Designs, Codes and Cryptography (2024) 92:1175–1199
https://doi.org/10.1007/s10623-023-01336-z

Observations on the branch number and differential analysis of SPEEDY

Lei Zhang[1,2]

Received: 8 September 2022 / Revised: 3 November 2023 / Accepted: 5 November 2023 /
Published online: 8 December 2023

Abstract
In this paper, we present some new observations on the branch number and study concrete differential analysis of SPEEDY. It is a new low-latency block cipher proposed at TCHES 2021. It employs SPS-type round function and consists of only 5/6/7 rounds. Since the iteration rounds are rather small so as to achieve ultra low-latency in encryption speed, it will be crucially important to analyze its security margin accurately. In this paper, we first propose a new notation of *partition branch number* which can describe the minimum number of active S-boxes for 2-round SPEEDY more accurately. An efficient algorithm to compute the value of *partition branch number* is also given. Then by extending the notation to *higher-order partition branch number*, we can obtain more accurate results of the minimum number of active S-boxes for 3–7 rounds. As a result, the maximum expected differential probabilities are significantly higher than the results estimated by designers. Based on this, we search for optimal differential characteristics of SPEEDY while considering the difference distribution table of S-box. We present examples of differential characteristics for 2–7 rounds. Furthermore, by utilizing the simple bit-permutation key schedule of SPEEDY, we can extend the differential trail search method and construct an efficient 6-round related-key differential trail with probability $2^{-179.2}$. Based on it, we can present related-key differential attack on full round SPEEDY-7-192 with data complexity of $2^{186.2}$ chosen-plaintexts and time complexity of $2^{160.13}$ encryptions.

Keywords Block cipher · Branch number · Low-latency · Differential trail

Mathematics Subject Classification 11T71 · 14G50 · 94A60

Communicated by X. Wang.

✉ Lei Zhang
zhanglei@iscas.ac.cn

[1] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

1 Introduction

In recent years, the design of lightweight block ciphers is one of the most important research areas and has attracted lots of attention from all over the world. Lightweight encryption is motivated by the need to provide acceptable security for specific applications used in resource constrained environment, such as RFID, sensor network, and microcontroller in the Internet of Things (IoT). The most important concern is an optimal implementation with much lower area, or power consumption than traditional encryption standard such as AES [16]. A variety of lightweight schemes aiming at various goals have been proposed in the last few years. The first generation of lightweight block ciphers such as PRESENT [8] and KATAN [13], mainly focused on hardware implementation performance such as area cost. Then software-oriented designs such as nibble-wise block cipher LBlock [23] and TWINE [22] have emerged, which take not only hardware but also 8/16/32-bit multiple software platforms into account. Nowadays, designs aiming at other goals have also been studied, such as the lightweight tweakable block cipher SKINNY [7], Midori [3] which is focusing on low energy, CRAFT [6] with efficient protection against DFA attacks. Moreover, designs capable of serialized implementation such as SIMON [5], Piccolo [20], and WARP [2] can achieve ultra lightweight with a very small hardware footprint compared to round-based implementation but at the cost of long cycles.

On the other hand, some new designs aiming at low-latency have been proposed rather recently. It is motivated by the need to provide real-time security for some specific applications, such as instant authentication, storage or memory encryption, high-speed encryption for secure processor architectures, and embedded systems with real-time requirements. For all these cases, low-latency encryption and instant response time are highly desirable. Hence an encryption scheme should be optimized for latency and the entire process should be completed within the shortest possible delay. Therefore, the most important feature of low-latency block cipher is to encrypt a block of data within one single clock cycle. However, for this kind of fully-unrolled implementation it will be a huge challenge to achieve acceptable chip area. Therefore, the number of rounds must be moderate and the round function may be relatively complex. For example, SP-type or SPS-type round function with lightweight low latency components such as 4-bit S-box and heavy linear layer such as MDS matrix are usually employed.

So far, only a few low-latency ciphers have been proposed. PRINCE [9] is the first low-latency block cipher proposed by Borghoff *et al.* at ASIACRYPT 2012, and it has already been deployed in a number of products including LPC55S of NXP Semiconductors [17]. Later, Bozilov *et al.* proposed an updated version called PRINCEv2 [12] which claims to increase the security level of PRINCE by making only small modifications. Inspired by the design of PRINCE, Beierle *et al.* proposed a low-latency tweakable block cipher called MANTIS [7]. Qualcomm company has also proposed a low-latency block cipher family called QARMA [1], which targets at applications such as memory encryption and pointer authentication. QARMA has already been used to achieve *control flow integrity* (CFI) in the products of ARMv8.3 [18]. In 2021, Banik *et al.* proposed Orthros [4], a low-latency pseudorandom function (PRF) which ignors the support of decryption to achieve ultra low-latency. Leander *et al.* proposed another ultra low-latency block cipher family called SPEEDY at CHES 2021 [15]. It primarily targets at secure process architectures embedded in high-end CPUs. It achieves ultra low-latency in single-cycle encryption speed, while the decryption is less efficient. Up to now, SPEEDY outperforms all the other known encryption primitives with the lowest latency in hardware.

On the other hand, in the security aspect, there exists more risk for low-latency cipher since the number of rounds must be as small as possible in order to achieve ultra low-latency. Not surprisingly, many low-latency block ciphers suffer serious security problems. Several theoretical attacks which can achieve even full-round have been reported. For example, Soleimany *et al.* [21] proposed an effective reflection attack which can apply to full-round $\texttt{PRINCE}_{core}$, and Dobraunig *et al.* [14] proposed a practical clustering differential attack on full-round $\texttt{MANTIS}_5$. Therefore, for low-latency block cipher designs, study about the security margin and accurate security evaluation against known attacks may be very important and desirable.

Related Work: The notation of branch number is usually used to evaluate the minimum number of active S-boxes for SP-type round function. However, for block ciphers employing SPS-type round function, it will be difficult to obtain the accurate number of active S-boxes based on branch number directly. Therefore, the combination of SPS is usually considered together as a Super S-box and then the minimum number of active Super S-boxes is evaluated instead. Obviously, it will be less accurate and this method is usually used as a rough security evaluation against differential analysis in the design of block ciphers with SPS-type round function.

As a newly proposed low-latency block cipher, SPEEDY has not only outstanding performance in latency, but also has an unusual structure in contrast to other known low-latency block ciphers. Instead of the α-reflection structure [9] with SP-type round function used in PRINCE, MANTIS and QARMA, SPEEDY employs an SPS-type round function with two layers of 6-bit S-box connected by bit rotations and then a binary matrix multiplication as the linear function. The block size and key size of SPEEDY are both 192-bit, and the number of rounds can be 5/6/7 which corresponds to different levels of security. In order to evaluate its security against differential and linear attacks, the designers try to analyze the minimum number of active S-boxes to give an upper bound for the probabilities of differential and linear trails. A direct way to compute the minimum number of S-boxes is by making use of the notation of branch number.

However, since the round function of SPEEDY consists of two layers of 6-bit S-box connected by bit rotations which can be considered as a Super S-box, it will be difficult to obtain the number of active 6-bit S-boxes directly. Therefore, in [15] the designers only considered 1-bit to 1-bit transitions through Super S-box whose maximum differential probability is 2^{-6}, and then the binary matrix with branch number $bn = 8$ can ensure that the maximum expected differential probability of differential trails over tow rounds of SPEEDY is $2^{-6\times 8}$. Moreover, they proposed an extension for the definition of branch number called *higher-order branch number* bn_r, which represents the minimum number of 1-bit to 1-bit transitions over r rounds. Then the maximum expected differential probability over r-round SPEEDY is estimated by $2^{-6\times bn_r}$.

Obviously, this is inaccurate since only 1-bit to 1-bit transitions through Super S-box are considered. Without the restriction of 1-bit difference, there should exist differential trails with less active 6-bit S-boxes. Not surprisingly, when they search for the minimum number of active S-boxes by assuming that all the differentials through the S-box are possible and there are at most 8 active S-boxes in the internal state, the results of minimum number of active 6-bit S-boxes for 2–4 rounds are all significantly smaller than the values estimated according to branch number [15]. This may threaten the security margin since the round number for low-latency block ciphers are usually rather small. Therefore, for block ciphers employing SPS-type round function, accurate evaluation of the minimum number of active S-boxes may be vital to the designers, especially when heuristic search for differential trail of long rounds is impractical.

Moreover, we have noticed that after the submission of our paper and during the review phase, Boura *et al*. reported some similar results about differential attack of SPEEDY in [10, 11]. They implemented a search for finding optimal trails under some constraints. The main search strategy was to precompute all good one-round trails such that both columns x and $M(x)$ have at most 7 active bits each. Then they chained them to create longer trails and finally obtained a 5.5-round differential distinguisher with probability $2^{-183.59}$. Their main contributions focused on improvements of the key-recovery part. They proposed some non-trivial techniques of efficient data and key-sieving, and based on them they presented a full-round differential attack of SPEEDY-7-192 with a time complexity of $2^{187.84}$ and data complexity of $2^{187.28}$.

Our Contribution: In this paper, we present some new observations on the branch number and study concrete differential analysis of SPEEDY. First of all, we propose a new notation of *partition branch number*, which can describe the minimum number of active S-boxes for SPEEDY more accurately. We also give an efficient algorithm to compute the value of *partition branch number*. Then by extending the notation to *higher-order partition branch number*, we can obtain more accurate results of the minimum number of active S-boxes for more rounds. Based on this, we have obtained the minimum number of active 6-bit S-boxes for 2–7 rounds SPEEDY are as follows:

$$pbn_2 = 13, \quad pbn_3 = 23, \quad pbn_4 = 35, \quad pbn_5 = 45, \quad pbn_6 = 57, \quad pbn_7 = 67.$$

It is noteworthy that these results are significantly smaller than the minimum number of active S-boxes in [15]. Hence, it will contradict the security margin evaluated by the designers. Moreover, our computation of *higher-order partition branch number* can be used to evaluate the minimum number of active S-boxes more directly and efficiently. Compared to the time-consuming search method, we can obtain accurate results for arbitrary rounds while automatic search may be infeasible for long rounds.

