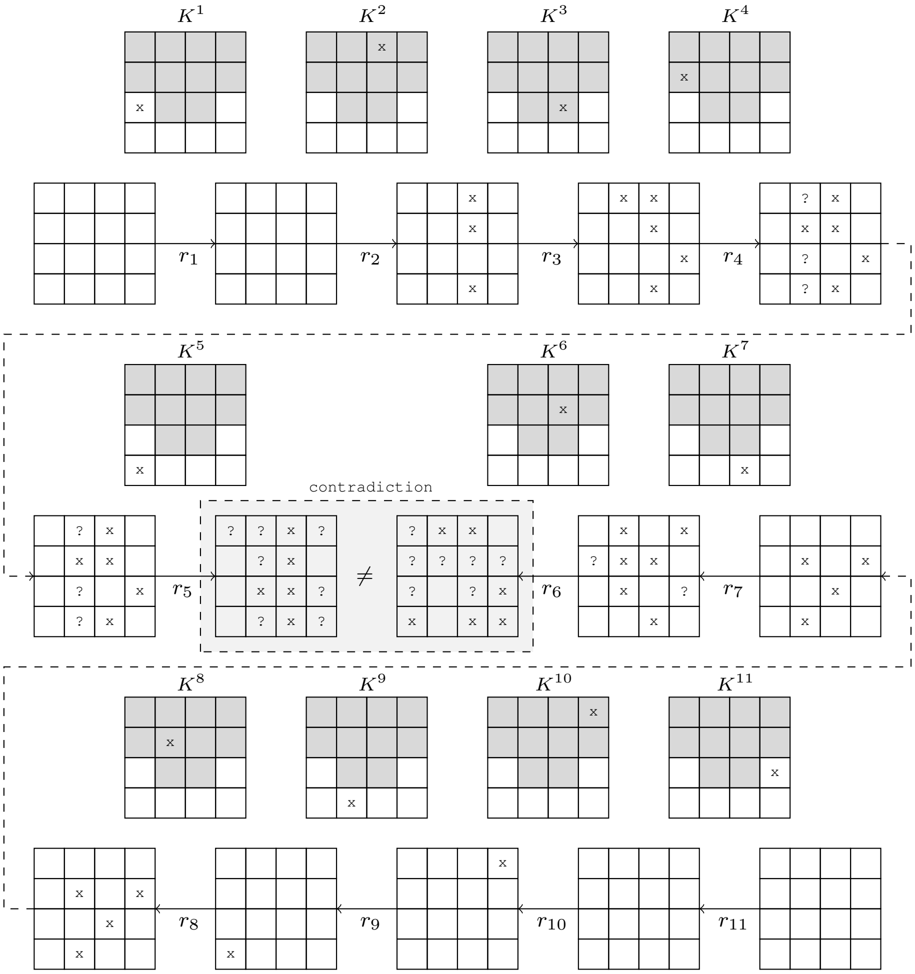

Fig. 3: Related-Key Impossible Differential Trail for 11 round SKINNY 64/128 (grey cells are the key, white cells are the tweak)

Impossible Differential Trail. Fig. 3 presents the 11-round related key differential trail that we use in this paper. We introduce a nibble difference in cell 8 of the combined tweakey. Since the initial difference is in cell number 8, *i.e.* in one of the bottom two rows in the tweakey, it does not affect the state in the first round, and will be introduced in the state from the second round onwards. Similarly in the backward trail, the difference in the 11^{th} round tweakey appears in cell 11 (also situated in one of the bottom two rows), due to which we get an extra round in the backward direction too.

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

Proof. The above fact can be deduced by analyzing the Differential Distribution Table (DDT) of the S-box S as illustrated in Table 1 in the appendix. The average can be calculated as $\frac{1}{225}\cdot\sum_{\Delta_i,\Delta_o\neq 0} DDT(\Delta_i,\Delta_o)\approx 1$. A similar exercise can be done for the inverse S-box yielding the same result.

Lemma 2. *For random values of x and $\Delta_i, \Delta_o \neq 0$, the equation $S(x+\Delta_i)+S(x)=\Delta_o$ holds with probability around 2^{-4}.*

Proof. The above fact can also be deduced by analyzing the Differential Distribution Table (DDT) of the S-box S as illustrated in Table 1 in the appendix. The probability can be calculated as (let $\mathsf{Pr}[(x,\delta_i,\delta_o)$ denote the probablility that the equation is satisfied for the triplet x,δ_i,δ_o)

$$\begin{aligned}\mathsf{Pr}[(x,\Delta_i,\Delta_o)] &= \sum_{\delta_i,\delta_o\neq 0}\mathsf{Pr}[(x,\delta_i,\delta_o)|\Delta_i=\delta_i,\Delta_o=\delta_o]\mathsf{Pr}[\Delta_i=\delta_i,\Delta_o=\delta_o]\\ &= \frac{1}{225}\cdot\sum_{\Delta_i,\Delta_o\neq 0} DDT(\Delta_i,\Delta_o)\cdot 2^{-4}\approx 2^{-4}\end{aligned}$$

Attack on 21 rounds. The impossible differential trail described in Fig. 3 can be extended by 6 and 4 rounds in backward and forward direction as will be explained in the following two lemmas.

