EasyChair Preprint

№ 9356

Differential Cryptanalysis of Round-Reduced SPEEDY Family

Qingyuan Yu, Keting Jia, Guangnan Zou and Guoyan Zhang

November 23, 2022

Differential Cryptanalysis of Round-reduced SPEEDY Family

Qingyuan Yu[1,4][0000−0003−2814−5431], Keting Jia*[2,5][0000−0002−6396−8882], Guangnan Zou[3], and Guoyan Zhang[1,4,6]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, 266237, China, yuqy@mail.sdu.edu.cn, guoyanzhang@sdu.edu.cn
[2] Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing, 10084, China, ktjia@tsinghua.edu.cn
[3] Department of Computer Sciences and Technology, Tsinghua University, Beijing, 10084, China
[4] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Qingdao, Shandong, 266237, China
[5] Zhongguancun Laboratory, Beijing, China
[6] Shandong Institute of Blockchain, Jinan, Shandong, China

Abstract. SPEEDY is a family of ultra low latency block ciphers proposed at TCHES 2021 by Leander *et al.*. The standard version, SPEEDY-6-192 offers 128-bit security with high encryption speed in hardware. Differential cryptanalysis proposed in 1990 by Biham and Shamir is one of the most popular methods of cryptanalysis of block ciphers. It is usually the first choice to evaluate the security for designers when designing a new block cipher. The automatic search for various distinguishers based on SAT and MILP models etc. boosts the cryptanalysis of block ciphers. However, the performance of the automatic search is not always satisfactory, especially for searching long differential trails of block ciphers with large state sizes. Hence, we endeavor to accelerate the SAT-based automatic search model for differentials of SPEEDY. In this paper, we give a 3.5-round differential characteristic with the probability of $2^{-104.83}$ and a 4.5-round differential characteristic with the probability of $2^{-150.15}$. Furthermore, by balancing the key recovery and the differential distinguisher, we adjust the distinguisher to speed up filtering wrong pairs with some tricks. Finally we launch a valid 6-round attack for SPEEDY-7-192 with a complexity of $2^{158.06}$. We also propose a 5-round attack utilizing a 3.5-round differential distinguisher with the time complexity of $2^{108.95}$.

Keywords: SPEEDY · Differential cryptanalysis · Automatic search · SAT model

1 Introduction

SPEEDY [9], proposed by Leander *et al.* at TCHES'21, is a family of ultra low latency block ciphers which is designed to be fast in CMOS hardware. The ultra

* Corresponding author

low-latency 6-bit S-box with a two-level NAND gates tree was introduced to provide confusion, and the linear layer with the depth of 3 XOR was applied to provide strong diffusion with branch number 8.

Differential cryptanalysis [3] is one of the most fundamental techniques for cryptanalysis of block ciphers, which was proposed by Biham and Shamir in 1990 to break the Data Encryption Standard (DES). Differential cryptanalysis is essential to evaluate the security of block ciphers. And many generalizations are proposed like truncated differentials [7], impossible differential attack [6,1], the boomerang attack [19] and the rectangle attack [2] etc.

Searching for a good differential characteristic is one of the most important parts to carry out a differential attack. In [11], Matsui proposed a depth-first branch-and-bound searching algorithm to identify the optimal differentials with the maximum probability of block ciphers. The advantage of this algorithm is enhanced by taking in the customized optimization for the specific cipher. In recent years, tools for solving the basic mathematical method have been used to search distinguishers in cryptanalysis. The Boolean satisfiability problem (SAT) is one of the important basic problems on which the automatic search models are based, it is NP-complete.

The efficiency of the automatic search model is one of the important problems we have to face, although some works aimed at improving the efficiency of the automatic search model proposed, it is still a disturbing problem. The runtime of solving the automatic search model mainly depends on the solvers. It has been experimentally shown that minimizing the number of inequalities in a MILP model did not always minimize the runtime [14], as well there are a few works considering the acceleration of the automatic search based on SAT method. The automatic search for bit-oriented block ciphers is more difficult for both methods, because more variables are introduced for each state and the linear layer mixes the variables fastly. It is challenging that building an efficient automatic search model for `SPEEDY` family, on account of 192-bit suggested block size.

