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] NTT Secure Platform Laboratories, Japan. `chakraborti.avik@lab.ntt.co.jp`
[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 the NSA designs SIMON and SPECK. SKINNY can be implemented efficiently in both soft- and hardware and supports block sizes of 64 and 128 bits as well as tweakey sizes of 64, 128, 192 and 128, 256, 384 bits respectively. This paper presents a related-tweakey impossible-differential attack on up to 23 (out of 36) rounds of SKINNY-64/128 for different tweak sizes. All our attacks can be trivially extended to SKINNY-128/128.

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 is based 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.

Table 1: Summary of our attacks and comparison to existing cryptanalysis of SKINNY-64/128.

Instance	Rounds	Attack Type	Time	Data	Memory	Ref
SKINNY-64/128	20	Impossible	$2^{121.1}$	$2^{47.7}$	$2^{74.7}$	[9]
SKINNY-64/128	21	Rectangle	$2^{87.9}$	$2^{54.0}$	$2^{54.0}$	[7]
SKINNY-64/128	21	Impossible	$2^{71.4}$	$2^{71.4}$¶	$2^{68.0}$	Sect. 3.1
SKINNY-64/128	22	Rectangle	$2^{109.9}$	$2^{63.0}$	$2^{63.0}$	[7]
SKINNY-64/128	22	Impossible‖	$2^{71.6}$	$2^{71.4}$¶	$2^{64.0}$	Sect. 3.2
SKINNY-64/128	23	Impossible	$2^{124.2}$	$2^{62.5}$	2^{124}	[7]
SKINNY-64/128	23	Impossible‖	2^{79}	$2^{71.4}$¶	$2^{64.0}$	Sect. 3.3

Related Work. Recently and independent of our analysis Liu *et al.* [7] analyzed SKINNY in the related-tweakey model, showing impossible-differential and rectangle attacks on 19, 23, and 27 rounds of SKINNY-n/n, SKINNY-n/2n and SKINNY-n/3n, respectively. In [9], Tolba *et al.* showed impossible-differential attacks for 18, 20, 22 rounds of SKINNY-n/n, SKINNY-n/2n and SKINNY-n/3n, respectively. Additionally, Sadeghi *et al.* [8] studied related-tweakey impossible- differential and zero-correlation linear characteristics. In comparison to the other attacks, our 23-round related-tweakey impossible-differential attack on SKINNY-64/128 has the lowest time complexity so far. Table 1 summarizes our attacks and compares them to existing attacks on SKINNY-64/128.

Contributions and Outline. In this paper, we propose an impossible-differential attack on SKINNY-64/128 reduced to 23 rounds in the related-key model. The attack uses an 11-round impossible differential trail, to which six and four rounds can be added for obtaining a 21-round attack. Later, we show that another round can be appended leading to a 22-round attack, and even a 23-round attack.

The paper is organized as follows. 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.

[¶] The data complexity of our 21-round attack is beyond codebook. Our attack is more efficient than a full codebook attack in the case where SKINNY is used in a tweak-updating mode (i.e. where the tweak changes every time, but the key stays the same). This does not effect the 22/23 round attack as 48 bits of the tweakey are public (i.e. data complexity for full codebook would be 2^{64} from the state + 2^{48} from the tweak).

[‖] Our attack on 22/23 rounds uses the tweak against the recommendation of the SKINNY designers but still conform to the specification in [3].

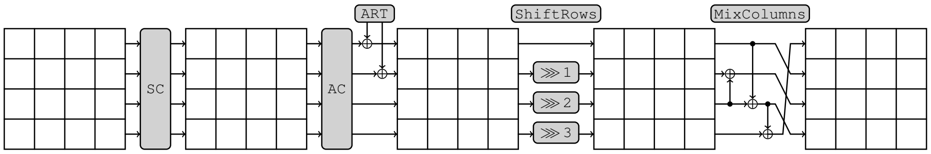

Fig. 1: Round function of SKINNY.

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 Fig. 1. A cell represents a 4-bit value in SKINNY-64/* and an 8-bit value in SKINNY-128/*.
We concentrate on SKINNY-64/128, which has a 64-bit block size and a 128-bit tweakey size. The data is arranged nibble-by-nibble in a row-wise fashion in a 4×4-matrix. SKINNY-64/128 recommends 36 rounds.

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 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:

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

Tweakey Schedule. The tweakey schedule of SKINNY, as illustrated in Fig. 2, follows the TWEAKEY framework [5]. In contrast to the previous TWEAKEY designs Deoxys-BC and Joltik-BC, SKINNY employs a significantly more lightweight strategy. In each round, only the two 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.
The 128-bit tweakey is arranged in two 64-bit tweakey words, represented by TK_1 and TK_2. In each round, the tweakey words are updated by a cell permutation

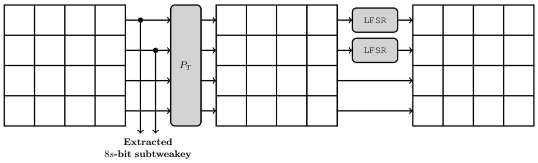

Fig. 2: Tweakey schedule of SKINNY.

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. 7 in the appendix. Moreover, each individual cell in the two topmost rows of TK_2 is transformed by a 4-bit LFSR to minimise the cancellation of differences from TK_1 and TK_2; TK_1 employs no LFSR transformation. The LFSR 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 of every tweakey nibble.