Lemma 3. *It is possible to find plaintext pairs $P,\overline{P}$ and related-tweakey pairs $K,\overline{K}$ such that if the tweakey pairs differ only in nibble-position* 11*, then there is no difference in the internal state after executing 6 rounds of* SKINNY*-64/128 with the plaintext-tweakey pairs (P,K) and $(\overline{P},\overline{K})$.*

Proof. We will proceed to demonstrate how the required plaintext and tweakey pairs are generated. We choose the nibble at position 11 to introduce the initial difference because after completing 6 rounds, the difference is shuffled to the nibble-position 8 of the roundkey and it coincides with the beginning of the impossible differential trail, shown in Fig. 3. To begin with, it can be seen that the AddRoundTweakey in the first round can be pushed to after the MixColumns operation, by changing the first round key to $\mathsf{Lin}(K_1)$ where $\mathsf{Lin}=$ MC $\circ$ SR represents the linear layer (please refer to Fig. 4).

$$\mathsf{Lin}(K^1) = \begin{bmatrix} k^1[0] & k^1[1] & k^1[2] & k^1[3] \\ k^1[0] & k^1[1] & k^1[2] & k^1[3] \\ k^1[7] & k^1[4] & k^1[5] & k^1[6] \\ k^1[0] & k^1[1] & k^1[2] & k^1[3] \end{bmatrix}$$

Furthermore, the initial difference between $K = TK_1^1 + TK_2^1$ and $\overline{K} = \overline{TK_1^1} + \overline{TK_2^1}$ can be selected in a specific form, so that in round 6, the tweakey difference is 0. Let us denote $\delta_1 = tk_1^1[11] + \overline{tk_1^1}[11]$ and $\delta_2 = tk_2^1[11] + \overline{tk_2^1}[11]$. In round 6 the difference will appear in location 0 of the roundkey and so we want:

$$k^6[0] + \overline{k^6}[0] = tk_1^6[0] + \overline{tk_1^6}[0] + tk_2^6[0] + \overline{tk_2^6}[0]$$

$$= tk_1^1[11] + \overline{tk_1^1}[11] + L^3(\ tk_2^1[11]\ ) + L^3(\ \overline{tk_2^1}[11]\ )$$

$$= \delta_1 + L^3(\delta_2) = 0$$

So if the attacker chooses δ_1, δ_2 satisfying the equation $\delta_1 + L^3(\delta_2) = 0$, then there is no difference introduced via the roundkey addition in round 6. The attacker should thus follow the following steps:

1. Take any Plaintext P and compute the state after the first round MixColumns, *i.e.* E^1.
2. Take any 3 nibble difference $\varDelta_1, \varDelta_3, \varDelta_4$, to construct $\overline{E^1}$ such that

$$E^1 \oplus \overline{E^1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \varDelta_1 & 0 & \varDelta_2 \\ \varDelta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varDelta_4 \end{bmatrix}$$

   The value of $\varDelta_2$ will be determined shortly. The attacker can recover $\overline{P}$ by inverting the MC, SR, AC and SC layers on $\overline{E^1}$.

3. The attacker chooses the difference α in cell 14 of E^2. She then calculates $k^1[1], k^1[3], k^1[7]$, so that

$$B^2 \oplus \overline{B^2} = \mathsf{Lin}^{-1}(E^2) \oplus \mathsf{Lin}^{-1}(\overline{E^2})$$
$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta \\ \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha \end{bmatrix}$$

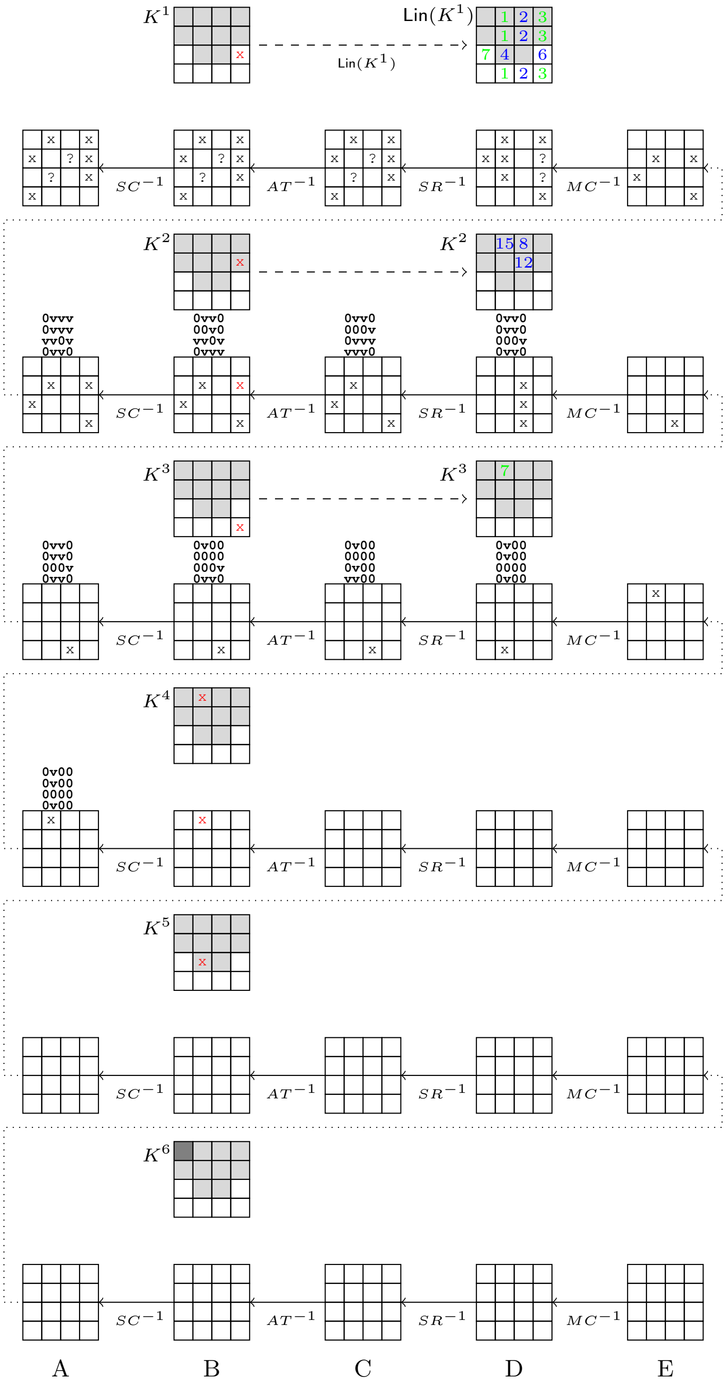

Fig. 4: Trail for the forward 6 rounds (the values of active nibbles in red are functions of δ_1, δ_2, grey cells are the key, white cells are the tweak, the dark gray cell visualises the tweakey cancelation)

For example $k^1[1]$ is a solution of the equation:

$$S(E^1[5] \oplus k^1[1]) \oplus S(E^1[5] \oplus \varDelta_1 \oplus k^1[1]) = \alpha$$

Note that the above equation, according to Lemma 1 has one solution on average.