On this basis, we also search for the optimal differential characteristics of SPEEDY. Instead of the assumption that all the differential transitions through the S-box are possible, we take the Difference Distribution Table (DDT) of the 6-bit S-box into consideration. Therefore, we can study concrete differential characteristics which satisfy the actual difference distribution table and evaluate the security margin of SPEEDY more precisely. We first present a special kind of 1-bit to 1-bit differential trails for 2–7 rounds SPEEDY together with the maximum differential probabilities. Then, based on the computation of *higher-order partition branch number*, we present examples of optimal differential characteristics for 2–7 rounds SPEEDY, whose differential probabilities are $2^{-46.2}$, $2^{-76.72}$, $2^{-129.2}$, $2^{-170.0}$, $2^{-216.0}$ and $2^{-266.2.0}$ respectively.

Furthermore, by utilizing the simple bit-permutation key schedule of SPEEDY, we can extend the differential trail search method and construct an efficient 6-round related-key differential trail with probability $2^{-179.2}$. Based on it, we can present a related-key differential attack on full round SPEEDY-7-192 with data complexity of $2^{186.2}$ chosen-plaintexts and time complexity of $2^{160.13}$ encryptions. Compared to other third-party cryptanalysis results on SPEEDY [10, 19], this is the first full-round related-key differential attack reported so far and the attack complexity outperforms all the previous known results.

Organization of the Paper: First of all, we give a brief description of SPEEDY in Sect. 2. Then, some observations of the branch number are explained in Sect. 3. In Sect. 4, we present the definition of *partition branch number* and based on it we give some more accurate results about the security evaluation of SPEEDY against differential analysis. Concrete differential trails for 2–7 rounds SPEEDY and experimental validation are provided in Sect. 5. In Sect. 6,

Table 1 Contents of the 6-bit S-box used in SPEEDY (in hexadecimal notation)

$x_0x_1x_2$	$x_3x_4x_5$							
	0	1	2	3	4	5	6	7
0	08	00	09	03	38	10	29	13
1	0c	0d	04	07	30	01	20	23
2	1a	12	18	32	3e	16	2c	36
3	1c	1d	14	37	34	05	24	27
4	02	06	0b	0f	33	17	21	15
5	0a	1b	0e	1f	31	11	25	35
6	22	26	2a	2e	3a	1e	28	3c
7	2b	3b	2f	3f	39	19	2d	3d

we present related-key differential attack on full round SPEEDY-7-192. Finally, Sect. 7 concludes the paper.

2 Specification of SPEEDY

SPEEDY is a family of ultra low-latency block ciphers with different block size and varying number of rounds. For example, SPEEDY-r-$6l$ is an instance with block and key sizes $6l$ bits and it iterates r rounds. The suggested parameter is SPEEDY-r-192 which achieves 128-bit security when iterated $r = 6$ rounds and full 192-bit security when iterated $r = 7$ rounds, while a decent security level ($\geq 2^{128}$ time complexity when data complexity is limited to $\leq 2^{64}$) when iterated $r = 5$ rounds.

For SPEEDY-r-192, the internal state is viewed as an 32×6 array of bits. The notation $x[i, j]$ denotes the bit located at row i and column j of the state x with $0 \leq i < 32$ and $0 \leq j < 6$. First of all, the internal state is initialized with the 192-bit plaintext with the same order used for indexing bits. Then, round functions are applied on the internal state. Each round function is composed of the following operations: AddRoundKey (A_{k_r}), SubBox(SB), ShiftColumns(SC), SubBox(SB), ShiftColumns(SC), MixColumns (MC), AddRoundConstant (A_{c_r}). Note that in the last round, the linear layer and constant addition are omitted, and instead an extra key addition is applied. Therefore, the encryption procedure can be expressed as follows:

$$\mathrm{A}_{k_0} \rightarrow \mathrm{SB} \rightarrow \mathrm{SC} \rightarrow \mathrm{SB} \rightarrow \mathrm{SC} \rightarrow \mathrm{MC} \rightarrow \mathrm{A}_{c_0} \rightarrow \cdots \rightarrow \mathrm{A}_{k_{r-1}} \rightarrow \mathrm{SB} \rightarrow \mathrm{SC} \rightarrow \mathrm{SB} \rightarrow \mathrm{A}_{k_r}$$

- SubBox(SB): The 6-bit S-box S is applied to each row of the state. The contents of the 6-bit S-box is given in Table 1.
- ShiftColumns(SC): The j-th column of the state is rotated upside by j bits.
- MixColumns(MC): A cyclic binary matrix is multiplied to each column of the state. Namely, it can be computed as follows and the addition in the indices of the state is in module 32 for the first (row) index.

$$y[i, j] = x_{i,j} \oplus x_{[i+1,j]} \oplus x_{[i+5,j]} \oplus x_{[i+9,j]} \oplus x_{[i+15,j]} \oplus x_{[i+21,j]} \oplus x_{[i+26,j]}, \forall i, j.$$

- AddRoundKey(A_{k_r}): The 192-bit round key k_r is XORed to the state.
- AddRoundConstant(A_{c_r}): The 192-bit constant c_r is XORed to the state.

The KeySchedule of SPEEDY receives a 192-bit master key as the first round key and then applies a simple bit permutation to compute the next round key iteratively. We will omit the description of RoundConstant values since they are not related to our work. Interested readers can refer to [15] for more details.

3 Observations on branch number

3.1 (*Higher-order*) branch number

The SP-type round function is one of the most widely used structures in the design of block cipher. Normally, a layer of non-linear S-box is used to provide confusion effect and a linear function is used to provide diffusion effect. For measuring diffusion effect, the notation of branch number is defined to represent the minimum number of active S-boxes in consecutive 2-round. However, for block ciphers employing SPS-type round function, such as SPEEDY and Piccolo, it will be difficult to obtain the number of active S-boxes directly based on branch number. Therefore, they have to consider the two layers of S-boxes together as Super S-box and then evaluate the minimum number of active Super S-box instead [15, 20].

For the round function of SPEEDY, the linear function MC can be considered as a cyclic binary matrix multiplied to each column of the state. Denote the corresponding binary cyclic matrix of MC as M, and then its branch number is defined as:

$$bn = \min_{x \in F_2^l \setminus \{0\}} hw(x) + hw(M \times x^T)$$

where $hw(\cdot)$ denotes the hamming weight of a binary array. According to the differential analysis in [15], they only considered 1-bit to 1-bit differential over SB $\circ$ SC $\circ$ SB where the first and second SB operations are both 1-bit to 1-bit transitions, and its maximum differential probability is 2^{-6}. Then, considering that the branch number of MC is $bn = 8$, they can ensure that the maximum expected differential probability of this kind of 1-bit to 1-bit differential trails over 2-round SPEEDY is $(2^{-6})^8 = 2^{-48}$.

Moreover, in order to evaluate the maximum differential probability of 1-bit to 1-bit differential trails over several rounds of SPEEDY, they proposed a new notation of *higher-order branch number* bn_r. For example, the *higher-order branch number* for 3-round SPEEDY can be computed as follows:

$$bn_3 = \min_{\substack{i_1, i_2, i_3 \neq 0 \\ H[i_1][i_2]=H[i_2][i_3]=1}} i_1 + i_2 + i_3$$

where H is a binary table such that the element in the position $H[i][j]$ is 1 if and only if there is an $x \in F_2^l \setminus \{0\}$ with $hw(x) = i$ and $hw(M \times x^T) = j$. Obviously, the branch number bn is the same as bn_2 defined as follows:

$$bn_2 = \min_{\substack{i_1, i_2 \neq 0 \\ H[i_1][i_2]=1}} i_1 + i_2$$

Then the maximum expected differential probability of 1-bit to 1-bit differential trails over r-round SPEEDY can be estimated as $2^{-6 \times bn_r}$.

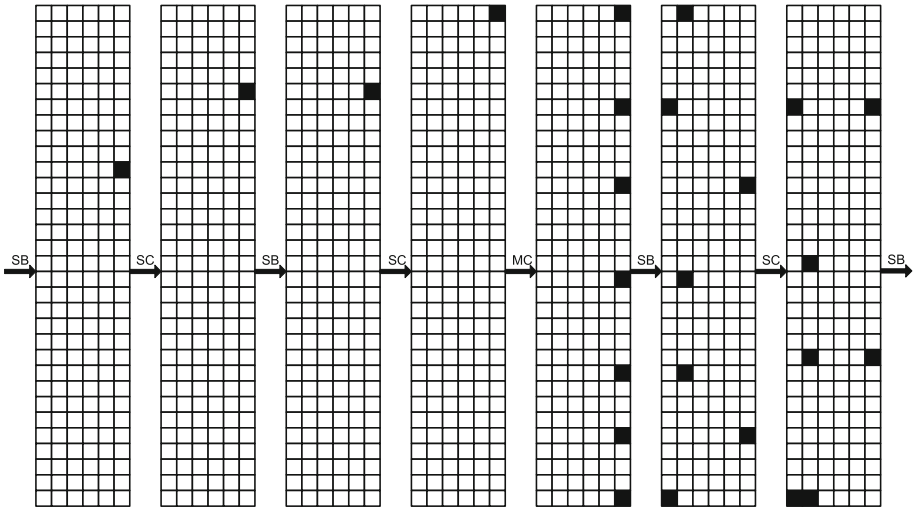

Fig. 1 An example of 2-round differential trail for SPEEDY

3.2 Differential analysis of 2-round SPEEDY

In the above analysis, they have only considered 1-bit to 1-bit differential over $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$ in order to use the branch number of MC to evaluate the maximum expected differential probability over two rounds of SPEEDY. However, for all the differential trails over $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$, if we can consider multiple nonzero bits differential transition for the second SB operation, the number of active 6-bit S-boxes can be reduced significantly.

Without loss of generality, we set one column of the input and output states of MC as active, and then the operations for two rounds of SPEEDY can be simplified as follows:

$$\overleftarrow{\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}} \circ \mathtt{SC} \circ \mathtt{MC} \circ \overrightarrow{\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}}.$$

Take the central operation MC as a starting point, and assume the second SB operation in both forward and backward directions can take n-bit differential transitions with $1 \leq n \leq 6$. Then we can obtain differential trail of 2-round SPEEDY with much higher probability. We give a simple example to illustrate the differential transitions of this kind of trails in the following.

Example 1 We present an example of 2-round differential trail for SPEEDY. Fig. 1 illustrates its differential propagations in detail. The black box denotes '1' difference and the empty box denotes '0' difference, respectively. There are only 13 active S-boxes and the maximum differential probability should be $(2^{-3})^{13} = 2^{-39}$. Obviously, it is much higher than the maximum differential probability of 1-bit to 1-bit differential trails which is estimated to be 2^{-48}.