3 Related-Key Impossible-Differential Attack

Impossible-differential attacks were introduced independently 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. 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 filter as many wrong keys as possible and exhaustively search the rest of the keys.

Notations. Let us state a few notations that are used in the attack description:

K^r represents the r^{th} round key. This is equal to $TK_1^r \oplus TK_2^r$. 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.
B^r represents the internal state after SC in round r.
C^r represents the internal state after AT in round r.
D^r represents the internal state after SR in round r.
E^r represents the internal state after MC in round r. Furthermore, $E^r = A^{r+1}$.
L^t represents the t-times composition of LFSR function L.
$\overline{X}$ represents the corresponding variable X in the related-key setting.
$X[i]$ represents the i^{th} nibble of the corresponding variable X.

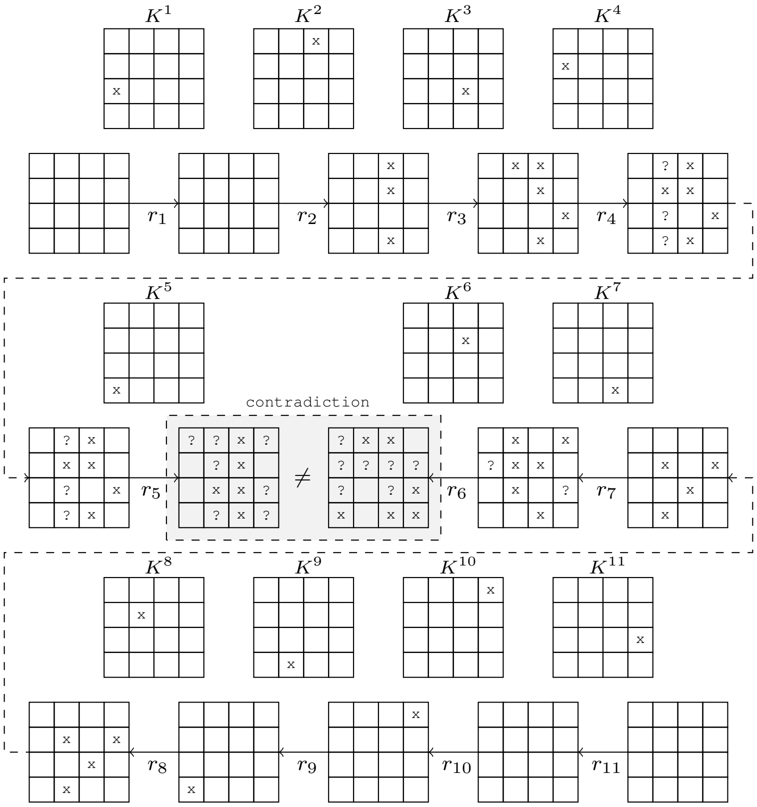

Fig. 3: Related-key impossible-differential trail over 11 rounds of SKINNY-64/128.

Impossible-Differential Trail. Fig. 3 presents the 11-round related-key differential trail that we use. We introduce a nibble difference in Cell 8 of the combined tweakey. Since the initial difference is in Cell 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 added to the state from the second round onwards. Similarly in the backward trail, the difference in the 11^{th} round-tweakey appears in Cell 11 (in a bottom row), due to which we get an extra round in the backward direction.

Lemma 1. *The equation* $S(x \oplus \Delta_i) \oplus S(x) = \Delta_o$ *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 2 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* $S(x \oplus \Delta_i) \oplus S(x) = \Delta_o$ *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 2 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 six and four 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 six rounds of* SKINNY-*64/128 with the plaintext-tweakey pairs (P, K) and $(\overline{P}, \overline{K})$.*

Proof. We will show how the required plaintext and tweakey pairs are generated. We choose the nibble at Position 11 to introduce the initial difference because after completing six rounds, the difference is shuffled to Cell 8 of the round key, which coincides with the beginning of the impossible- differential trail, shown in Fig. 3. It can be seen that the AddRoundTweakey in the first round can be pushed behind the MixColumns operation by changing the first round key to $\mathsf{Lin}(K_1)$ where $\mathsf{Lin} = \mathrm{MC} \circ \mathrm{SR}$ represents the linear layer (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 $K = TK_1^1 \oplus TK_2^1$ and $\overline{K} = \overline{TK_1^1} \oplus \overline{TK_2^1}$ can be selected in a specific form, so that in Round 6, the tweakey difference is zero. Let us denote $\delta_1 = tk_1^1[11] \oplus \overline{tk_1^1}[11]$ and $\delta_2 = tk_2^1[11] \oplus \overline{tk_2^1}[11]$. In Round 6, the difference will appear in Cell 0 of the round key and so we want:

$$\begin{aligned} k^6[0] \oplus \overline{k^6}[0] &= tk_1^6[0] \oplus \overline{tk_1^6}[0] + tk_2^6[0] \oplus \overline{tk_2^6}[0] \\ &= tk_1^1[11] \oplus \overline{tk_1^1}[11] \oplus L^3\left(tk_2^1[11]\right) \oplus L^3\left(\overline{tk_2^1}[11]\right) \\ &= \delta_1 \oplus L^3\left(\delta_2\right) = 0 \end{aligned}$$