4. β needs to be equal to $k^2[7] \oplus \overline{k^2}[7] = tk_1^2[7] + tk_2^2[7] + \overline{tk_1^2}[7] + \overline{tk_2^2}[7]$. This is equal to $tk_1^1[11] + L(\ tk_2^2[11]\ ) + \overline{tk_1^1}[11] + L(\ \overline{tk_2^2}[7]\ ) = \delta_1 \oplus L(\delta_2)$.
So the attacker chooses δ_1 and δ_2 satisfying $\delta_1 + L^3(\delta_2) = 0$ and calculates $\beta = \delta_1 \oplus L(\delta_2)$. $\varDelta_2$ can then be determined as a solution of the equation:
$$S(E^1[7] \oplus k^1[3]) \oplus S(E^1[7] \oplus \varDelta_2 \oplus k^1[3]) = \beta \qquad (1)$$
Again by Lemma 1, there exists on average one solution of the above equation. The attacker now has the values of $\varDelta_1, \varDelta_2, \varDelta_3, \varDelta_4$ and so he can compute $E^1, \overline{E^1}$ and hence $P, \overline{P}$.
5. However, the attacker still needs that in round 4 the active nibble in cell 1 of B^4 be equal to $\delta_1 \oplus L^2(\delta_2)$ to make all the state cells inactive in C^4, D^4, E^4.
6. For that the attacker needs to guess three additional key values in round 1 (*i.e.* $k^1[2], k^1[4], k^1[6]$) and three additional key values in round 2 (*i.e.* $k^2[1] = tk_1^1[15] + L(tk_2^1[15])$, $k^2[2] = tk_1^1[8] + L(tk_2^1[8])$, $k^2[6] = tk_1^1[12] + L(tk_2^1[12])$). If the attacker can guess these values, then he knows the actual values (marked with v) of the state cells for the plaintext pair $P, \overline{P}$ as opposed to only differences (marked by 0) in both Fig. 4 and Fig. 5.
7. Guessing the above tweakey nibbles enable the attacker to calculate the value of $B^3[1]$. She then calculates $k^3[1] = tk_1^1[7] \oplus L(tk_2^1[7])$ as follows. Since $D^3[1] = B^3[1] \oplus k^3[1]$ we should have:
$$S(D^3[1] \oplus D^3[9] \oplus D^3[13]) \oplus S(D^3[1] \oplus D^3[9] \oplus \overline{D^3}[13]) = \delta_1 \oplus L^2(\delta_2)$$
Since the knowledge of the guessed key nibbles already allow the attacker to calculate $D^3[9], D^3[13], \overline{D^3}[13]$, $k^3[1] = tk_1^1[7] \oplus L(tk_2^1[7])$ is the solution to the above equation. Again Lemma 1 guarantees one solution on average. Since the attacker has already determined $k^1[7] = tk_1^1[7] \oplus tk_2^1[7]$, this gives her the values of $tk_1^1[7]$ and $tk_2^1[7]$ uniquely.
8. This guarantees that there are no more active nibbles after round 4. The key difference does not add to the state in round 5, and due to the fact that $\delta_1 + L^3(\delta_2) = 0$, the tweak difference becomes 0 in round 6.

Thus by guessing 6 key nibbles, and calculating 3 key nibbles we can construct $P, \overline{P}$ and $K, \overline{K}$ so that the internal state after 6 rounds has no active nibbles.

Lemma 4. *Given $C, \overline{C}$ as the two output ciphertexts after querying plaintext-tweakey pairs (P, K) and $\overline{P}, \overline{K}$ as described above, to a 21-round* SKINNY*-64/128 encryption oracle. Then for a fraction 2^{-40} of the ciphertext pairs, it is possible to construct a backward trail for round 21 to round 18 by guessing intermediate tweakey nibbles so that there are no active nibbles in the internal state at the end of round 17.*