4 Partition branch number

Inspired by the analysis of 2-round differential trail of SPEEDY, we can define a new form of branch number to describe the least number of active S-boxes more accurately. For the

round function of SPEEDY, since SC does not change the column position and MC is a cyclic matrix, we can omit the second SC in the computation of branch number as follows.

$$\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}.$$

Similar to the definition of hamming weight of a binary array $hw(\cdot)$, we can define a new form of *partition hamming weight* as follows.

Definition 1 (*Partition Hamming Weight*) For a binary array x, partition all the active bits into several sets in order and each set satisfies that the maximum distance does not exceed k. Then the minimum number of active sets is defined as the *partition hamming weight* of a binary array, which is denoted as $phw_k(x)$.

Considering the round function of SPEEDY, the binary array x defined in the above definition can be considered as a column of the state. Since the operation of SC is a rotation with 5 bits at most, we will only consider the situation of $k = 5$. Therefore, in the remaining of this paper, we will abbreviate $phw_5(x)$ as $phw(x)$. The definition of *partition hamming weight* represents the minimum number of active S-boxes for the output difference after the operations of SB and SC. In Fig. 1, the difference propagation of the second round illustrates a sample of partition. Moreover, we will give a more detailed example as follows.

For example, given the following binary array x, which can be considered as a column of the state. There are 7 active bits in the positions {0,1,5,9,15,21,26}.

$$x^T = [1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$$

They can be partitioned into 4 active sets as follows. The first set contains the 0-th and 1-th bits and its maximum distance is 1. The second set contains the 5-th to 9-th bits and its maximum distance is 4. The third set contains only the 15-th bit and its maximum distance is denoted as 0. The last set contains the 21-th to 26-th bits and its maximum distance is 5. Obviously, this is a valid partition. By checking all the possible valid partitions, we can get the minimum number of active sets is 4, namely $phw(x) = 4$.

$$\{1, 1\}, 0, 0, 0, \{1, 0, 0, 0, 1\}, 0, 0, 0, 0, 0, \{1\}, 0, 0, 0, 0, 0, \{1, 0, 0, 0, 0, 1\}, 0, 0, 0, 0, 0$$

On this basis, we can define a new kind of branch number to evaluate the minimum number of active S-boxes considering multi-bit differentials instead of only 1-bit to 1-bit differentials. Similar to the above analysis, we set one column of input and output states of MC as active, and assume the second SB operation in both forward and backward directions can take n-bit differential transitions with $1 \leq n \leq 6$. According to the definition of *partition hamming weight*, the maximum distance of each active set does not exceed 5. Therefore, after the operations of SB and SC, they can be transited into the same S-box. Then an improved branch number can be defined as follows.

Definition 2 (*Partition Branch Number*) For the binary cyclic matrix M of the MC operation, we define its *partition branch number* as follows:

$$pbn = \min_{x \in F_2^l \setminus \{0\}} hw(x) + hw(M \times x^T) + phw(x) + phw(M \times x^T).$$

The *partition branch number* can ensure that there are at least pbn active S-boxes over two rounds. Here we give a brief explanation for this claim. For any input difference $x \in F_2^l \setminus \{0\}$, the output difference after MC should be $M \times x^T$. We can use $hw(x) + hw(M \times x^T)$ to denote the number of active S-boxes for the operations of $\mathtt{SB} \circ \mathtt{MC} \circ \mathtt{SB}$. Then according to

the definition of *partition hamming weight*, $phw(x) + phw(M \times x^T)$ denotes the minimum number of active S-boxes for the operations of $\mathtt{SB} \circ \mathtt{SC}$. Therefore, the *partition branch number* defined above can represent the the minimum number of active S-boxes through two rounds operations of $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB} \circ \mathtt{MC} \circ \mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$.

According to the difference distribution table (DDT) of the 6-bit S-box used in SPEEDY, the maximum differential probability is 2^{-3}. Therefore, the maximum expected differential probability of differential trails over two rounds of SPEEDY can be estimated as $(2^{-3})^{pbn}$. It is noteworthy that based on *partition branch number*, we can directly evaluate the minimum number of active S-boxes. Compared to the traditional branch number which is used to evaluate the minimum number of active $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$, we can obtain more accurate results about the maximum differential probability of SPEEDY.

4.1 Computation of *pbn*

In order to compute the value of pbn, the main problem is to compute the *partition hamming weight* of a binary array. For an arbitrary binary array x, we can use a recursive algorithm to compute the value of $phw(x)$. First of all, we choose all the non-zero bits to form a set. Then we use a recursive algorithm to traverse all the possible partitions and check whether the condition is satisfied for each partition. Finally, the minimum number of active sets for all valid partitions is output as $phw(x)$.

Denote an l-bit binary array as $x = [x_0, x_1, \ldots x_{l-1}]$, and choose the indices of all non-zero bits in order to form a set $\{i_1, i_2, \ldots, i_j\}$, where $0 \leq i_1 < i_2 < \ldots < i_j \leq l-1$ and $x_{i_1} = x_{i_2} = \ldots = x_{i_j} = 1$. Moreover, since SC is a bit rotation operation, the indexes $i_1, i_2, \ldots, i_j$ can be considered as modular l in the computation of maximum distance. For example, for the binary array displayed above after Definition 1, the index set formed by all non-zero bits is $\{0, 1, 5, 9, 15, 21, 26\}$.

In order to compute the *partition hamming weight*, we present the following Algorithm 1 to traverse and check all the valid partitions for the set $\{i_1, i_2, \ldots, i_j\}$. Since there are j elements in the set, the following trivial partition with j subsets is always valid.

$$\{i_1\}, \{i_2\}, \ldots, \{i_j\}$$

Therefore, we can start to traverse and check if there is a valid partition with $j-1$ subsets. If we obtain a valid partition, then we can abort the traversal and go on to check partitions with $j-2$ subsets and so on. The algorithm stops when all the possible partitions with n $(1 \leq n \leq j-1)$ subsets are invalid and the output is $phw = n+1$. As an exceptional case, if the partition with $n = 1$ namely the set $\{i_1, i_2, \ldots, i_j\}$ is also valid, then the output is $phw = 1$.

Algorithm 1 Computation of partition hamming weight

Require: a set with j elements, $\{i_1, i_2, \ldots, i_j\}$
Ensure: phw
1: **for** $n = j-1$ to 1 **do**
2: $valid = TraverseCheck(n)$
3: **if** $valid$ is FALSE **then return** $phw = n+1$
4: **end if**
5: **end for**
6: **return** $phw = 1$

For partitions with n subsets, we use the following recursive function $TraverseCheck$ to traverse and check if there is a valid partition. According to the definition of phw, a partition is valid if it satisfies the condition that the maximum distance of each subset does not exceed 5. Moreover, since MC is a cyclic binary matrix, without loss of generality, all the partitions can start from the first element i_1. Therefore, the set $\{i_1, i_2, \ldots, i_j\}$ can be partitioned into n subsets in order, and each subset contains $s_1, s_2, \ldots, s_n$ elements, where $s_1+s_2+\ldots+s_n = j$. Note that all the subsets should not be empty, namely $1 \leq s_1, s_2, \ldots, s_n \leq j$. Considering that the element in each set is just the index of active bit, the maximum distance of each subset can be simply computed as the subtraction of the last element and the first element in the subset. For example, the maximum distance of the first subset is equal to $i_{s_1} - i_1$, and the maximum distance of the second subset is equal to $i_{s_1+s_2} - i_{s_1+1}$, etc.

Algorithm 2 $TraverseCheck$

Require: n, j, $\{i_1, i_2, \ldots, i_j\}$
Ensure: $valid$
1: Initialize an array with n elements: subset[n]
2: Set global variable $valid \leftarrow$ FALSE
3: **for** $i = 1$ to j **do**
4: subset[0] = i
5: $Traverse(1, n, j)$
6: **end for**
7: **return** $valid$

Algorithm 3 $Traverse$

Require: i, n, j
1: **if** $i < n$ **then**
2: **for** $l = 1$ to j **do**
3: subset[i] = l
4: $sum \leftarrow 0$
5: **for** $k = 0$ to i **do**
6: $sum +=$ subset[i]
7: **end for**
8: **if** $sum > j$ **then**
9: break
10: **end if**
11: $Traverse(i + 1, n, j)$;
12: **end for**
13: **else**
14: $flag \leftarrow$ TRUE
15: $k \leftarrow 0$
16: **for** $l = 0$ to $n - 1$ **do**
17: **if** $(i_{k+subset[l]-1} - i_k) > 5$ **then**
18: $flag$ = FALSE
19: **end if**
20: $k = k$+subset[l]
21: **end for**
22: **if** $flag$ is TRUE **then**
23: $valid$ = TRUE
24: **end if**
25: **end if**
26: **return**

4.2 Higher-order partition branch number

Similarly, in order to evaluate the minimum number of active S-boxes over several rounds, we can further define the notation of *higher-order partition branch number* pbn_r. For example, the *higher-order partition branch number* for 3-round SPEEDY can be computed as follows:

$$pbn_3 = \min_{\substack{i_{10},i_{11},i_{20},i_{21},i_{30},i_{31}\neq 0 \\ R[i_{10}][i_{11}]=R[i_{20}][i_{21}]=R[i_{30}][i_{31}]=1 \\ H[i_{11}][i_{20}]=H[i_{21}][i_{30}]=1}} i_{10}+i_{11}+i_{20}+i_{21}+i_{30}+i_{31}$$

where H is a binary table such that the element in the position $H[i][j]$ is 1 if and only if there is an $x \in F_2^l \backslash \{0\}$ with $hw(x) = i$ and $hw(M \times x^T) = j$, and R is a binary table such that the element in the position $R[i][j]$ is 1 if and only if there is an $x \in F_2^l \backslash \{0\}$ with $hw(x) = i$ and $phw(x) \leq j \leq hw(x)$. Namely, the table R records all the possible differential trails over SB $\circ$ SC $\circ$ SB where the second layer of SB can be n-bit to 1-bit differential.

Based on the above definition, we can compute the *higher-order partition branch number* for 3–7 rounds SPEEDY and the results are as follows:

$$pbn_3 = 21, \quad pbn_4 = 31, \quad pbn_5 = 41, \quad pbn_6 = 51, \quad pbn_7 = 61.$$

Considering that the best differential probability of the 6-bit S-box is 2^{-3}, the maximum probability of differential trails over 3–7 rounds is estimated to be smaller than 2^{-63}, 2^{-93}, 2^{-123}, 2^{-153} and 2^{-183}.