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

1. Take any Plaintext P and compute the state after the first round MixColumns, *i.e.* E^1.

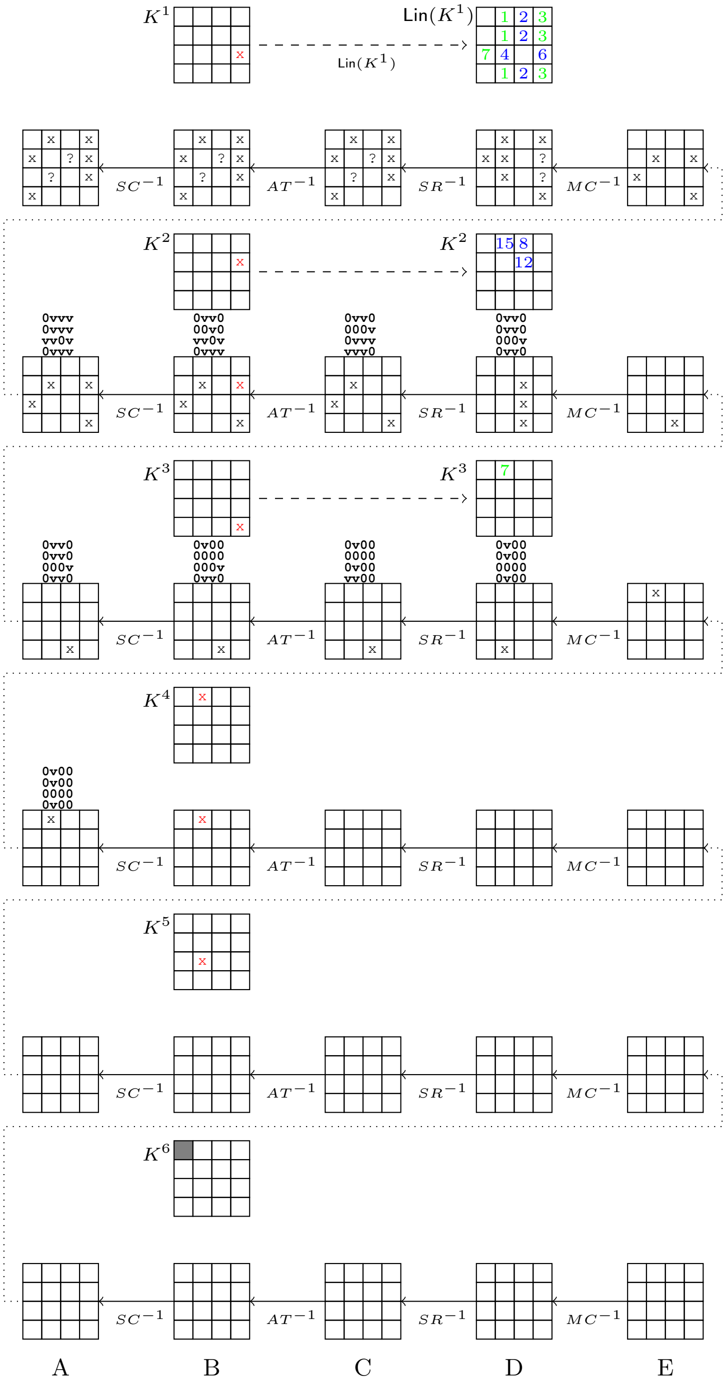

Fig. 4: Trail for the six forward rounds (the values of active nibbles in red are functions of δ_1, δ_2, the dark gray cell visualises the tweakey cancelation).

2. Take any three-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 calculates then $k^1[1]$, $k^1[3]$, $k^1[7]$ so that

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

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

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

   Lemma 1 ensures that the equation above has one solution on average.

4. β needs to be equal to $k^2[7] \oplus \overline{k^2}[7] = tk_1^2[7] \oplus tk_2^2[7] \oplus \overline{tk_1^2}[7] \oplus \overline{tk_2^2}[7]$. This is equal to $tk_1^1[11] \oplus L(tk_2^1[11]) \oplus \overline{tk_1^1}[11] \oplus L(\overline{tk_2^1}[11]) = \delta_1 \oplus L(\delta_2)$. So, the attacker chooses δ_1 and δ_2 satisfying $\delta_1 \oplus L^3(\delta_2) = 0$ and calculates $\beta = \delta_1 \oplus L(\delta_2)$. $\varDelta_2$ can then be determined as a solution of the equation:

$$S\left(E^1[7] \oplus k^1[3]\right) \oplus S\left(E^1[7] \oplus \varDelta_2 \oplus k^1[3]\right) = \beta \tag{1}$$

   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 $B^4[1]$ is equal to $\delta_1 \oplus L^2(\delta_2)$ to make all the state cells inactive in C^4, D^4, and E^4.