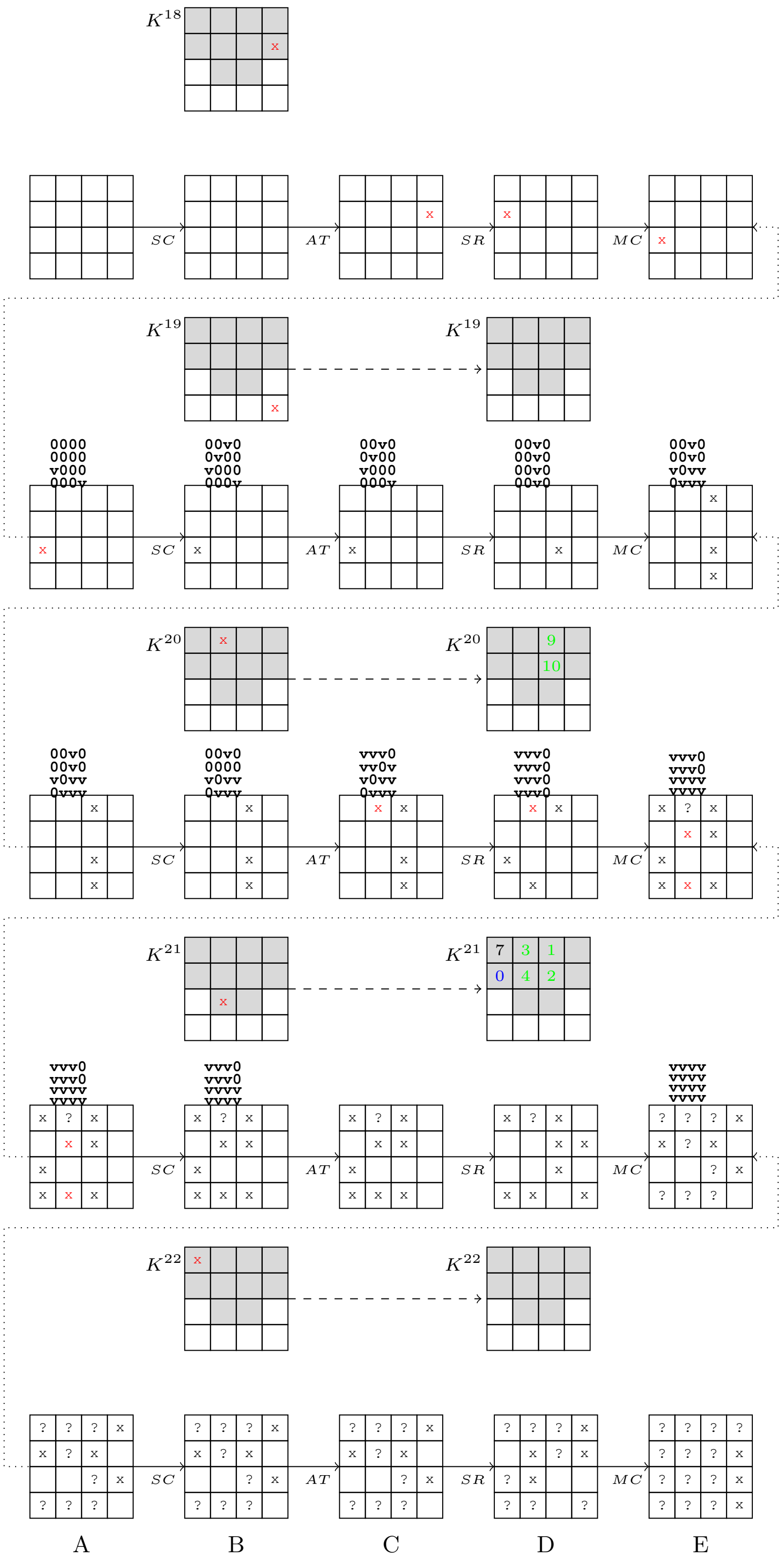

Fig. 5: Trail for the backward 5 rounds (the values of active nibbles in red are functions of δ_1, δ_2, grey cells are the key, white cells are the tweak)

Proof. The attacker starts working backward from the ciphertext pairs $C, \overline{C}$ and proceeds as follows (illustrated in Fig. 5):

1. The attacker rejects ciphertext pairs which do not have 7 inactive cells (*i.e.* in cell positions 3, 4, 5, 8, 9, 11, 14) after peeling off the last MixColumns layer (*i.e.* D^{21}). Thus a fraction of 2^{-28} pairs are filtered after this stage.
2. Furthermore, the attacker rejects ciphertext pairs which do not have the difference $\delta_1 \oplus L^{10}(\delta_2)$ in cell 13 of A^{21}, *i.e.* reject if $A^{21}[13] \oplus \overline{A^{21}}[13] \neq \delta_1 \oplus L^{10}(\delta_2)$. Since calculating this cell does not require any key guess, the attacker can do this filtering. Thus a further 2^{-4} pairs are filtered after this stage.
3. Since the last two rows of the state are not affected by the tweakey addition and since $tk_1^1[7], tk_2^1[7]$ are already known, she can calculate the actual values in the cells 0, 8, 12 in A^{21} for the ciphertext pairs. These have to be equal since they are the output of the 20^{th} round MixColumns on column 0 with only one active nibble in its input. If the active nibble in cell 8 and 12 are not equal, the attacker can reject the pair. This adds another filter of probability 2^{-4}
4. Continuing the previous point, since the actual values in cell 0 in A^{21} for the ciphertext pairs were already calculated in the previous step, she now checks if the value of the active nibble in cell 0 is equal to the active nibble in 8, 12. If it is not equal to the active nibble 8, 12 she can reject it. This adds another filter of probability 2^{-4}.
5. The attacker determines $k^{21}[5] = tk_1^1[4] \oplus L^{10}(tk_2^1[4])$ so that the active nibble in cell 5 of A^{21} is $\delta_1 \oplus L^{10}(\delta_2)$. Since $A^{21}[5] = S^{-1}(\ k^{21}[5] \oplus C^{21}[5]\ )$, $k^{21}[5]$ is a solution to the equation below:

$$S^{-1}(\ k^{21}[5] \oplus C^{21}[5]\ ) \oplus S^{-1}(\ k^{21}[5] \oplus \overline{C^{21}}[5]\ ) = \delta_1 \oplus L^{10}(\delta_2)$$

6. The attacker determines $k^{21}[2] = tk_1^1[1] \oplus L^{10}(tk_2^1[1])$ and $k^{21}[6] = tk_1^1[2] \oplus L^{10}(tk_2^1[2])$ so that the active nibble in cell 2,6 of A^{21} is equal to the active nibble in cell 14. Again this works, since they are output of the 20^{th} round MixColumns on column 2 with only one active nibble in its input.
7. Additionally the attacker guesses $k^{21}[4] = tk_1^1[0] \oplus L^{10}(tk_2^1[0])$. This enables the attacker to compute the actual values for the entire 0^{th} column of A^{21} and hence D^{20} after applying the inverse MixColumns.
8. The value of the active nibble in cell 10 of A^{20} is given as:

$$\begin{aligned} A^{20}[10] \oplus \overline{A^{20}}[10] &= S^{-1}(\ B^{20}[10]\ ) \oplus S^{-1}(\ \overline{B^{20}}[10]\ ) \\ &= S^{-1}(\ D^{20}[8]\ ) \oplus S^{-1}(\ \overline{D^{20}}[8]\ ) = \eta \end{aligned} \tag{2}$$