Our Contributions. In this paper, we deliberate on the security of `SPEEDY-r-192` with reduced rounds using differential attack. We unveil some new distinguishers, their structural properties, and key recovery attacks on `SPEEDY-r-192` which were not reported before. Table 1 gives a summary of attacks on SPEEDY till date.

Firstly, we proposed an accelerated automatic search model for `SPEEDY-r-192` based on SAT method. Due to a large internal state of 192 bit and the fast diffusion property, it is hard to exhaust all the values of the bit-level state for long rounds. Thus it seems difficult to build an effective automatic search model for `SPEEDY-r-192`. In this paper, we revisit the constraints of the upper bound, which is called the Sequential Encoding Method [15], and reduce the number of auxiliary variables introduced in the clauses by utilizing the properties of the weight of the probability in differentials for `SPEEDY-r-192`. In this way, we build an effective automatic model for searching the differential trails of `SPEEDY-r-192`. To evaluate the probability of the differential distinguisher more precisely, we search for the clustering of differentials with the same input and out-

put differences. We get the longest differential distinguishers for SPEEDY-r-192, and the runtime is practical and much lower than the previous method.

Secondly, We make a balance in the probability of the differential distinguisher and the non-active bits in the plaintext state that can be used to filter the wrong pairs. The balance strategy speeds up filtering the pairs that do not satisfy the differential distinguisher for SPEEDY. Since the differential distinguisher with maximum probability does not necessarily lead to the most effective key recovery attack, the truncated differentials in the extended rounds also impact the complexity of the differential cryptanalysis. This case has been discussed in some rectangle attacks [20,12,5]. Therefore, we adjust the input difference of the distinguisher and add some conditions to control the difference propagation in the extended rounds to make there are some bits with zero difference in the plaintext. The zero difference in the plaintext can filter the wrong pairs in advance in the data collection phase, which greatly reduces the time complexity in key recovery phase.

With these techniques, we launch a 6-round key-recovery attack for SPEEDY-7-192 within the claimed security, which is the longest attack on SPEEDY-r-192 as far as we know. We also proposed a 5-round attack with lower complexity. The results are shown in Table 1.

Table 1: Summary of cryptanalytic results on SPEEDY.

Distinguishers					
Method	Round	Data	Time	Memory	Ref.
Differential and linear	2	2^{39}	2^{39}	-	[9]
	3	2^{69}	2^{69}	-	[9]
Cube	2	2^{14}	2^{14}	-	[13]
Cube	3	2^{13}	2^{13}	-	[13]
Differential	4.5	$2^{150.15}$	$2^{150.15}$	-	Sect. 4.1
Differential	3.5	$2^{104.83}$	$2^{104.83}$	-	Sect. 5.1
Key recovery					
Integral	3	$2^{17.6}$	$2^{52.5}$	$2^{25.2}$	[13]
Differential	5	$2^{108.91}$	$2^{108.95}$	$2^{108.91}$	Sect. 4
Differential	6	$2^{158.04}$	$2^{158.06}$	$2^{158.04}$	Sect. 5

2 Preliminary

2.1 Description of SPEEDY

SPEEDY [9] is a family of ultra-low latency block ciphers designed by Leander *et al.* at TCHES 2021, the designers use SPEEDY-r-6ℓ to differentiate all the variants, where $6l$ denotes the block and key size, and r indicates the number of iterated rounds.

The internal state is viewed as an $\ell \times 6$ binary matrix, and we use $x_{[i,j]}$ to denote the bit located at row i, column j of the state x, where $0 \leq i < \ell$ and $0 \leq j < 6$.

The default block and key size for `SPEEDY` is 192, *i.e.* $\ell = 32$. And this is the only block size we considered in this paper, the designers claimed the security for this instance with iterated rounds 5, 6 and 7. The 5-round version `SPEEDY-5-192` has a security level of 2^{128} time complexity with 2^{64} data complexity as restriction, `SPEEDY-6-192` and `SPEEDY-7-192` can achieve 128-bit and 192-bit security levels, respectively. We pay attention to the differential cryptanalysis of the default version `SPEEDY-r-192`.