6. The attacker needs to guess three roundkey values in Round 1 (*i.e.* $k^1[2]$, $k^1[4]$, $k^1[6]$) and three roundkey values in Round 2 (*i.e.* $k^2[1] = tk_1^1[15] \oplus L(tk_2^1[15])$, $k^2[2] = tk_1^1[8] \oplus L(tk_2^1[8])$, $k^2[6] = tk_1^1[12] \oplus L(tk_2^1[12])$).
   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 tweakey nibbles mentioned above enables the attacker to calculate the value of $B^3[1]$. Then, she calculates $k^3[1] = tk_1^1[7] \oplus L(tk_2^1[7])$ as follows. Since $D^3[1] = B^3[1] \oplus k^3[1]$ holds, we have:

$$S\left(D^3[1] \oplus D^3[9] \oplus D^3[13]\right) \oplus S\left(D^3[1] \oplus D^3[9] \oplus \overline{D^3}[13]\right) = \delta_1 \oplus L^2(\delta_2).$$

Since the knowledge of the guessed key nibbles already allows the attacker to calculate $D^3[9]$, $D^3[13]$, and $\overline{D^3}[13]$, $k^3[1] = tk_1^1[7] \oplus L(tk_2^1[7])$ is the solution to the equation above. 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 also determines the values of $tk_1^1[7]$ and $tk_2^1[7]$.

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 $\delta_1 \oplus L^3(\delta_2) = 0$, the tweak difference becomes 0 in Round 6.

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

Lemma 4. *Given $C, \overline{C}$ as the two output ciphertexts after querying plaintext-tweakey pairs (P, K) and $(\overline{P}, \overline{K})$ 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.*

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 seven inactive cells in Cells 3, 4, 5, 8, 9, 11, and 14) after peeling off the final 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, she can do this filtering instantly leaving a fraction of 2^{-4} pairs after this stage.
3. Since the two bottommost rows of the state are not affected by the tweakey addition, and since $tk_1^1[7], tk_2^1[7]$ are already known, the attacker can calculate the actual values in Cells 0, 8, and 12 in A^{21} for the ciphertext pairs. These have to be equal since they are the output of the 20^{th}-round MixColumns operation on the leftmost column which had only one active nibble in its input. If the active Cells 8 and 12 are different, the attacker can reject the pair. This adds another filter with probability 2^{-4}.
4. Since the actual values in Cell 0 in A^{21} for the ciphertext pairs were already calculated in the previous step, the attacker checks if the value of the active Cell 0 is equal to that of Cells 8 and 12, and rejects the pair otherwise. 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 $A^{21}[5] = S^{-1}\left(k^{21}[5] \oplus C^{21}[5]\right)$, $k^{21}[5]$ is a solution to the equation below:

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

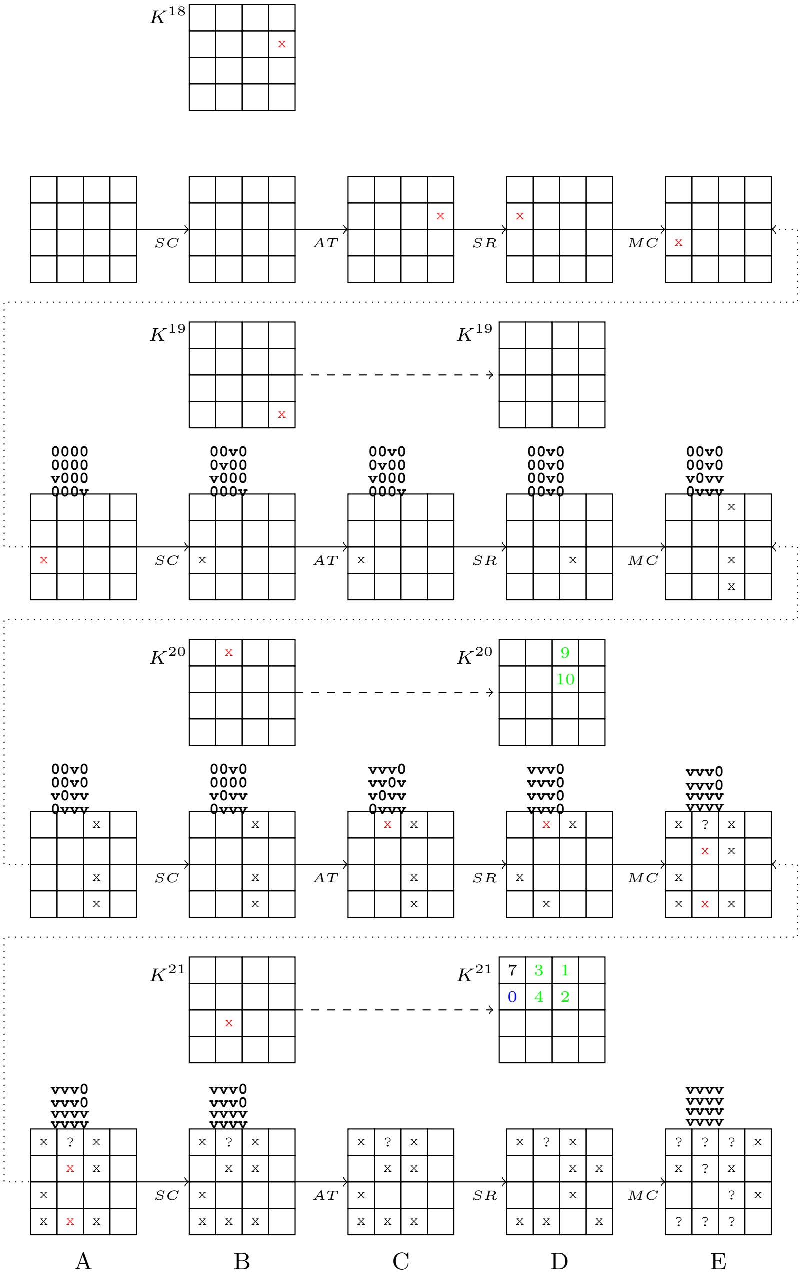

Fig. 5: Trail for the four backward rounds (the values of active nibbles in red are functions of δ_1 and δ_2).