Since the 0^{th} column of D^{20} is known, the attacker can calculate η. This must be equal to active nibble in cell 14 of A^{20}, (since they are output of the 19^{th} round MixColumns with one active input nibble). This is given as:

$$\begin{aligned} A^{20}[14] \oplus \overline{A^{20}}[14] &= S^{-1}(\ D^{20}[13]\ ) \oplus S^{-1}(\ \overline{D^{20}}[13]) \\ &= S^{-1}(\ A^{21}[1] \oplus A^{21}[13]\ ) \oplus S^{-1}(\ \overline{A^{21}}[1] \oplus \overline{A^{21}}[13]\ ) \end{aligned} \tag{3}$$

$A^{21}[1] = S^{-1}(\ C^{21}[1] \oplus k^{21}[1]\ )$. Similarly $\overline{A^{21}}[1] = S^{-1}(\ \overline{C^{21}}[1] \oplus k^{21}[1]\ )$. By calculating Equation (2) and (3) she can solve for $k^{21}[1] = tk_1^1[3] \oplus L^{10}(tk_2^1[3])$. One solution on average is guaranteed by Lemma 1.

9. The values $tk_1^1[i] \oplus tk_2^1[i]$ $(i = 1, 2, 3, 4)$, were already determined during the calculation of the forward trail. So using this the attacker can determine the actual values $tk_1^1[i],\ tk_2^1[i]$ $(i = 1, 2, 3, 4)$.
10. The attacker calculates $k^{20}[2] = tk_1^1[9] \oplus L^{10}(tk_2^1[9])$ so that the active nibble in cell 2 in A^{20} is equal to the active value η in cell 10, 14 (since they are output of the 19^{th} round MixColumns with one input active nibble). This is done by solving

$$\eta = A^{20}[2] \oplus \overline{A^{20}}[2] = S^{-1}(C^{20}[2] \oplus k^{20}[2]) \oplus S^{-1}(\overline{C^{20}}[2] \oplus k^{20}[2]) \tag{4}$$

11. The final condition to be satisfied is that the active nibble in cell 8 of A^{19} has to be equal to $\delta_1 \oplus L^9(\delta_2) = \gamma$.

$$\begin{aligned} \gamma &= S^{-1}(D^{19}[10]) \oplus S^{-1}(\overline{D^{19}}[10]) \\ &= S^{-1}(A^{20}[6] \oplus A^{20}[14]) \oplus S^{-1}(\overline{A^{20}}[6] \oplus \overline{A^{20}}[14]) \end{aligned} \tag{5}$$

Note that $A^{20}[6] = S^{-1}(C^{20}[6] \oplus k^{20}[6])$. And since $\overline{A^{20}}[6] = A^{20}[6]$, solving Equation (5) helps to determine $k^{20}[6] = tk_1^1[10] \oplus L^{10}(tk_2^1[10])$.

Since in the steps 1, 2, 3, 4 a total of $2^{-28-4-4-4} = 2^{-40}$ ciphertext pairs are filtered the result follows.

3.1 Attack Algorithm

We now put together the findings of Lemma 3 and 4 into an attack procedure as follows:

1. The adversary chooses a random base plaintext P and gets the corresponding ciphertext C for (P, K)
2. She choose a fixed δ_1, δ_2 satisfying $\delta_1 \oplus L^3(\delta_2) = 0$.
3. For each nonzero difference $(\Delta_1, \Delta_3, \Delta_4)$ $((2^4 - 1)^3$ choices):
   - Choose α $(2^4 - 1$ choices) and so determine Δ_2.
   - With the value of $(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$, compute $\overline{P}$
   - Get the ciphertext $\overline{C}$ for $(\overline{P}, \overline{K})$

- If $C \oplus \overline{C}$ does not pass the 2^{-36} filter (Step 1, 2, 3 in Lemma 4) abort and start again.
- If they pass the filter: the adversary can guess 7 tweakey cells (2^{28} guesses) and calculate 17 key/tweak cells as follows:

#	Guessed	Rnd	Calculated	Rnd
1	$tk_1^1[i] \oplus tk_2^1[i]$ for $i = 2, 4, 6$	1		
2	$tk_1^1[i] \oplus L(tk_2^1[i])$ for $i = 8, 12, 15$	2		
3	$tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 0$	21		
4			$tk_1^1[i],\ tk_2^1[i]$ for $i = 7$	3
5			$tk_1^1[i],\ tk_2^1[i]$ for $i = 1, 2, 3, 4$	21
6			$tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 9, 10$	20

  The 17 tweakey nibbles used for elimination are therefore:

  (a) $tk_1^1[i],\ tk_2^1[i]$ for $i = 1, 2, 3, 4, 7$

  (b) $tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 9, 10$

  (c) $tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 0$

  (d) $tk_1^1[i] \oplus L(tk_2^1[i])$ for $i = 8, 12, 15$

  (e) $tk_1^1[i] \oplus tk_2^1[i]$ for $i = 6$
- A fraction of 2^{-4} tweakeys will fail the condition required in Step 4 of Lemma 4.
- Therefore the adversary has a set of $2^{28-4} = 2^{24}$ wrong key candidates.

The above procedure is repeated with 2^x chosen plaintexts till a single key solution remains for the 17 nibbles of the tweakey.