We review the details of the round function for encryption of `SPEEDY-r-192`. The round function consists of the following five different operations: **SubSbox(SB)**, **ShiftColumns (SC)**, **MixColumns (MC)**, **AddRoundConstant** (A_{C_i}) and **AddRoundKey** (A_{K_i}). For encryption, the iterated round function except the last is defined as

$$\mathcal{R}_i = A_{C_i} \circ \mathbf{MC} \circ \mathbf{SC} \circ \mathbf{SB} \circ \mathbf{SC} \circ \mathbf{SB} \circ A_{k_i}, \text{ with } 0 \leq i \leq r-2.$$

The round function in the last round is

$$\mathcal{R}_{r-1} = A_{k_r} \circ \mathbf{SB} \circ \mathbf{SC} \circ \mathbf{SB} \circ A_{k_{r-1}}.$$

The last round omit the linear layer and constant addition, and append an extra key addition. Here, we introduce the round operations in the following.

SubSbox(SB): The 6-bit S-box S (seen Table 2) is applied to each row of the state, *i.e.* for $0 \leq i < 32$,

$$(y_{[i,0]}, y_{[i,1]}, y_{[i,2]}, y_{[i,3]}, y_{[i,4]}, y_{[i,5]}) = S(x_{[i,0]}, x_{[i,1]}, x_{[i,2]}, x_{[i,3]}, x_{[i,4]}, x_{[i,5]}).$$

Table 2: The S-box S *in* `SPEEDY`

s_0s_1	$s_2s_3s_4s_5$															
	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	08	00	09	03	38	10	29	13	0c	0d	04	07	30	01	20	23
1.	1a	12	18	32	3e	16	2c	36	1c	1d	14	37	34	05	24	27
2.	02	06	0b	0f	33	17	21	15	0a	1b	0e	1f	31	11	25	35
3.	22	26	2a	2e	3a	1e	28	3c	2b	3b	2f	3f	39	19	2d	3d

ShiftRows(SC): The j-th column of the state is rotated upside by j bits.

$$y_{[i,j]} = x_{[i+j,j]}, \ 0 \leq i < 32, \ 0 \leq j < 6.$$

MixColumns(MC): For `SPEEDY-r-192`, a cyclic binary matrix $M(32 \times 32)$ is multiplied to each column of the state. Use $\boldsymbol{x}_{[j]}$ to denote the input of the j-th

column, and use $\boldsymbol{y}_{[j]}$ to denote the output of the column transform. The column transform $\boldsymbol{y}_{[j]} = M \cdot \boldsymbol{x}_{[j]}$ is

$$\begin{aligned}\boldsymbol{y}_{[j]} &= \boldsymbol{x}_{[j]} \oplus (\boldsymbol{x}_{[j]} \lll 1) \\ &\oplus (\boldsymbol{x}_{[j]} \lll 5) \oplus (\boldsymbol{x}_{[j]} \lll 9) \oplus (\boldsymbol{x}_{[j]} \lll 15) \oplus (\boldsymbol{x}_{[j]} \lll 21) \oplus (\boldsymbol{x}_{[j]} \lll 26),\end{aligned}$$

where $\boldsymbol{x}_{[j]} \lll t$ means the column $\boldsymbol{x}_{[j]}$ rotated upside by t bits, i.e., $x_{[i,j]} = x_{[i+t,j]}$, $\forall\, 0 \le i < 32$.

AddRoundKey(A_{k_r}): The 192-bit round key k_r is XORed to the internal state, as:

$$y_{[i,j]} = x_{[i,j]} \oplus k_{r_{[i,j]}},\ 0 \le i < 32,\ 0 \le j < 6.$$

AddRoundConstant(A_{c_r}): The 192-bit constant c_r is XORed to the whole of the state.

$$y_{[i,j]} = x_{[i,j]} \oplus c_{r_{[i,j]}},\ 0 \le i < 32,\ 0 \le j < 6.$$

Since AddRoundConstant does not alter the validities of attacks in this paper, the constants $c_{r_{[i,j]}}$ are not introduced.