Notice that in the above computation, we only consider the hamming weight of a binary array x, which is rather imprecise especially taking into account the operation MC between two rounds. In order to improve the accuracy, we can precompute a table $R2$ to record all the possible differential trails over 2-round SPEEDY (namely SB $\circ$ SC $\circ$ SB $\circ$ MC $\circ$ SB $\circ$ SC $\circ$ SB). The table $R2$ is first initialized with 0, and then the element in the position $R2[i][j]$ is updated with the minimum value of $(hw(x) + hw(M \times x^T) + i + j)$ if and only if there is an $x \in F_2^l \backslash \{0\}$ with $phw(x) \leq i \leq hw(x)$ and $phw(M \times x^T) \leq j \leq hw(M \times x^T)$. Namely, the element in $R2[i][j]$ stores the minimal number of active S-boxes for all the possible differential trails over 2-round where the hamming weight of input difference is i and the hamming weight of output difference is j.

Then, we can compute the value of *higher-order partition branch number* more accurately. For even rounds, it can be computed as follows:

$$pbn_4 = \min_{\substack{i_{10},i_{11},i_{20},i_{21}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0, R2[i_{20}][i_{21}]\neq 0 \\ H[i_{11}][i_{20}]=1}} R2[i_{10}][i_{11}] + R2[i_{20}][i_{21}]$$

$$pbn_6 = \min_{\substack{i_{10},i_{11},i_{20},i_{21},i_{30},i_{31}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0, R2[i_{20}][i_{21}]\neq 0, R2[i_{30}][i_{31}]\neq 0 \\ H[i_{11}][i_{20}]=H[i_{21}][i_{30}]=1}} R2[i_{10}][i_{11}] + R2[i_{20}][i_{21}] + R2[i_{30}][i_{31}]$$

For odd rounds, we can combine the two tables R and $R2$ together, and the value of *higher-order partition branch number* is computed as follows:

$$pbn_3 = \min_{\substack{i_{10},i_{11},i_{20},i_{21}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0 \\ H[i_{11}][i_{20}]=1 \\ R[i_{20}][i_{21}]=1}} R2[i_{10}][i_{11}] + i_{20} + i_{21}$$

$$pbn_5 = \min_{\substack{i_{10},i_{11},i_{20},i_{21},i_{30},i_{31}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0, R2[i_{20}][i_{21}]\neq 0 \\ H[i_{11}][i_{20}]=H[i_{21}][i_{30}]=1 \\ R[i_{30}][i_{31}]=1}} R2[i_{10}][i_{11}] + R2[i_{20}][i_{21}] + i_{30} + i_{31}$$

Note that the *partition branch number* pbn is the same as pbn_2 defined as

$$pbn_2 = \min_{\substack{i_{10}, i_{11} \neq 0 \\ R2[i_{10}][i_{11}] \neq 0}} R2[i_{10}][i_{11}]$$

Based on the above formulas of *higher-order partition branch number*, we can obtain more accurate results about the minimum number of active 6-bit S-boxes for 3–7 rounds SPEEDY as follows:

$$pbn_2 = 13, \quad pbn_3 = 23, \quad pbn_4 = 35, \quad pbn_5 = 45, \quad pbn_6 = 57, \quad pbn_7 = 67.$$

Considering that the best differential probability of the S-box is 2^{-3}, the maximum probability of differential trails over 2–7 rounds SPEEDY is estimated to be smaller than 2^{-39}, 2^{-69}, 2^{-105}, 2^{-135}, 2^{-171} and 2^{-201}. It is noteworthy that these results are significantly higher than the maximum expected differential probabilities estimated by the designers in [15].

Moreover, these results are also consist with the search results of minimum number of active S-boxes obtained in [15]. By considering that there are at most 8 active words (of 6-bit) per state, they searched for the minimum number of active S-boxes for up to 4 rounds and the result are 13, 23 and 35 for 2, 3 and 4 rounds. Compared to the time-consuming search method, our computation of *higher-order partition branch number* can directly evaluate the minimum number of active S-boxes over several rounds more efficiently. Moreover, it can obtain accurate results for arbitrary rounds while automatic search may be infeasible for long rounds.

5 Concrete differential trails

In the above analysis, we always assume that all the differential transitions through the S-box are possible. However, when taking the actual Difference Distribution Table (DDT) of 6-bit S-box used in SPEEDY into consideration, the expected differential trail with maximum differential probability may not exist. Therefore, in order to verify the effectiveness of *higher-order partition branch number* obtained above and evaluate the security margin of SPEEDY more precisely, we will study concrete differential trails which satisfy the differential distribution table and compute its maximum differential probability.

In this section, we first present a special kind of 1-bit to 1-bit differential trails for 2–7 rounds SPEEDY and evaluate the corresponding concrete differential probabilities. Then, based on the computation of *higher-order partition branch number*, we present several effective n-bit concrete differential trails for 2–5.5 rounds SPEEDY and evaluate the maximum differential probabilities.

5.1 Concrete 1-bit to 1-bit differential trails for SPEEDY

First of all, according to the analysis of MC having branch number $bn = 8$ and the maximum differential probability of 1-bit to 1-bit transitions over SB ∘ SC ∘ SB is 2^{-6}, the maximum expected differential probability over two rounds of SPEEDY is supposed to be $(2^{-6})^8 = 2^{-48}$. However, we can prove that there is no possible 1-bit to 1-bit transition with such maximum differential probability.

Similar to the notation used in [15], we also use a table $T_1[i, j]$ to present the 1-bit to 1-bit differential probabilities of the SPEEDY S-box, and a table $T_2[i, j]$ to present the 1-bit to 1-bit differential probabilities over SB ∘ SC ∘ SB.

Table 2 1-bit to 1-bit differential probabilities of 6-bit S-box (T_1) and SB ∘ SC ∘ SB (T_2)

	T_1: ($\times 2^{-5}$)						T_2: ($\times 2^{-10}$)					
$i \setminus j$	0	1	2	3	4	5	0	1	2	3	4	5
0	–	1	3	2	1	1	4	6	9	9	6	3
1	4	3	4	4	–	–	12	12	12	12	12	4
2	1	1	3	3	1	1	4	9	9	9	9	3
3	1	3	–	2	3	–	12	9	12	12	6	3
4	2	2	4	4	2	1	8	12	12	12	12	4
5	2	4	2	4	–	2	16	12	16	16	12	4

Table 3 Probabilities of 1-bit to 1-bit differential trails for 2-round SPEEDY (T_4)

T_4						
$i \setminus j$	0	1	2	3	4	5
0	$2^{-50.42}$	$2^{-52.32}$	$2^{-50.42}$	$2^{-50.42}$	$2^{-52.32}$	$2^{-63.42}$
1	2^{-50}	$2^{-51.32}$	2^{-50}	2^{-50}	$2^{-51.32}$	$2^{-62.42}$
2	$2^{-50.42}$	$2^{-51.74}$	$2^{-50.42}$	$2^{-50.42}$	$2^{-51.74}$	$2^{-62.83}$
3	$2^{-50.42}$	$2^{-51.74}$	$2^{-50.42}$	$2^{-50.42}$	$2^{-51.74}$	$2^{-62.83}$
4	2^{-50}	$2^{-51.32}$	2^{-50}	2^{-50}	$2^{-51.32}$	$2^{-62.42}$
5	2^{-50}	$2^{-51.32}$	2^{-50}	2^{-50}	$2^{-51.32}$	$2^{-62.42}$

$$T_2[i, j] = \max_k T_1[i, k] \cdot T_1[k, j]$$

The results of (T_1, T_2) are listed in Table 2. We find that for most of the entries in T_2, there is a unique solution of k to achieve the maximum differential probability. For example, for the entry $T_2[5, 2] = 16$, the only possible transition is $T_1[5, 1] \cdot T_1[1, 2]$. There are only seven entries which have 2 or 3 solutions of k, and for simplicity we can choose the first solution as the transition.

Moreover, since SC and MC do not change the column position of active bits, the 1-bit to 1-bit differential trail over two rounds of SPEEDY can be illustrated as the following connection:

$$\mathrm{SB} \circ \mathrm{SC} \circ \mathrm{SB} \xrightarrow{\mathrm{MC}} \mathrm{SB} \circ \mathrm{SC} \circ \mathrm{SB}.$$

Based on the branch number of MC, there are at least 8 active 1-bit to 1-bit transitions over SB ∘ SC ∘ SB. All the possible patterns satisfy that there is one active SB ∘ SC ∘ SB in the first round and seven active SB ∘ SC ∘ SB in the second round. Therefore, we can define a table T_4 to store the maximum differential probability of the above connection, where each entry $T_4[i, j]$ denotes the maximum possible differential probability that the active input bits in the column i transit to active output bits in the column j. Then, the values of $T_4[i, j]$ can be computed by the following equation:

$$T_4[i, j] = \max_k T_2[i, k] \cdot (T_2[k, j])^7.$$

The results of T_4 are listed in Table 3. We can see that the maximum probability of concrete 1-bit to 1-bit differential trail over two rounds of SPEEDY is 2^{-50}, which is slightly lower than the estimated maximum probability of 2^{-48}.

Table 4 Probabilities of 1-bit to 1-bit differential trails for 3-round SPEEDY (T_6)

T_6						
$i \setminus j$	0	1	2	3	4	5
0	$2^{-82.49}$	$2^{-84.40}$	$2^{-82.49}$	$2^{-82.49}$	$2^{-84.40}$	$2^{-92.32}$
1	$2^{-82.08}$	$2^{-83.40}$	$2^{-82.08}$	$2^{-82.08}$	$2^{-83.40}$	$2^{-91.32}$
2	$2^{-82.49}$	$2^{-83.81}$	$2^{-82.49}$	$2^{-82.49}$	$2^{-83.81}$	$2^{-91.74}$
3	$2^{-82.49}$	$2^{-83.81}$	$2^{-82.49}$	$2^{-82.49}$	$2^{-83.81}$	$2^{-91.74}$
4	$2^{-82.08}$	$2^{-83.40}$	$2^{-82.08}$	$2^{-82.08}$	$2^{-83.40}$	$2^{-91.32}$
5	$2^{-82.08}$	$2^{-83.40}$	$2^{-82.08}$	$2^{-82.08}$	$2^{-83.40}$	$2^{-91.32}$

Similarly, we can analyze the maximum concrete probability of 1-bit to 1-bit differential trails for 3–7 rounds SPEEDY. Based on the above analysis, in order to keep the active columns as few as possible, all the active SB ∘ SC ∘ SB in one round should transit to the same active column. Moreover, since SC and MC are both column-rotation equivalent, the number of active bits in each column will not be affected by SC. Therefore, the minimum number of active 1-bit to 1-bit transitions over SB ∘ SC ∘ SB for r-round can be computed as:

$$\min_{x \in F_2^l \setminus \{0\}} \sum_{i=1}^{r} hw(M^{i-1} \times x^T)$$

where M is the corresponding binary matrix of MC and M^0 denotes the identity matrix I. By traversing all the possible values of x, we can know that the results of minimum number of active 1-bit to 1-bit transitions over SB ∘ SC ∘ SB for 3–7 rounds are 13, 32, 36, 48 and 56 respectively. Then, similar to the computation of T_4, we can traverse all the possible patterns of x minimizing the above equation and compute the maximum concrete probability of 1-bit to 1-bit differential trails for 3–7 rounds. For example, the table T_6 stores the maximum concrete probabilities of 1-bit to 1-bit differential trails for 3-round SPEEDY, and the value of each entry $T_6[i, j]$ can be computed as follows.

$$T_6[i, j] = \max_{k1,k2} (T_2[i, k1])^{hw(x)} \cdot (T_2[k1, k2])^{hw(M \cdot x)} \cdot (T_2[k2, j])^{hw(M^2 \cdot x)}.$$

Finally, the results of T_6 are listed in Table 4, and more results of T_8-T_{14} for 4–7 rounds are listed in Table 9 in Appendix A.