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 and 6 of A^{21} are equal to the active nibble in Cell 14. Again, this works since those cells are output of the 20^{th}-round MixColumns operation on Column 2 which had 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 leftmost column of A^{21} and hence to compute the leftmost column of D^{20}.
8. The value of the active nibble in cell 10 of A^{20} is given as:

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

   Since the leftmost column of D^{20} is known, the attacker can calculate η, which must be equal to Cell 14 of A^{20} since they are output of the 19^{th}-round MixColumns operation with one active input nibble.

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

   It holds that $A^{21}[1] = S^{-1}\left(C^{21}[1] \oplus k^{21}[1]\right)$ and $\overline{A^{21}}[1] = S^{-1}(\overline{C^{21}}[1] \oplus k^{21}[1])$. By calculating Equations (2) and (3), the attacker 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 $tk_1^1[i] \oplus tk_2^1[i]$, for $i = 1, 2, 3, 4$, were already determined during the calculation of the forward trail. So, using their values, the attacker can determine the actual values $tk_1^1[i],\ tk_2^1[i]$ for $i = 1, 2, 3, 4$.
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 Cells 10 and 14 since they are output of the 19^{th}-round MixColumns operation with one active input nibble. This is done by solving

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

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

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

   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 $k^{20}[6] = tk_1^1[10] \oplus L^{10}(tk_2^1[10])$.

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

3.1 First Attack

Now, we put together the findings of Lemma 3 and 4 into an attack procedure (see Figure 8 in the appendix for details):

1. The attacker chooses the nibble values of the random base variable E^1 in all locations except Cells 5, 7, 8, and 15.
2. She chooses fixed differences δ_1, δ_2 satisfying $\delta_1 = L^3(\delta_2)$.
3. For each choice of ($E^1[5]$, $E^1[7]$, $E^1[8]$, $E^1[15]$) (2^{16} choices):
    - Calculate P by inverting the first round.
    - 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 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 Cells 5, 7, 8, and 15. For every P_i, there exist $(2^4 - 1)^4 \approx 2^{15.6}$ such values of P_j, and so $2^{16+15.6} = 2^{31.6}$ pairs to work with. The attack now proceeds as follows. 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 Δ_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 proceeds; otherwise, she aborts.
- Request the ciphertext $\overline{C}$ for $(\overline{P}, \overline{K})$ and the ciphertext C for (P, K).
- If $C \oplus \overline{C}$ does not pass the 2^{-36} filter (Steps 1, 2, and 3 in Lemma 4), then abort and start again.
- If they pass the filter, the attacker can guess seven 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 fulfills the condition required in Step 4 of Lemma 4.
- Therefore, the attacker has a set of $2^{28-4} = 2^{24}$ wrong key candidates.