Complexity. For every plaintext: to begin the adversary has $(2^4 - 1)^3$ choices of differences, and for each α she has on average 1 values of $\varDelta_2$. Since there are $2^4 - 1$ choices of α there are on average $2^4 - 1$ choices of $\varDelta_2$. This makes a total of $(2^4 - 1)^4 \approx 2^{16}$ encryption calls. With 2^x such base plaintexts she has 2^{x+16} encryption calls. With probability 2^{-36} the adversary gets a workable ciphertext difference to process. Each such instance generates $2^{28-4} = 2^{24}$ key candidates (in 17 nibbles) for elimination. On average after $2^{x+16-36} = 2^{x-20}$ times, she gets to guess a set of 2^{24} tweakey candidates to eliminate.

$$\text{Time complexity} = \max(2^{x+16}, 2^{x-20+24}) \ = \ 2^{x+16}$$

The attacker gets wrong solutions for $2^{x-20+24} = 2^{x+4}$ incorrect solutions for 17 nibbles. To reduce the keyspace to 1 we need:

$$2^{17\times4}(1 - 2^{-17\times4})^{2^{x+4}} \approx 2^{17\times4}e^{-2^{x-64}} = 1$$

For this we need $x = 70$. So the total number of encryption calls to 21 round SKINNY-64/128 is $2^{70+16} = 2^{86}$.

3.2 2nd Attack

We will present another attack procedure that changes the way the related plaintext/tweakey pairs are constructed:

1. The adversary chooses the nibble values of the random base variable E^1 in all locations except nibbles indexed by 5, 7, 8, 15.
2. She chooses a fixed δ_1, δ_2 satisfying $\delta_1 \oplus L^3(\delta_2) = 0$.
3. For each choice of $(E^1[5], E^1[7], E^1[8], E^1[15])$ (2^{16} choices):
   - Calculate P by inverting 1st round operations.
   - Query the 21 round encryption oracle for P, K and $P, \overline{K}$

So for every choice of the base variable E^1 we have 2^{17} encryption calls. We can pair up related plaintext and tweakey pairs in the following way. For every plaintext P_i choose a plaintext P_j so that E^1 for P_i and P_j have a non zero difference in all the locations 5, 7, 8, 15. For every P_i there will exist $(2^4 - 1)^4 \approx 2^{15.6}$ values of P_j, and so $2^{16+15.6} = 2^{31.6}$ pairs to work with. The attack now proceeds as follows.

1. For each choice of P_i, P_j ($2^{31.6}$ choices):
   - Denote $P = P_i$ and $\overline{P} = P_j$.
   - The attacker can choose α and proceed with the steps of the above attack with one exception
   - She can no longer choose $\varDelta_2$ as in Step 4 of Lemma 3 since she has already chosen $P, \overline{P}, K, \overline{K}$.
   - With probability 2^{-4} (as per Lemma 2), the plaintext pair satisfies Equation (1) in Step 4 of Lemma 3 and he proceeds if it does. Else abort.
   - Get the ciphertext $\overline{C}$ for $(\overline{P}, \overline{K})$ and C for P, K.
   - If $C \oplus \overline{C}$ does not pass the 2^{-36} filter (Step 1, 2, 3 in Lemma 4) abort and start again.
   - If they pass the filter: the adversary can guess 7 tweakey cells (2^{28} guesses) and calculate 17 key/tweak cells as in previous attack.
   - A fraction of 2^{-4} tweakeys will fail the condition required in Step 4 of Lemma 4.
   - Therefore the adversary has a set of $2^{28-4} = 2^{24}$ wrong key candidates.

The above procedure is repeated with 2^x chosen plaintexts till a single key solution remains for the 17 nibbles of the tweakey.

Complexity. For every base value of E^1: the adversary makes 2^{17} encryption calls. Out of these she has has $2^{31.6}$ pairs to work with. For each pair, the attacker can then choose α in $2^4 - 1$ ways, which gives her around $2^{35.6}$ initial guesses for the forward key nibbles $k^1[1], k^1[3], k^1[7]$. Of which only a fraction 2^{-4} passes the filter in Equation (1) and so she has $2^{31.6}$ pairs to work with. In effect for every P_i, P_j there is only once choice of α on average going forward.
With 2^x such base plaintexts she has 2^{x+17} encryption calls but $2^{x+31.6}$ plaintext and hence ciphertext pairs. With probability 2^{-36} the adversary gets a workable ciphertext difference to process. Each such instance generates $2^{28-4} = 2^{24}$ key candidates (in 17 nibbles) for elimination. On average after $2^{x+31.6-36} = 2^{x-4.4}$ times, she gets to guess a set of 2^{24} tweakey candidates to eliminate.

$$\text{Time complexity} = \max(2^{x+17} \text{ encryptions }, 2^{x-4.4+24} \text{ guesses }) \;=\; 2^{x+19.6}$$

The attacker gets wrong solutions for $2^{x-4.4+24} = 2^{x+19.6}$ incorrect solutions for 17 nibbles. To reduce the keyspace to 1 we need:

$$2^{17\times 4}(1 - 2^{-17\times 4})^{2^{x+19.6}} \approx 2^{17\times 4} e^{-2^{x-48.4}} = 1$$

For this we need $x = 55$. So the total number of encryption calls to 21 round SKINNY-64/128 is $2^{55+17} = 2^{72}$ and total guesses is $2^{74.6}$.