To sum up, for this special kind of 1-bit to 1-bit differential trails, the maximum concrete probabilities for 2–7 rounds are 2^{-50}, $2^{-82.08}$, $2^{-196.49}$, $2^{-229.96}$, $2^{-306.60}$ and $2^{-356.60}$. Moreover, in order to provide experimental verification, we present example differential trails for 2–7 rounds SPEEDY in Table 5. The input difference of each round is represented column-wise in hexadecimal and R/R$_{-1}$ denotes one complete round and the final round respectively.

5.2 Concrete *n*-bit differential trails for SPEEDY

In the above analysis of 1-bit to 1-bit differential trails, we have only considered 1-bit to 1-bit differential transitions over the 6-bit S-box. In this section, we present concrete differential trails which can take n-bit differential transitions over the S-box with $1 \leq n \leq 6$. According to the analysis in Sect. 4, the least number of active S-boxes for this kind of differential trails

Table 5 Example 1-bit to 1-bit differential trails for 2–7 rounds SPEEDY

Rounds	Propagation of the differential trail	Prob.
2	$(0, 80000000, 0, 0, 0, 0) \xrightarrow{R} (0, 0, 0, 0, 0, 42082230)$ $\xrightarrow{R_{-1}} (84104460,0,0,0,0,0)$	2^{-50}
3	$(0, 80000000, 0, 0, 0, 0) \xrightarrow{R} (0, 0, 0, 0, 0, 42082230)$ $\xrightarrow{R} (0, 0, 14000540, 0, 0, 0) \xrightarrow{R_{-1}} (28000a80, 0, 0, 0, 0, 0)$	$2^{-82.08}$
4	$(0, 0, 0, 0, 0, ff753c44) \xrightarrow{R} (00000080, 0, 0, 0, 0, 0)$ $\xrightarrow{R} (0, 0, 0, 0, 0, 2088c108) \xrightarrow{R} (0, 0, 0, 00150050, 0, 0)$ $\xrightarrow{R_{-1}} (002a00a0, 0, 0, 0, 0, 0)$	$2^{-196.49}$
5	$(0, 80080000, 0, 0, 0, 0) \xrightarrow{R} (0, 0, 0, 0, 0, 610c02b2)$ $\xrightarrow{R} (0, 0, 0, 40014540, 0, 0) \xrightarrow{R} (0, 0, 0, 909c275c, 0, 0)$ $\xrightarrow{R} (0, 0, 0, 01010011, 0, 0) \xrightarrow{R_{-1}} (02020022, 0, 0, 0, 0, 0)$	$2^{-229.96}$
6	$(0, 80000000, 0, 0, 0, 0) \xrightarrow{R} (0, 0, 0, 0, 0, 42082230)$ $\xrightarrow{R} (0, 0, 0, 14000540, 0, 0) \xrightarrow{R} (0, 0, 0, 6f8adff1, 0, 0)$ $\xrightarrow{R} (0, 0, 0, 01011000, 0, 0) \xrightarrow{R} (0, 0, 0, 90514e0a, 0, 0)$ $\xrightarrow{R_{-1}} (40a29c15, 0, 0, 0, 0, 0)$	$2^{-306.60}$
7	$(0, 0000000a, 0, 0, 0, 0) \xrightarrow{R} (0, 0, 0, 0, 0, 28a2abc5)$ $\xrightarrow{R} (0, 0, 0, 10004101, 0, 0) \xrightarrow{R} (0, 0, 0, 468680d3, 0, 0)$ $\xrightarrow{R} (0, 0, 0, 14154000, 0, 0) \xrightarrow{R} (0, 0, 0, 4451d443, 0, 0)$ $\xrightarrow{R} (0, 0, 0, 44010141, 0, 0) \xrightarrow{R_{-1}} (88020282, 0, 0, 0, 0, 0)$	$2^{-356.60}$

can be reduced significantly. Therefore, we can construct effective differential trails with much higher probability. Based on the result of *higher-order partition branch number* pbn_r, we can obtain the least number of active S-boxes for r-round. Then by traversing all the possible patterns and taking the actual Difference Distribution Table (DDT) of the S-box into consideration, we can get concrete differential trail with maximum differential probability for r-round SPEEDY.

5.2.1 2-Round differential trail

According to the analysis of *partition branch number* $pbn_2 = 13$, there are at least 13 active S-boxes over two rounds. Moreover, based on the analysis in Sect. 4.2, in order to compute the value of pbn, we precompute a table $R2[i][j]$ to store the minimal number of active S-boxes for all the possible differential trails over 2-round SPEEDY, where the hamming weight of input difference is i and the hamming weigh of output difference is j. Then, by checking all the entries of table $R2$, we can find that there is only one entry satisfying $R2[1][4] = 13$. Namely, the only possible pattern of 2-round differential trail with 13 active S-boxes satisfies that there is only one active bit in the input difference and 4 active S-boxes in the output difference.

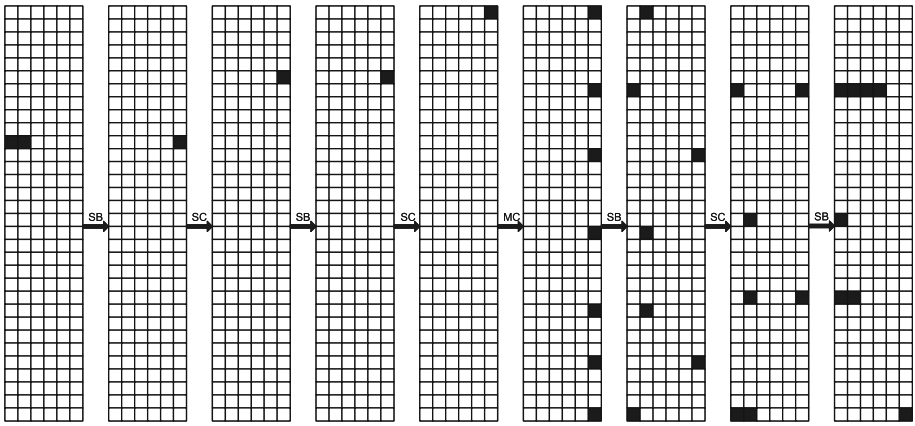

Fig. 2 Concrete differential trail for 2-round SPEEDY

Therefore, by traversing all the possible patterns and taking the actual difference distribution table of the S-box into consideration, we can get the maximum differential probability of 2-round differential trail for SPEEDY. In Fig. 2, we present an example differential trail for 2-round SPEEDY, where the black box denotes '1' difference and the empty box denotes '0' difference. There are 13 active S-boxes and the concrete differential probability is equal to $2^{-3.4} \times 2^{-4} \times (2^{-4})^5 \times (2^{-3})^4 \times (2^{-3.4})^2 \approx 2^{-46.2}$. Obviously, it is much better than the concrete 1-bit to 1-bit differential trail for 2-round SPEEDY presented in Sect. 5.1.

5.2.2 3-Round differential trail

Similarly, according to the analysis of $pbn_3 = 23$, there are at least 23 active S-boxes over three rounds. Moreover, based on the definition of pbn_3,

$$pbn_3 = \min_{\substack{i_{10},i_{11},i_{20},i_{21}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0 \\ H[i_{11}][i_{20}]=1 \\ R[i_{20}][i_{21}]=1}} R2[i_{10}][i_{11}] + i_{20} + i_{21}$$

we traverse all the possible values of $i_{10}, i_{11}, i_{20}, i_{21}$ and find that there is only one solution satisfying 23, namely $i_{10} = 1, i_{11} = 7, i_{20} = 5, i_{21} = 2$. Note that in this pattern the first two rounds $R2[1][7]$ just represents a 1-bit to 1-bit differential trail. Therefore, we can exploit the results of concrete 1-bit to 1-bit differential probabilities of 2-round SPEEDY in Table 3. By checking all the possible connections of 2-round 1-bit to 1-bit differential trail with 1-round n-bit differential trail satisfying the given pattern, we can get the maximum differential probability of concrete 3-round differential trail for SPEEDY.

In Fig. 3, we present an example of the best differential trail for 3-round SPEEDY, where the black box denotes '1' difference and the empty box denotes '0' difference. There are 23 active S-boxes and the first 2-round (2R) is 1-bit to 1-bit differential trail corresponding to $T_4[1, 4]$ in Table 3 whose differential probability is $2^{-51.32}$. Therefore, the total probability of the 3-round differential trail is equal to $2^{-51.32} \times (2^{-3})^2 \times (2^{-4})^3 \times 2^{-4} \times 2^{-3.4} \approx 2^{-76.72}$. As a contrast, the best differential probability of concrete 1-bit to 1-bit differential trail for 3-round SPEEDY is only $2^{-82.08}$ as listed in Table 4.