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

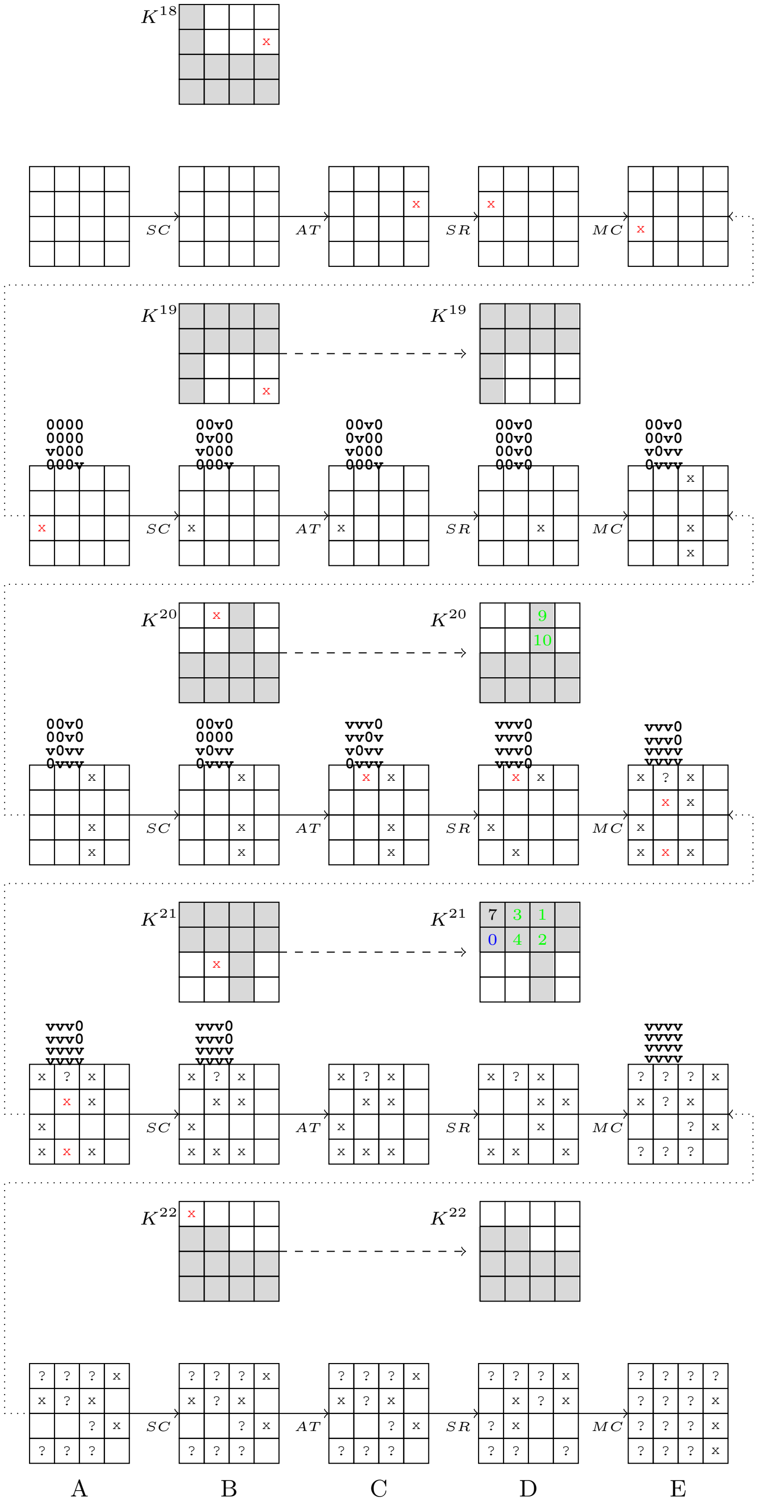

Fig. 6: Trail for the five backward rounds (the values of active nibbles in red are functions of δ_1, δ_2, grey cells are the key, white cells are the tweak).

Complexity. For every base value of E^1, the attacker 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 $k^1[1]$, $k^1[3]$, and $k^1[7]$, of which a fraction of 2^{-4} passes the filter in Equation (1). So, she has $2^{31.6}$ pairs to work with. In fact, for every pair (P_i, P_j) there is only one choice of α going forward on average.

$$\text{Time complexity} = \max\left\{2^{x+17} \text{ encryptions }, 2^{x-4.4+24} \text{ guesses}\right\} = 2^{x+19.6}.$$

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:

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

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 the total number of guesses is $2^{74.6}$. We also need 2^{68} memory accesses, which are negligible in the total complexity. The memory complexity is upper bounded by storing one bit per key candidate which is therefore 2^{68} bits. The memory for storing the approximately $2 \cdot 2^{17}$ plaintexts and corresponding ciphertexts of a structure at each time is negligible.

3.2 22-Round Attack under Partially Known Tweak

The attack above can be extended to 22-round SKINNY-64/128 under the assumption that 48 of the 128 bits in the tweakey are publicly known tweak (see Figure 9 in the appendix for details). In particular, we assume that $tk_1^1[i], tk_2^1[i]$ for $i = 8, 11, 12, 13, 14, 15$ are reserved for the tweak. The remaining 80 bit constitute the secret key.
In this case, the attacker can add a round at the end (see Fig. 6 for details). Knowing six out of eight cells in the lower half of the tweakey blocks helps in the following way. From the ciphertext (i.e., E^{22}), one can revert the final round to compute E^{21} if we guess $k^{22}[4,5]$, i.e., $tk_1^1[9,10] \oplus L^{11}(tk_2^1[9,10])$. The attack is almost the same as the previous attack, except that the tweakey indices $i = 8, 11, 12, 13, 14, 15$ and their functions are known and need not be guessed.

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 $k^1[1]$, $k^1[3]$, and $k^1[7]$ 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 proceeds; otherwise, she aborts.

- The attacker doesn't need to guess the Round 2 tweakey nibbles since these are in the lower half of the tweakey blocks and therefore known.
- Retrieve the ciphertext $\overline{C}$ for $(\overline{P}, \overline{K})$ and the ciphertext C for (P, K).
- Guess $k^{22}[4,5] = tk_1^1[9,10] \oplus L^{11}(tk_2^1[9,10])$ to get E_{21}.
- If $E_{21} \oplus \overline{E_{21}}$ does not pass the 2^{-36} filter, then abort and restart.
- 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 and 11 of Lemma 4, the attacker can uniquely determine $tk_1^1[9,10]$ since $tk_1^1[9,10] \oplus L^{11}(tk_2^1[9,10])$ is already guessed.
- If they pass the filter, the attacker can guess six 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 fulfills the condition in Step 4 of Lemma 4.
- Therefore, the attacker has a set of $2^{24-4} = 2^{20}$ wrong key candidates.

The procedure above is repeated with 2^x chosen plaintexts until a single key solution remains for the 16 nibbles of the tweakey.