Key Schedule: The algorithm receives a 192-bit master key and initializes it as the subkey k_0. Then a bit permutation PB is used to compute the next round subkey, *i.e.*

$$k_{r+1} = PB(k_r).$$

For more details of SPEEDY, please refer to [9].

2.2 Observations on Differential Properties of SPEEDY

For the **SB** operation with the input difference α and the output difference β, and differential pair (α, β) satisfies the equation 1. We have the following observations according to the Differential Distribution Table.

$$S(x) \oplus S(x \oplus \alpha) = \beta. \tag{1}$$

Observation 1 *For given $\alpha =$ `100000` and $\beta =$ `*****0`$(\beta \neq 0)$, the probability of the propagation $\Pr\{\alpha \xrightarrow{SB} \beta\} = 3/4 \approx 2^{-0.42}$, and the number of β is 15, where $'*'$ means the unknown bit value. Each differential pair (α, β) satisfies the equation 1.*

Observation 2 *For given $\alpha =$ `001000` and $\beta =$ `0*****`$(\beta \neq 0)$, the probability $\Pr\{\alpha \xrightarrow{SB} \beta\} = 15/16 \approx 2^{-0.09}$, and the number of β is 17, where $'*'$ means the unknown bit value. The differential pair (α, β) satisfies equation (1).*

Observation 3 *For given $\alpha =$ `**0***** $(\alpha \neq 0)$ and $\beta =$ `010000`, the probability $\Pr\{\alpha \xrightarrow{SB} \beta\} \approx 2^{-0.54}$, where $'*'$ means the unknown bit value, and the differential pair $(\alpha, \beta)$ satisfies equation (1). Given $\beta =$ `010000`, when $\alpha =$ `0*****` or $\alpha =$ `*****0`, the probability becomes 2^{-1} or $2^{-0.67}$.*

For each column of **MC** operation, we have the following observation:

Observation 4 *Let $\boldsymbol{y}$ be a column of the input of the inverse of $\boldsymbol{MC}$ and the corresponding output be $\boldsymbol{x}$, i.e. $\boldsymbol{y} = M \cdot \boldsymbol{x}$. We simply consider the output form $\boldsymbol{x}$, where $\boldsymbol{y}$ has the form $\boldsymbol{y}_t \neq 0$ $(t = i, j)$ and $\boldsymbol{y}_t = 0$ $(t \notin \{i, j\})$, $\boldsymbol{y}_t$ denotes the t-th bit of $\boldsymbol{y}$.*

- *$j = i + 1$, the Hamming weight $H(\boldsymbol{x})$ is 14, when $i = 0$, $\boldsymbol{x}$ =`0x4CD019F4;`*
- *$j = i + 2$, the Hamming weight $H(\boldsymbol{x})$ is 14, when $i = 0$, $\boldsymbol{x}$ =`0x6AB8150E;`*
- *$j = i + 3$, the Hamming weight $H(\boldsymbol{x})$ is 16, when $i = 0$, $\boldsymbol{x}$ =`0x798C1373;`*
- *$j = i + 4$, the Hamming weight $H(\boldsymbol{x})$ is 12, when $i = 0$, $\boldsymbol{x}$ =`0xF016104D;`*
- *$j = i + 5$, the Hamming weight $H(\boldsymbol{x})$ is 15, when $i = 0$, $\boldsymbol{x}$ =`0xB4DB11D2.`*

2.3 Complexity Analysis of the Differential Attack

Let $\Delta_{in} \to \Delta_{out}$ be a r-round differential characteristic of an algorithm $E(x, k)$, which is a $\mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ mapping, the couple of $(\Delta_{in}, \Delta_{out})$ should satisfy

$$Pr\{E(x, k) \oplus E(x \oplus \Delta_{in}, k) = \Delta_{out}\} > 2^{-n}$$

for $x \in \mathbb{F}_2^n$ and any fixed $k \in \mathbb{F}_2^m$.