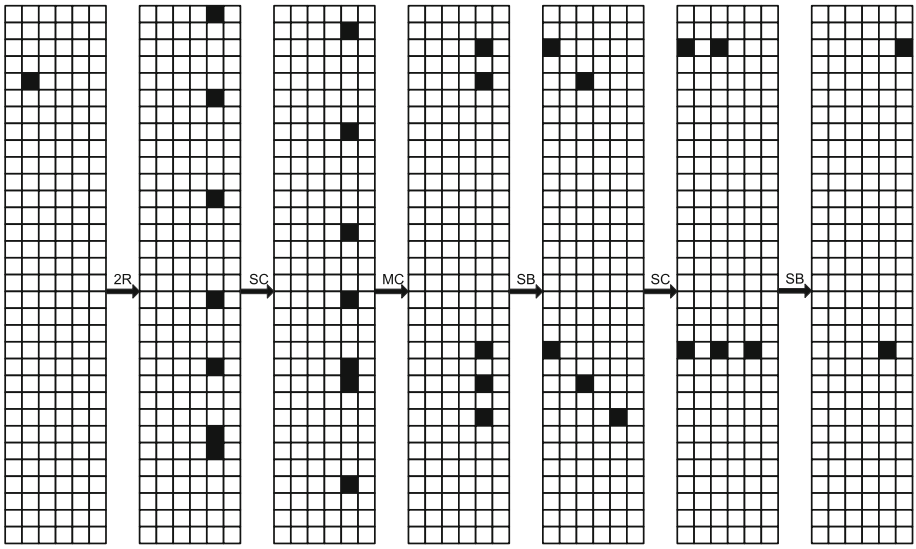

Fig. 3 Concrete differential trail for 3-round SPEEDY

5.2.3 4-Round differential trail

There are at least 35 active S-boxes for 4-round SPEEDY since $pbn_4 = 35$,

$$pbn_4 = \min_{\substack{i_{10},i_{11},i_{20},i_{21}\neq 0 \\ R2[i_{10}][i_{11}]\neq 0 \\ R2[i_{20}][i_{21}]\neq 0 \\ H[i_{11}][i_{20}]=1}} R2[i_{10}][i_{11}] + R2[i_{20}][i_{21}]$$

Unfortunately, there are only four solutions of $i_{10}, i_{11}, i_{20}, i_{21}$ satisfying there are 35 active S-boxes, and by traversing all the possible patterns we find that it is impossible to construct a concrete differential trail when the actual difference distribution table of the S-box is considered. Therefore, we further check patterns with the almost least number of active S-boxes. By traversing all the possible patterns with 36 and 37 active S-boxes, we can get the maximum differential probability of concrete differential trails for 4-round SPEEDY.

In Table 6, we present an example of the best concrete differential trail for 4-round SPEEDY. The input and output differences of each operation are expressed column-wise in hexadecimal as follows. The total probability of this 4-round differential trail is $2^{-129.2}$. Obviously, it is significantly higher than the best concrete 1-bit to 1-bit 4-round differential trail listed in Table 5, whose probability is $2^{-196.49}$ exceeding the bound of 192-bit security already.

5.2.4 5-Round differential trail

Similar to the situation of 4-round concrete differential trail, there is also no possible 5-round differential trail satisfying $pbn_5 = 45$ when the actual difference distribution table of the S-box is considered. Therefore, we further check all the possible patterns with 46-48 active S-boxes and search for the best concrete differential trail for 5-round SPEEDY. As a result, we present an example of the best concrete differential trail for 5-round SPEEDY in Table 7.

Table 6 Concrete differential trail for 4-round SPEEDY

Rounds	Differential propagations for each round	Prob.
Input	(00200000, 00200000, 0, 0, 0, 0)	–
1	$\xrightarrow{\text{SB}}$ (0, 0, 0, 0, 0, 00200000) $\xrightarrow{\text{SC}}$ (0, 0, 0, 0, 0, 04000000)	$2^{-4} \times 2^{-3.4}$
	$\xrightarrow{\text{SB}}$ (0, 0, 0, 0, 0, 04000000) $\xrightarrow{\text{SC}}$ (0, 0, 0, 0, 0, 80000000)	
	$\xrightarrow{\text{MC}}$ (0,0,0,0,0,82104111)	
2	$\xrightarrow{\text{SB}}$ (0, 82104111, 0, 0, 0, 0) $\xrightarrow{\text{SC}}$ (0, 04208223, 0, 0, 0, 0)	$(2^{-3})^7$
	$\xrightarrow{\text{SB}}$ (0, 0, 0, 04208223, 0, 0) $\xrightarrow{\text{SC}}$ (0, 0, 0, 21041118, 0, 0)	$\times(2^{-3})^7$
	$\xrightarrow{\text{MC}}$ (0,0,0,00a0002a,0,0)	
3	$\xrightarrow{\text{SB}}$ (00800002, 0, 0, 00200020, 00000008, 0)	$(2^{-5})^2 \times (2^{-4})^2$
	$\xrightarrow{\text{SC}}$ (00800002, 0, 0, 01000100, 00000080, 0)	$\times 2^{-3.4}$
	$\xrightarrow{\text{SB}}$ (0, 0, 0, 01800182, 0, 0) $\xrightarrow{\text{SC}}$ (0, 0, 0, 0c000c10, 0, 0)	$\times(2^{-4})^4 \times 2^{-3}$
	$\xrightarrow{\text{MC}}$ (0,0,0,50203018,0,0)	
4	$\xrightarrow{\text{SB}}$ (0, 40200000, 0, 10002010, 00001008, 0)	$(2^{-3.4})^4 \times (2^{-4})^3$
	$\xrightarrow{\text{SC}}$ (0,80400000,0,80010080,00010080,0)	$\times 2^{-3} \times 2^{-4}$
	$\xrightarrow{\text{SB}}$ (00400000,0,0,00010080,80010080,0)	$\times(2^{-3.4})^2$
Total		$2^{-129.2}$

The input difference and output difference can be expressed column-wise in hexadecimal as follows.

$$(00200000, 00200000, 0, 0, 0, 0) \xrightarrow{5\text{R}} (0, 00001000, 0, 01001000, 01000040, 0)$$

The total probability of this 5-round differential trail is $2^{-170.0}$. It is noteworthy that this is still higher than 2^{-192} and hence it can be used to mount a key-recovery attack on SPEEDY-7-192.

5.2.5 Differential trails for more rounds

We have also searched for the optimal differential trails for more rounds by considering the least number of active S-boxes ($pbn_6 = 57$ and $pbn_7 = 67$). Unsurprisingly, there is no valid differential trail for 6-round or 7-round SPEEDY with probability higher than 2^{192}, when considering the difference distribution table of S-box. For simplicity, we only present input/output differences and the probabilities of 6-round and 7-round differential trails we have obtained.

The input and output differences of 6-round differential trail can be expressed column-wise in hexadecimal as follows, and the probability is $2^{-216.0}$.

$$(00200000, 00200000, 0, 0, 0, 0) \xrightarrow{6\text{R}} (0, 08200000, 0, 08000200, 0, 00200200)$$

Similarly, we can extend to get the following 7-round differential trail and its differential probability is $2^{-266.2}$.

Table 7 Concrete differential trail for 5-round SPEEDY

Rounds	Differential propagations for each round	Prob.
Input	(00200000, 00200000, 0, 0, 0, 0)	–
1	$\xrightarrow{SB}$ (0, 0, 0, 0, 0, 00200000) $\xrightarrow{SC}$ (0, 0, 0, 0, 0, 04000000)	$2^{-4} \times 2^{-3.4}$
	$\xrightarrow{SB}$ (0, 0, 0, 0, 0, 04000000) $\xrightarrow{SC}$ (0, 0, 0, 0, 0, 80000000)	
	$\xrightarrow{MC}$ (0,0,0,0,0,82104111)	
2	$\xrightarrow{SB}$ (0, 82104111, 0, 0, 0, 0) $\xrightarrow{SC}$ (0, 04208223, 0, 0, 0, 0)	$(2^{-3})^7$
	$\xrightarrow{SB}$ (0, 0, 0, 04208223, 0, 0) $\xrightarrow{SC}$ (0, 0, 0, 21041118, 0, 0)	$\times(2^{-3})^7$
	$\xrightarrow{MC}$ (0,0,0,00a0002a,0,0)	
3	$\xrightarrow{SB}$ (00800002, 0, 0, 00200020, 00000008, 0)	$(2^{-5})^2 \times (2^{-4})^2$
	$\xrightarrow{SC}$ (00800002, 0, 0, 01000100, 00000080, 0)	$\times 2^{-3.4}$
	$\xrightarrow{SB}$ (0, 0, 0, 01800182, 0, 0) $\xrightarrow{SC}$ (0, 0, 0, 0c000c10, 0, 0)	$\times(2^{-4})^4 \times 2^{-3}$
	$\xrightarrow{MC}$ (0,0,0,50203018,0,0)	
4	$\xrightarrow{SB}$ (00000018, 10001000, 0, 0, 40202000, 0)	$(2^{-5})^2$
	$\xrightarrow{SC}$ (00000018,20002000,0,0,02020004,0)	$\times(2^{-3.4})^5$
	$\xrightarrow{SB}$ (0, 0, 0, 2202201c, 0, 0) $\xrightarrow{SC}$ (0, 0, 0, 101100e1, 0, 0)	$\times(2^{-3})^5 \times (2^{-4})^2$
	$\xrightarrow{MC}$ (0,0,0,0800110a,0,0)	
5	$\xrightarrow{SB}$ (00001000, 08000008, 0, 00000002, 00000100, 0)	$2^{-5} \times 2^{-4}$
	$\xrightarrow{SC}$ (00001000,10000010,0,00000010,00001000,0)	$\times(2^{-3.4})^3$
	$\xrightarrow{SB}$ (10000010,0,0,00001000,0,00001000)	$\times(2^{-4})^2 \times 2^{-3}$
Total		$2^{-170.0}$

$$(00200000, 00200000, 0, 0, 0, 0) \xrightarrow{7R} (0, 00000040, 04100040, 00000040, 0, 00010000)$$

6 Full-round related-key differential attack

Utilizing the above differential characteristic, together with the simple linear key schedule of SPEEDY, we can construct an efficient 6-round related-key differential trail and mount a key recovery attack on full-round SPEEDY-7-192.

The key schedule of SPEEDY receives a 192-bit master key and initializes it to the state of the round key (k_0). Then, it applies a simple bit permutation PB to compute the next round key. Contents of the bit-permutation PB are listed in Table 10 in Appendix A.