Complexity. For every base value of E^1, the attacker makes 2^{17} encryption calls. Out of these, she has $2^{31.6}$ pairs to work with. For each pair, the attacker can choose then 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 of 2^{-4} passes the filter in Equation (1). So, the attacker has $2^{31.6}$ pairs to work with. In effect, for every pair (P_i, P_j) there is only once choice of α going forward on average.

$$\text{Time complexity} = \max\left\{2^{x+17} \text{ encryptions}, 2^{x-4.4+20} \text{ guesses}\right\} = 2^{x+17}.$$

The attacker gets wrong solutions for $2^{x-4.4+20} = 2^{x+15.6}$ incorrect solutions for 16 nibbles. To reduce the keyspace to 1 we need:

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

For this, we need $x = 54$. So, the total number of encryption calls to 22-round SKINNY-64/128 is $2^{54+17} = 2^{71}$. We also need 2^{64} memory accesses, which are negligible in the total complexity. The memory complexity is upper bounded by storing one bit per key candidate which is therefore 2^{64} bits. The memory for storing the approximately $2 \cdot 2^{17}$ plaintexts and corresponding ciphertexts of a structure at each time is negligible.

3.3 23-Round Attack under Partially Known Tweak

We can extend the 22 round attack to a 23 round attack by prepending one round at the beginning. In order to not disturb the notation, we denote the additonal round prepended at the beginning as the 0-th round. That is, the 23 rounds are labelled as rounds 0 to 22, and the variables A^0, B^0 etc. are defined as above. The plaintext is denoted by A^0 and the ciphertext by E^{22}. Note that, from the base value of E^1, the plaintext can be calculated if we guess $k^0[9, 10]$. There are two principal differences to the 22-round attack.

1. When the attacker guesses $k^{22}[4, 5]$ which is $tk_1^1[9, 10] \oplus L^{11}(tk_2^1[9, 10])$ to invert the final round to get E_{21}, he uniquely determines $tk_1^1[9, 10]$ and $tk_2^1[9, 10]$. This is because at the beginning of the outer loop $k^0[9, 10]$ has already been guessed by the attacker to invert the initial round.
2. As the attacker can no longer determine $k^{20}[2] = tk_1^1[9] \oplus L^{10}(tk_2^1[9])$ and $k^{20}[6] = tk_1^1[10] \oplus L^{10}(tk_2^1[10])$ using Steps 10 and 11 of Lemma 4. The probability that with the given values of $tk_1^1[9, 10]$ and $tk_2^1[9, 10]$, Equations (4) and (5) are satisfied is 2^{-8}. This decreases the probability of ciphertext filter from 2^{-36} to 2^{-44}.

For each initial guess of $k^0[9, 10]$, the guessed and calculated key bytes are:

#	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

The 14 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 $i = 0.$

(c) $tk_1^1[i] \oplus tk_2^1[i]$ for $i = 6.$

(d) $tk_1^1[i] \oplus L^{11}(tk_2^1[i])$ for $i = 9, 10$

As before, a fraction of 2^{-4} tweakeys fulfills the condition in Step 4 of Lemma 4. Therefore, the attacker has a set of $2^{24-4} = 2^{20}$ wrong key candidates.

Complexity. For each iteration of the outer loop, the complexity is calculated as follows: For every base value of E^1, the attacker makes 2^{17} encryption calls. Out of those, she has $2^{31.6}$ pairs to work with. For each pair, the attacker can choose then 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 of 2^{-4} passes the filter in Equation (1). In effect, for every pair (P_i, P_j) there is only one choice of α going forward on average.

$$\text{Time complexity} = \max\left\{2^{x+17} \text{ encryptions}, 2^{x+31.6-44+20} \text{ guesses}\right\} = 2^{x+17}.$$

The attacker gets $2^{x+31.6-44+20} = 2^{x+7.6}$ incorrect solutions for 14 nibbles. To reduce the keyspace to 1 we need:

$$2^{14\times 4} \cdot \left(1 - 2^{-14\times 4}\right)^{2^{x+7.6}} \approx 2^{14\times 4} e^{-2^{x-48.4}} = 1.$$

We need $x = 54$ leaving the total number of encryption calls to 22-round SKINNY-64/128 with $2^{54+17} = 2^{71}$. Multiplying this by 2^8 for the outer loop gives a total complexity of $2^{71+8} = 2^{79}$ which is just short of exhaustive search for the 80-bit key. We also need $2^{56+8} = 2^{64}$ memory accesses, which are negligible in the total complexity. The memory complexity is upper bounded by storing one bit per key candidate which is therefore 2^{64} bits. The memory for storing the approximately $2 \cdot 2^{17}$ plaintexts and ciphertexts of a structure is negligible.

4 Conclusion

In this paper, we outline related-key impossible-differential attacks against 21-round SKINNY-64/128 as well as attacks on 22 and 23 rounds under the assumption of having 48 of the 128-bit tweakey as public tweak. Our attacks are based on an 11-round impossible differential trail, to which we prepend six and append five rounds before and after the trail, respectively, to obtain an attack on 22 rounds. Finally, we can prepend a 23-rd round under similar assumptions.

Acknowledgements

This work was initiated during the group sessions of the 6th Asian Workshop on Symmetric Cryptography (ASK 2016) held in Nagoya, Japan. Ralph Ankele is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET. Gaoli Wang is supported by National Natural Science Foundation of China (Grant Nos. 61572125, 61373142), Shanghai High-Tech Field Project (Grant No. 16511101400). Siang Meng Sim is supported by the Singapore National Research Foundation Fellowship 2012 (NRF-NRFF2012-06). This work has been supported in part by the Austrian Science Fund (project P26494-N15).