3.3 Attacking 22 round SKINNY 64/128 in known tweak setting

The above attack can be extended to 22 round SKINNY 64/128 under the assumption that 48 of the 128 bits in the tweakey is a publicly known tweak block. In particular we assume that $tk_1^1[i], tk_2^1[i]$ for $i = 8, 11, 12, 13, 14, 15$ is reserved for the tweak. The remaining 80 bits constitute the secret key.
In that event he can add one round at the end. Having the 6 of the 8 cells in the lower half of the tweakey blocks known is helpful in the following way: from the the ciphertext (which is E^{22}) one can work back to compute E^{21}, if we guess $k^{22}[4,5]$ which is $tk_1^1[9,10] \oplus L^{11}(tk_2^1[9,10])$, so it allows easy stripping-off of the extra round that we have added. Thereafter the attack is almost the same as the previous attack, with the excetion that tweakey indices $i = 8, 11, 12, 13, 14, 15$ and their functions are known and need not be guesssed.

1. Generate $2^{31.6}$ plaintext/ciphertext pairs from every base choice of E^1 and 2^{17} encryption calls.
2. For each choice of P_i, P_j ($2^{31.6}$ choices):
   - Denote $P = P_i$ and $\overline{P} = P_j$.
   - The attacker can choose α and calculate $k^1[1], k^1[3], k^1[7]$ as per Step 3 of Lemma 3.
   - She can no longer choose Δ_2 as in Step 4 of Lemma 3 since she has already chosen $P, \overline{P}, K, \overline{K}$.
   - With probability 2^{-4}, the plaintext pair satisfies Equation (1) in Step 4 of Lemma 3 and he proceeds if it does. Else abort.

- As already outlined: she does not need to guess the round 2 tweakey nibbles in step 6 of Lemma 3: i.e. functions of $k^1[8, 12, 15]$ as these are in the lower half of the tweakey blocks and assumed to be known.
- Retrieve the ciphertext $\overline{C}$ for $(\overline{P}, \overline{K})$ and C for P, K.
- Guess $k^{22}[4, 5]$ which is $tk_1^1[9, 10] \oplus L^{11}(tk_2^1[9, 10])$ to peel off last round and get E_{21}.
- If $E_{21} \oplus \overline{E_{21}}$ does not pass the 2^{-36} filter (Step 1, 2, 3 in Lemma 4) abort and start again.
- After determining $k^{20}[2] = tk_1^1[9] \oplus L^{10}(tk_2^1[9])$ and $k^{20}[6] = tk_1^1[10] \oplus L^{10}(tk_2^1[10])$ in steps 10, 11 of Lemma 4, she can uniquely determine $tk_1^1[9, 10]$ as $tk_1^1[9, 10] \oplus L^{11}(tk_2^1[9, 10])$ is already guessed.
- If they pass the filter: the adversary can guess 6 tweakey cells (2^{24} guesses) and calculate 16 key cells as follows.

#	Guessed	Rnd	Calculated	Rnd
1	$tk_1^1[i] \oplus tk_2^1[i]$ for $i = 2, 4, 6$	1		
2	$tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 0$	21		
3	$tk_1^1[i] \oplus L^{11}(tk_2^1[i])$ for $i = 9, 10$	22		
4			$tk_1^1[i],\ tk_2^1[i]$ for $i = 7$	3
5			$tk_1^1[i],\ tk_2^1[i]$ for $i = 1, 2, 3, 4$	21
6			$tk_1^1[i],\ tk_2^1[i]$ for $i = 9, 10$	20

  The 16 tweakey nibbles used for elimination are therefore:

  (a) $tk_1^1[i],\ tk_2^1[i]$ for $i = 1, 2, 3, 4, 7, 9, 10$

  (b) $tk_1^1[i] \oplus L^{10}(tk_2^1[i])$ for $i = 0$

  (c) $tk_1^1[i] \oplus tk_2^1[i]$ for $i = 6$
- A fraction of 2^{-4} tweakeys will fail the condition required in Step 4 of Lemma 4.
- Therefore the adversary has a set of $2^{24-4} = 2^{20}$ wrong key candidates.

The above procedure is repeated with 2^x chosen plaintexts till a single key solution remains for the 12 nibbles of the tweakey.

Complexity. For every base value of E^1: the adversary makes 2^{17} encryption calls. Out of these she has has $2^{31.6}$ pairs to work with. For each pair, the attacker can then choose α in $2^4 - 1$ ways, which gives her around $2^{35.6}$ initial guesses for the forward key nibbles $k^1[1], k^1[3], k^1[7]$. Of which only a fraction 2^{-4} passes the filter in Equation (1) and so she has $2^{31.6}$ pairs to work with. In effect for every P_i, P_j there is only once choice of α on average going forward.
With 2^x such base plaintexts she has 2^{x+17} encryption calls but $2^{x+31.6}$ plaintext and hence ciphertext pairs. With probability 2^{-36} the adversary gets a workable ciphertext difference to process. Each such instance generates $2^{24-4} = 2^{20}$ key

candidates (in 16 nibbles) for elimination. On average after $2^{x+31.6-36} = 2^{x-4.4}$ times, she gets to guess a set of 2^{20} tweakey candidates to eliminate.

$$\text{Time complexity} = \max(2^{x+17} \text{ encryptions }, 2^{x-4.4+20} \text{ guesses }) \;=\; 2^{x+17}$$

The attacker gets wrong solutions for $2^{x-4.4+20} = 2^{x+15.6}$ incorrect solutions for 12 nibbles. To reduce the keyspace to 1 we need:

$$2^{16\times4}(1-2^{-16\times4})^{2^{x+15.6}} \approx 2^{16\times4}e^{-2^{x-48.4}} = 1$$

For this we need $x = 54$. So the total number of encryption calls to 22 round SKINNY-64/128 is $2^{54+17} = 2^{71}$.

4 Conclusion

In this paper, we outline a related-key impossible differential attack against 22 round SKINNY-64/128. Our attack is based on a 11 round impossible differential trail, to which we prepend and append 6 and 5 rounds before and after the trail respectively to get an attack on 22 rounds.