The probability is calculated as the sum of probabilities regarding all trails sharing the same input and output differences with the differential [8]. Denote the probability of the r-round differential distinguisher as p_0 and the number of plaintext (or ciphertext) pairs utilized in the attack as N_D. Then under the right key guess, the counter memorizing the number of pairs satisfying the differential distinguisher follows a binomial distribution of parameters (N_D, p_0). On the other side, suppose that the probability of a pair fulfilling the differential under a wrong key guess is p_1. Consequently, the counter follows a binomial distribution of parameters (N_D, p_1). We set a threshold τ_D for the attack, if the counter of the right pairs is no less than τ_D, the key guess will be accepted.

There are two types of errors which are always need to face in the hypothesis test, which are denoted by α, the non-detection error probability, and β, the false alarm error probability. α and β can be got from the formulas in [4].

Then the total time complexity of the differential cryptanalysis T can be departed into three parts, denoted by $T = T_1 + T_2 + T_3$. T_1 is the number of encryptions to prepare the necessary plaintext and ciphertext pairs which lead to the right pairs passing the distinguisher. We can estimate T_1 by N times of encryption, where N is the number of plaintexts (ciphertexts) we chose, which corresponds to the data complexity.

Time complexity T_2 denotes the average complexity needed to decide whether a pair satisfy the distinguisher under our key guess. For the N_D pairs we utilized in the attack, use T_E to denote the time for one encryption, if we need time T_F to determine whether a pair satisfy the distinguisher or not on average. Then the time complexity can be estimated by

$$T_2 = \frac{T_F}{T_E} \cdot N_D.$$

After the key recovery phase, there will be $2^m \cdot \beta$ keys remaining in the theory. Therefore, we expected

$$T_3 = 2^m \cdot \beta \cdot (1 - 2^{-n})$$

encryptions to recover the entire master key. And the success probability of the attack is equal to $1 - \alpha$.

2.4 Automatic Searching Model Based on SAT Problem

The Boolean Satisfiability (SAT) problem studies the satisfiability of a given Boolean formula, it is said satisfiable if there exists an assignment of Boolean values to variables so that the formula is evaluated to be True.

Conjunctive Normal Form (CNF) is a generic representation of SAT problem. The formula is expressed as conjunction ($\wedge$) of one or more clauses, where a clause is a disjunction ($\vee$) of many Boolean variables (possibly negated). The CNF encodings for basic operations in cryptographic primitives are introduced. In this section, we use $\alpha_i (0 \leq i < n)$ to denote the i-th element of the n-bit vector α, α_0 stands for the most significant bit.

-**Building constraints for non-probabilistic models.** For the linear operations in cryptographic primitives, we can also build the clauses of the SAT model by the same method of building clauses for S-boxes without introducing auxiliary variables, in this section, we just list the clauses for some basic operations.

Clauses for XOR operation. For a n-bit XOR operation with two input differences α and β, and the output difference is denoted by γ. The differential $\alpha \oplus \beta = \gamma$ holds if and only if the values of α, β and γ validate all the assertions in the following.

$$\left.\begin{array}{l} \overline{\alpha_i} \vee \overline{\beta_i} \vee \overline{\gamma_i} = 1 \\ \overline{\alpha_i} \vee \beta_i \vee \gamma_i = 1 \\ \alpha_i \vee \overline{\beta_i} \vee \overline{\gamma_i} = 1 \\ \alpha_i \vee \beta_i \vee \overline{\gamma_i} = 1 \end{array}\right\} 0 \leq i \leq n-1$$

Clauses for COPY operation. For the n-bit COPY operation with input difference α and output difference β. The differential $\beta = \alpha$ holds if and only if the values of α and β validate all the assertions in the following.

$$\left.\begin{array}{l} \alpha_i \vee \overline{\beta_i} = 1 \\ \overline{\alpha_i} \vee \beta_i = 1 \end{array}\right\} 0 \leq i \leq n-1$$

For differentials, the clauses of COPY operation $\alpha = \beta$ can be also applied to shifting operations.