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. In: Joye, M., Nitaj, A. (eds.) AFRICACRYPT 17: 9th International Conference on Cryptology in Africa. Lecture Notes in Computer Science, vol. 10239. Springer, Dakar, Senegal (2017)

Appendix

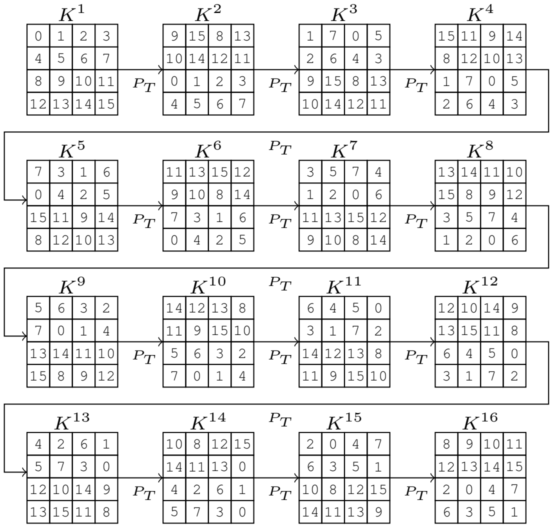

Fig. 7: The permutation P_T in the tweakey schedule has a period of 16.

Table 2: Difference-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

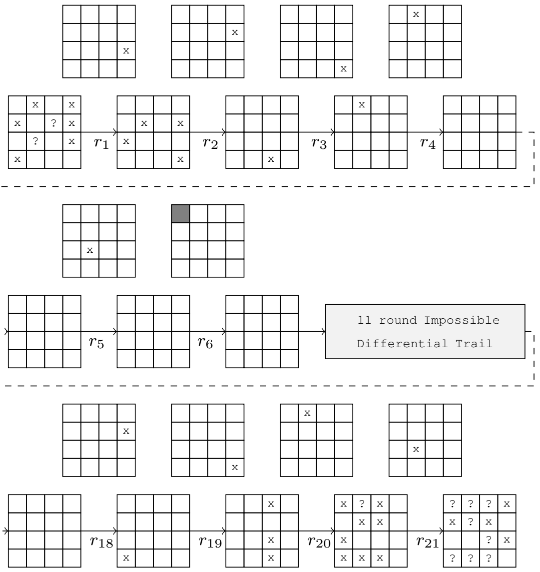

Fig. 8: Related-key impossible differential attack on 21-round SKINNY 64/128 (the dark gray cell visualises the cancelation of the tweakeys).

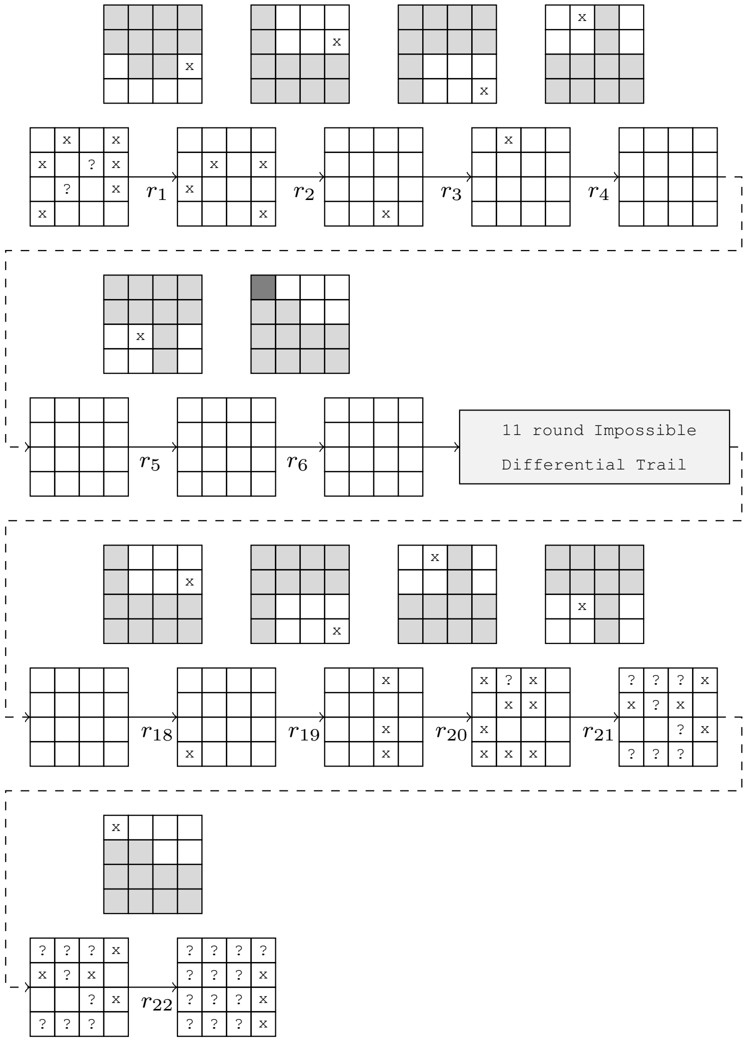

Fig. 9: Related-Key Impossible Differential Attack on 22 round SKINNY 64/128 (grey cells are the key, white cells are the tweak, the dark gray cell visualises the cancelation of the tweakeys)