References

1. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404 (2013), http://eprint.iacr.org/
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: Cryptanalysis competition. https://sites.google.com/site/skinnycipher/cryptanalysis-competition (2016)
3. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016), http://dx.doi.org/10.1007/978-3-662-53008-5_5
4. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 1592, pp. 12–23. Springer (1999)
5. Jean, J., Nikolic, I., Peyrin, T.: Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT (2). Lecture Notes in Computer Science, vol. 8874, pp. 274–288 (2014)
6. Knudsen, L.: DEAL - A 128-bit Block Cipher. In: NIST AES Proposal (1998)
7. Liu, G., Ghosh, M., Ling, S.: Security analysis of skinny under related-tweakey settings. Cryptology ePrint Archive, Report 2016/1108 (2016), http://eprint.iacr.org/2016/1108
8. Sadeghi, S., Mohammadi, T., Bagheri, N.: Cryptanalysis of reduced round skinny block cipher. Cryptology ePrint Archive, Report 2016/1120 (2016), http://eprint.iacr.org/2016/1120
9. Tolba, M., Abdelkhalek, A., Youssef, A.M.: Impossible differential cryptanalysis of reduced-round skinny. Cryptology ePrint Archive, Report 2016/1115 (2016), http://eprint.iacr.org/2016/1115

A Permutation P_T

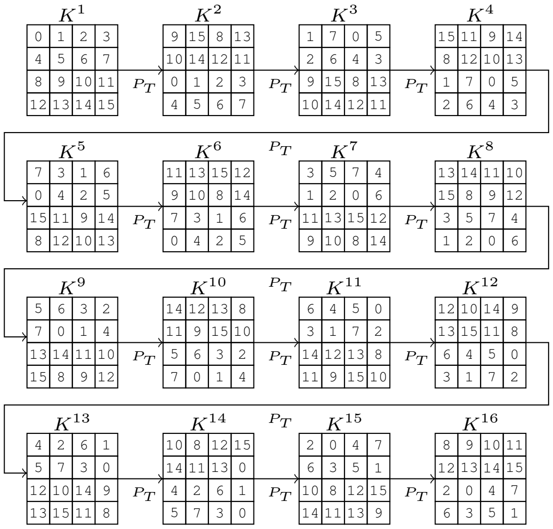

Fig. 6: The permutation P_T in the tweakey-schedule has a period of 16.

B Difference Distribution Table

Table 1: Differential Distribution Table

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.	4	4	4	4	.	.	.	.
2	.	4	.	4	.	4	4	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	2	2	2	2	2	2	2	2
4	.	.	4	.	.	.	2	2	.	.	.	4	2	2	.	.
5	.	.	4	.	.	.	2	2	.	.	4	.	2	2	.	.
6	.	2	.	2	2	.	.	2	2	.	2	.	.	2	2	.
7	.	2	.	2	2	.	.	2	.	2	.	2	2	.	.	2
8	.	.	.	.	4	4	.	.	.	.	.	.	2	2	2	2
9	.	.	.	.	4	4	.	.	.	.	.	.	2	2	2	2
a	.	.	.	.	.	4	4	.	2	2	2	2	.	.	.	.
b	.	4	.	4	.	.	.	.	.	.	.	.	2	2	2	2
c	.	.	4	.	.	.	2	2	4	.	.	.	.	.	2	2
d	.	.	4	.	.	.	2	2	.	4	.	.	.	.	2	2
e	.	2	.	2	2	.	.	2	.	2	.	2	.	2	2	.
f	.	2	.	2	2	.	.	2	2	.	2	.	2	.	.	2

C Related-Key Impossible Differential Attack

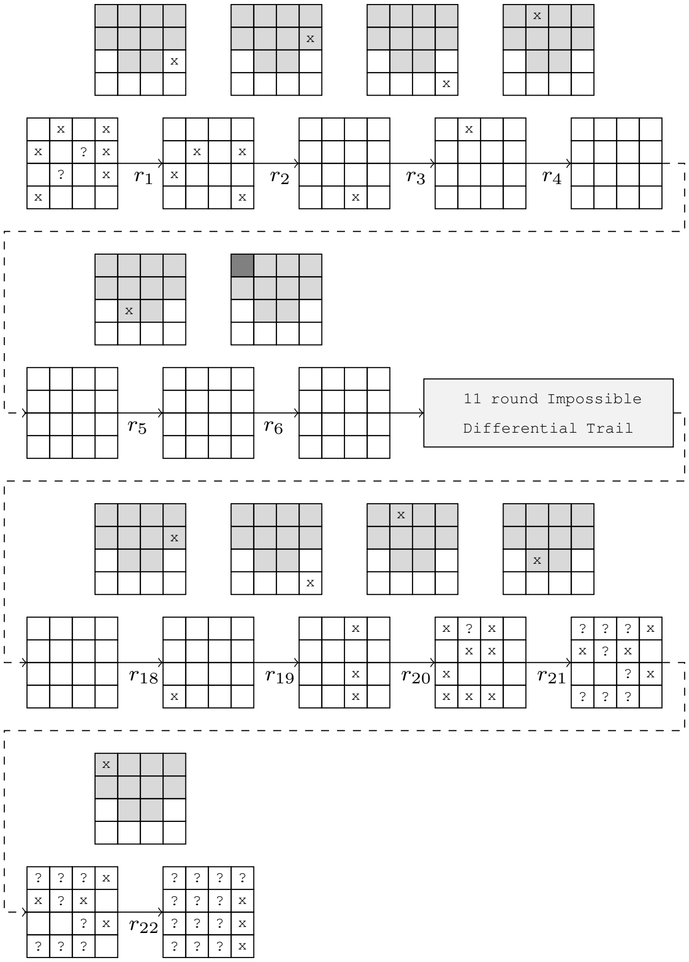

Fig. 7: Related-Key Impossible Differential Attack on 22 round SKINNY 64/128 (grey cells are the key, white cells are the tweak, the dark gray cell visualises the cancelation of the tweakeys)