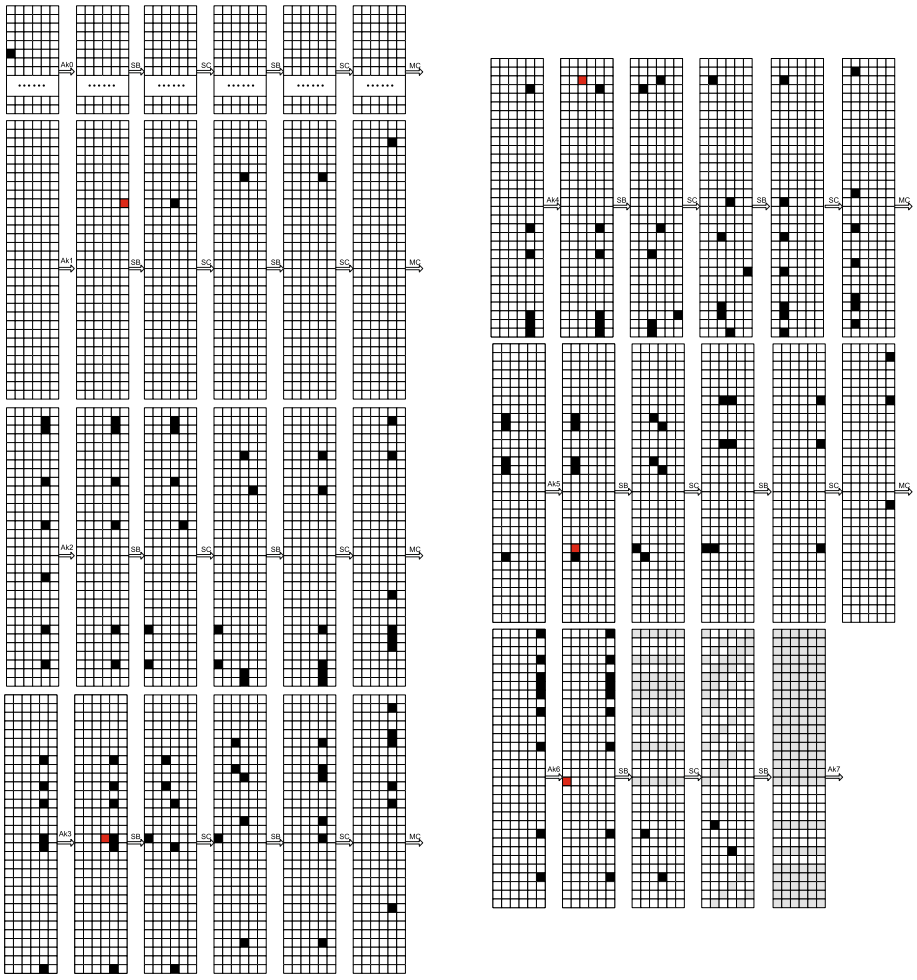

Fig. 4 6-Round related-key differential trail and full-round key recovery attack

6.1 Related-key differential trail

Based on the simple bit-permutation of key schedule, if we choose one active bit in the master key, then we can get a pair of related-key with only one active bit in each round. Moreover, all the subkey differences can be directly determined with probability one.

Therefore, by carefully choosing the position of active bit in master key, we can construct efficient related-key differential by trying best to cancel the input difference and control the difference propagation for each round. We combine the subkey difference with the differential trail search method in Sect. 5 and traverse all the possible positions of one active bit . Finally, we have obtained the following 6-round related-key differential trail and its probability is about $2^{-179.2}$. Its difference propagation is illustrated in Fig.4 in detail.

$$\begin{array}{l}\Delta X^0 = (\texttt{04000000}, 0, 0, 0, 0, 0) \\ \Delta K^0 = (\texttt{04000000}, 0, 0, 0, 0, 0)\end{array} \xrightarrow{\texttt{6R}} \Delta X^6 = (0, 0, 0, 0, 0, \texttt{97440108})$$

Table 8 Time complexity evaluation of the attack procedure

Row	Subkey guessed	Data filter	Time complexity	Row	Subkey guessed	Data filter	Time complexity
17	2^6	2^{-5}	2^6	5	2^6	2^{-2}	2^{13}
16	2^6	2^{-5}	2^7	4	2^6	$2^{-2} \cdot 2^{-6}$	2^{17}
15	2^6	2^{-5}	2^8	3	2^6	2^{-2}	2^{15}
14	2^6	2^{-5}	2^9	2	2^6	$2^{-2} \cdot 2^{-6}$	2^{19}
13	2^6	2^{-4}	2^{10}	1	2^6	$2^{-3} \cdot 2^{-6}$	2^{17}
12	2^6	$2^{-4} \cdot 2^{-6}$	2^{12}	0	2^6	$2^{-3} \cdot 2^{-6}$	2^{14}
11	2^6	2^{-5}	2^8	31	2^6	2^{-4}	2^{11}
10	2^6	2^{-5}	2^9	30	2^6	$2^{-4} \cdot 2^{-6}$	2^{13}
9	2^6	2^{-4}	2^{10}	29	2^6	2^{-5}	2^9
8	2^6	$2^{-4} \cdot 2^{-6}$	2^{12}	28	2^6	2^{-5}	2^{10}
7	2^6	2^{-4}	2^8	27	2^6	$2^{-5} \cdot 2^{-6}$	2^{11}
6	2^6	2^{-3}	2^{10}	Total			$2^{19.73}$

6.2 Full-round key recovery attack

We can extend the above 6-round related-key differential trail one round forwards to reach the full 7-round and recover the key used in the last round.

Note that the output difference ΔX^6 has only 9 active bits in the 5-th column and all the other bits are zero. In the last round, after the first subkey addition $\mathtt{Ak}_6$, there will be another active bit in the 0-th column. Then after the operations of $\mathtt{SB} \circ \mathtt{SC} \circ \mathtt{SB}$, all the rows of ΔC may be active. In order to enhance the filter of ciphertext pairs, we restrict the output differences of the first layer of SB in Rows 23 and 28 to be $0x10$ and $0x04$ respectively. Then the differential probability is reduced to $2^{-185.2}$, and the possible difference propagation of the last round is illustrated in Fig.4. We denote the empty box as '0' difference, the black box as '1' difference, the grey box as '?' difference, and the red box as key difference respectively.

The attack procedure can be described briefly as follows. First, choose randomly $2^{185.2}$ pairs of plaintexts with $\Delta P = (04000000, 0, 0, 0, 0, 0)$, and generate the corresponding ciphertext pairs under a pair of related-key with $\Delta K = (04000000, 0, 0, 0, 0, 0)$. Second, filter the ciphertext pairs with zero-difference conditions. There are seven inactive rows which means a filter probability of 2^{-42}. Hence, after this step, there remains about $2^{143.2}$ ciphertext pairs. Third, guess the value of subkey k_7 in Row 22 and 25 respectively, and partially decrypt through the second layer of SB to check if it satisfies the difference condition. The time complexity of this step is $2^{143.2} \cdot 2^6 + 2^{143.2} \cdot 2^{-6} \cdot 2^{12} = 2^{150.2}$, and after this step there remains about $2^{131.2}$ pairs. Last, guess the value of subkey k_7 row by row (from Row 17 to Row 0 and then Row 31 to Row 27) and partially decrypt through the second layer of SB to check if it satisfies the difference condition. Moreover, in some steps the filter condition of first layer of SB can bring another data filter probability of 2^{-6}. The data filter probability and time complexity of each step are listed in Table 8. The overall time complexity is about $2^{19.73} \cdot 2^{143.2} = 2^{162.93}$, and there remains about $2^{-6.8}$ pairs. Therefore, if there still remains a pair after all the filter steps, the corresponding subkey guess is correct.

To sum up, the data and time complexities of related-key differential attack on full-round SPEEDY are $2^{185.2} \times 2 = 2^{186.2}$ chosen-plaintexts and $2^{162.93} \div 7 \approx 2^{160.13}$ encryptions. We

have successfully recovered 150-bit keys, and the remaining 42-bit keys can be recovered simply by exhaustive search with negligible complexity.

7 Conclusion

In this paper, we present concrete differential analysis of SPEEDY based on some new observations on the branch number. First of all, we propose *(higher-order) partition branch number*, which can describe the minimum number of active S-boxes for SPEEDY more accurately. Then by utilizing an efficient algorithm to compute the value of pbn_r, we can obtain more accurate results about the minimum number of active S-boxes for 2–7 rounds SPEEDY. It is noteworthy that these results are significantly higher than the maximum expected differential probabilities estimated by the designers. Based on this, we search for optimal differential characteristics of SPEEDY while considering the difference distribution table of the 6-bit S-box. We present examples of optimal differential characteristics for 2–7 rounds SPEEDY, whose probabilities are $2^{-46.2}$, $2^{-76.72}$, $2^{-129.2}$, $2^{-170.0}$, $2^{-216.0}$ and $2^{-266.2.0}$ respectively.

Furthermore, by utilizing the simple bit-permutation key schedule of SPEEDY, we can extend the differential trail search method and construct an efficient 6-round related-key differential trail with probability $2^{-179.2}$. Based on it, we can present related-key differential attack on full round SPEEDY-7-192 with data complexity of $2^{186.2}$ chosen-plaintexts and time complexity of $2^{160.13}$ encryptions. Moreover, our work will provide new insights in evaluating the security against differential attack more accurately for block ciphers employing SPS-type round function with bit-wise rotation as the linear layer.

Acknowledgements This work is supported by the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-035), and National Natural Science Foundation of China (No. 62072445). Moreover, the author is very grateful to the anonymous reviewers for their helpful comments and suggestions.

Declarations

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A

See Tables 9 and 10

Table 9 Probabilities of 1-bit to 1-bit differential trails for 4–7 rounds SPEEDY

$i \setminus j$	0	1	2	3	4	5
T_8: 4-round						
0	$2^{-212.26}$	$2^{-213.58}$	$2^{-212.26}$	$2^{-212.26}$	$2^{-213.58}$	$2^{-221.50}$
1	$2^{-203.96}$	$2^{-205.28}$	$2^{-203.96}$	$2^{-203.96}$	$2^{-205.28}$	$2^{-213.21}$
2	$2^{-211.84}$	$2^{-213.16}$	$2^{-211.84}$	$2^{-211.84}$	$2^{-213.16}$	$2^{-221.09}$
3	$2^{-204.38}$	$2^{-205.70}$	$2^{-204.38}$	$2^{-204.38}$	$2^{-205.70}$	$2^{-213.62}$
4	$2^{-203.96}$	$2^{-205.28}$	$2^{-203.96}$	$2^{-203.96}$	$2^{-205.28}$	$2^{-213.21}$
5	$2^{-196.49}$	$2^{-197.81}$	$2^{-196.49}$	$2^{-196.49}$	$2^{-197.81}$	$2^{-205.74}$
T_{10}: 5-round						
0	$2^{-230.79}$	$2^{-232.45}$	$2^{-230.79}$	$2^{-230.79}$	$2^{-232.45}$	$2^{-238.79}$
1	$2^{-229.96}$	$2^{-230.94}$	$2^{-229.96}$	$2^{-229.96}$	$2^{-230.94}$	$2^{-237.28}$
2	$2^{-230.79}$	$2^{-231.77}$	$2^{-230.79}$	$2^{-230.79}$	$2^{-231.77}$	$2^{-238.11}$
3	$2^{-230.79}$	$2^{-231.77}$	$2^{-230.79}$	$2^{-230.79}$	$2^{-231.77}$	$2^{-238.11}$
4	$2^{-229.96}$	$2^{-230.94}$	$2^{-229.96}$	$2^{-229.96}$	$2^{-230.94}$	$2^{-237.28}$
5	$2^{-229.96}$	$2^{-230.94}$	$2^{-229.96}$	$2^{-229.96}$	$2^{-230.94}$	$2^{-237.28}$
T_{12}: 6-round						
0	$2^{-307.01}$	$2^{-308.26}$	$2^{-307.01}$	$2^{-307.01}$	$2^{-308.26}$	$2^{-325.70}$
1	$2^{-306.60}$	$2^{-307.85}$	$2^{-306.60}$	$2^{-306.60}$	$2^{-307.85}$	$2^{-325.28}$
2	$2^{-307.01}$	$2^{-308.26}$	$2^{-307.01}$	$2^{-307.01}$	$2^{-308.26}$	$2^{-325.70}$
3	$2^{-307.01}$	$2^{-308.26}$	$2^{-307.01}$	$2^{-307.01}$	$2^{-308.26}$	$2^{-325.70}$
4	$2^{-306.60}$	$2^{-307.85}$	$2^{-306.60}$	$2^{-306.60}$	$2^{-307.85}$	$2^{-325.28}$
5	$2^{-306.60}$	$2^{-307.85}$	$2^{-306.60}$	$2^{-306.60}$	$2^{-307.85}$	$2^{-325.28}$
T_{14}: 7-round						
0	$2^{-357.43}$	$2^{-359.09}$	$2^{-357.43}$	$2^{-357.43}$	$2^{-359.09}$	$2^{-368.60}$
1	$2^{-356.60}$	$2^{-358.26}$	$2^{-356.60}$	$2^{-356.60}$	$2^{-358.26}$	$2^{-367.77}$
2	$2^{-357.43}$	$2^{-359.09}$	$2^{-357.43}$	$2^{-357.43}$	$2^{-359.09}$	$2^{-368.60}$
3	$2^{-357.43}$	$2^{-359.09}$	$2^{-357.43}$	$2^{-357.43}$	$2^{-359.09}$	$2^{-368.60}$
4	$2^{-356.60}$	$2^{-358.26}$	$2^{-356.60}$	$2^{-356.60}$	$2^{-358.26}$	$2^{-367.77}$
5	$2^{-356.60}$	$2^{-358.26}$	$2^{-356.60}$	$2^{-356.60}$	$2^{-358.26}$	$2^{-367.77}$

Table 10 PB bit-permutation for SPEEDY-r-192

i	0	1	2	3	4	5	6	7	8	9	10	11
PB(i)	1	8	15	22	29	36	43	50	57	64	71	78
i	12	13	14	15	16	17	18	19	20	21	22	23
PB(i)	85	92	99	106	113	120	127	134	141	148	155	162
i	24	25	26	27	28	29	30	31	32	33	34	35
PB(i)	169	176	183	190	5	12	19	26	33	40	47	54
i	36	37	38	39	40	41	42	43	44	45	46	47
PB(i)	61	68	75	82	89	96	103	110	117	124	131	138
i	48	49	50	51	52	53	54	55	56	57	58	59
PB(i)	145	152	159	166	173	180	187	2	9	16	23	30
i	60	61	62	63	64	65	66	67	68	69	70	71
PB(i)	37	44	51	58	65	72	79	86	93	100	107	114
i	72	73	74	75	76	77	78	79	80	81	82	83
PB(i)	121	128	135	142	149	156	163	170	177	184	191	6
i	84	85	86	87	88	89	90	91	92	93	94	95
PB(i)	13	20	27	34	41	48	55	62	69	76	83	90
i	96	97	98	99	100	101	102	103	104	105	106	107
PB(i)	97	104	111	118	125	132	139	146	153	160	167	174
i	108	109	110	111	112	113	114	115	116	117	118	119
PB(i)	181	188	3	10	17	24	31	38	45	52	59	66
i	120	121	122	123	124	125	126	127	128	129	130	131
PB(i)	73	80	87	94	101	108	115	122	129	136	143	150
i	132	133	134	135	136	137	138	139	140	141	142	143
PB(i)	157	164	171	178	185	0	7	14	21	28	35	42
i	144	145	146	147	148	149	150	151	152	153	154	155
PB(i)	49	56	63	70	77	84	91	98	105	112	119	126
i	156	157	158	159	160	161	162	163	164	165	166	167
PB(i)	133	140	147	154	161	168	175	182	189	4	11	18
i	168	169	170	171	172	173	174	175	176	177	178	179
PB(i)	25	32	39	46	53	60	67	74	81	88	95	102
i	180	181	182	183	184	185	186	187	188	189	190	191
PB(i)	109	116	123	130	137	144	151	158	165	172	179	186

References

1. Avanzi R.: The QARMA block cipher family. IACR Trans. Symmetric Cryptol. **2017**(1), 4–44 (2017).
2. Banik S., Bao Z., Isobe T., Kubo H., Liu F., Minematsu K., Sakamoto K., Shibata N., Shigeri M.: WARP: revisiting GFN for lightweight 128-bit block cipher. In: Dunkelman O., et al. (eds.) SAC 2020, vol. 12804, pp. 535–564. LNCS. Springer, Heidelberg (2021).
3. Banik S., Bogdanov A., Isobe T., Shibutani K., Hiwatari H., Akishita T., Regazzoni F.: Midori: a block cipher for low energy. In: Iwata T., Cheon J.H. (eds.) ASIACRYPT 2015, vol. 9453, pp. 411–436. LNCS. Springer, Heidelberg (2015).
4. Banik S., Isobe T., Liu F., Minematsu K., Sakamoto K.: Orthros: a low-latency PRF. IACR Trans. Symmetric Cryptol. **2021**(1), 37–77 (2021).
5. Beaulieu R., Shors D., Smith J., Treatman-Clark S., Weeks B., Wingers L.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptology ePrint Archive 2013/404 (2013)

6. Beierle C., Leander G., Moradi A., Rasoolzadeh S.: CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. IACR Trans. Symmetric Cryptol. **2019**(1), 5–45 (2019).
7. Beierle C., Jean J., Kolbl 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 2016, vol. 9815, pp. 123–153. LNCS. Springer, Heidelberg (2016).
8. Bogdanov A., Knudsen L.R., Leander G., Parr C., Poschmann A., Robshaw M.J.B., Seurin Y., Vikkelsoe C.: PRESENT: an ultra-lightweight block cipher. In: Paillier P., Verbauwhed I. (eds.) CHES 2007, vol. 4727, pp. 450–466. LNCS. Springer, Heidelberg (2007).
9. Borghoff J., Canteaut A., Guneysu T., Kavun E.B., Knezevic M., Knudsen L.R., Leander G., Nikov V., Paar C., Rechberger C., Rombouts P., Thomsen S.S., Yalcm T.: PRINCE - a low-latency block cipher for pervasive computing applications. In: Wang X., Sako K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208-225. Springer (2011).
10. Boura C., David N., Heim Boissier R., Naya-Plasencia M.: Better steady than speedy: full break of SPEEDY-7-192. Cryptology ePrint Archive, Paper 2022/1351 (2022). https://eprint.iacr.org/2022/1351.
11. Boura C., David N., Heim Boissier R., Naya-Plasencia M.: Better steady than speedy: full break of SPEEDY-7-192. EUROCRYPT 2023. LNCS, vol. 14007, pp. 36-66. Springer, Heidelberg (2023).
12. Bozilov D., Eichlseder M., Knezevic M., Lambin B., Leander G., Moos T., Nikov V., Rasoolzadeh S., Todo Y., Wiemer F.: PRINCEv2: more security for (almost) no overhead. In: Dunkelman O., et al. (eds.) SAC 2020, vol. 12804, pp. 483–511. LNCS. Springer, Heidelberg (2021).
13. Canniere C.D., Dunkelman O., Knezevic M.: KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers. In: Clavier C., Gaj K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer (2009).
14. Dobraunig C., Eichlseder M., Kales D.: Practical key-recovery attack on MANTIS5. IACR Trans. Symmetric Cryptol. **2016**(2), 248–260 (2016).
15. Leander G., Moos T., Moradi A., Rasoolzadeh S.: The SPEEDY family of block ciphers. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(4), 510–545 (2021).
16. NIST: Advanced Encryption Standard (AES). FIPS PUB 197, National Institute of Standards and Technology (2001).
17. NXP: AN12278 LPC55S00 security solutions for IoT (2020). https://www.nxp.com/docs/en/application-note/AN12278.pdf.
18. Qualcomm Product Security: Pointer authentication on ARMv8.3—design and analysis of the new sotfware security instructions (2017). https://www.qualcomm.com/documents/whitepaper-pointer-authentication-armv83.
19. Rohit R., Sarkar S.: Cryptanalysis of reduced round SPEEDY. Africacrypt 2022. Cryptology ePrint Archive, Paper 2022/612 (2022). https://eprint.iacr.org/2022/612.
20. Shibutani K., Isobe T., Hiwatari H., Mitsuda A., Akishita T., Shirai T.: Piccolo: an ultra-lightweight blockcipher. In: Preneel B., Takagi T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342-357. Springer (2011).
21. Soleimany H., Blondeau C., Yu X., Wu W., Nyberg K., Zhang H., Zhang L., Wang Y.: Reflection cryptanalysis of PRINCE-like ciphers. J. Cryptol. **28**(3), 718–744 (2015).
22. Suzaki T., Minematsu K., Morioka S., Kobayashi E.: TWINE: a lightweight block cipher for multiple platforms. In: Knudsen L.R., Wu H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 339-354. Springer (2013).
23. Wu W., Zhang L.: LBlock: a lightweight block cipher. In: Lopez J., Tsudik G. (eds.) ACNS 2011, vol. 6715, pp. 327–344. LNCS. Springer, Heidelberg (2011).