-**Building constraints for S-box.** The propagations of differences and linear masks for S-box operations are probabilistic. Use an s-bit S-box for example, according to the method in [17], let $(I_0, I_1, \ldots, I_{s-1})$ denote the variables which indicate the input difference, and $(O_0, O_1, \ldots, O_{s-1})$ denote the output difference, introduce several variables $\rho_0, \rho_1, \ldots, \rho_{h-1}$ to denote the weight of the

opposite number of the binary logarithm of the probability. Because the SAT problem is oriented to binary variables, the number of auxiliary variables depends on the weight of the probability. With these variables, we can define a $(2s+h)$-bit Boolean function $f(\boldsymbol{z})$, where $\boldsymbol{z} = (I_0, I_1, \ldots, I_{s-1}, O_0, \ldots, O_{s-1}, \rho_0, \ldots, \rho_{h-1})$, if $(I_0, \ldots, I_{s-1}) \rightarrow (O_0, \ldots, O_{s-1})$ is a possible propagation with the probability weight $w_0 \cdot \rho_0 + w_1 \cdot \rho_1 + \cdots + w_{h-1} \cdot \rho_{h-1}$, then $f(\boldsymbol{z}) = 1$, else $f(\boldsymbol{z}) = 0$. Then we can get a set of Boolean equations by reformulating the $f(\boldsymbol{z})$ as the product-of-sum representation

$$f(\boldsymbol{z}) = \bigwedge_{\boldsymbol{c} \in \mathbb{F}_2^{2s+h}} \Big(f(\boldsymbol{c}) \vee \bigvee_{i=0}^{2s+h-1} (\boldsymbol{z}_i \oplus \boldsymbol{c}_i) \Big),$$

where $\boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2s+h-1})$, after getting the Boolean equations, we can simplify the expression utilizing some openly available programs like Logical Friday [7], and yield a smaller set of clauses.

-Sequential encoding method for constraining the upper bound. When we aim at r-round differential trails, denote the auxiliary variables stand for the probability for the j-th S-box in the i-th round as $\rho_l^{(i,j)}$, where $0 \leq i \leq r-1,\ 0 \leq j \leq n-1$ and $0 \leq l \leq h-1$. The weight equals to the opposite number of the binary logarithm of the probability of the differential trail should be $\sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \sum_{l=0}^{h-1} w_l \cdot \rho_l^{(i,j)}$. In theory, if we want to constrain the solution range with the prospective value ω as the weight of the trail, our model should add the additional constraint

$$\sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \sum_{l=0}^{h-1} w_l \cdot \rho_l^{(i,j)} \leq \omega.$$

However, all the variables in the SAT are binary, it is unfeasible to handle the decimal and the integer part at the same time. So we convert the bound into several parts with different decimal weights and handle the part with different weights separately. For example, let the $\rho_{h-1}^{(i,j)}$ denote the part with decimal weight for each S-box, and the other variables denote the part with integer weight. Then the constraints for the upper bound can be rewritten as $\sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \sum_{l=0}^{h-2} \rho_l^{(i,j)} + w_{h-1} \cdot \sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \rho_{h-1}^{(i,j)}$. The objective function of the SAT problem consists of the following two inequalities.

$$\sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \sum_{l=0}^{h-2} \rho_l^{(i,j)} \leq \omega_I,\ \sum_{i=0}^{r-1} \sum_{j=0}^{n-1} \rho_{h-1}^{(i,j)} \leq \omega_D \tag{2}$$

where ω_I and ω_D are two non-negative integers, and $\omega = \omega_I + w_{h-1} \cdot \omega_D$.

These two restrictions in 2 meet the form $\sum_{i=0}^{n-1} u_i \leq k$, where k is a non-negative integer. If $k = 0$, this constraint is equivalent to the following n Boolean expressions:

$$\bar{u_i} = 1, 0 \leq i \leq n-1.$$

[7] https://web.archive.org/web/20131022021257/http://www.sontrak.com/

Else if $k > 0$, according to the method in [10], which is called **sequential encoding method**. we introduce $(n-1) \cdot k$ auxiliary Boolean variables $v_{i,j} (0 \leq i \leq n-2, 0 \leq j \leq k-1)$, and use the following clauses to build the constraints for $\sum_{i=0}^{n-1} u_i \leq k$:

$$\left.\begin{array}{l} \overline{u_0} \vee v_{0,0} = 1 \\ \overline{v_{0,j}} = 1, 1 \leq j \leq k-1 \\ \overline{u_i} \vee v_{i,0} = 1 \\ \overline{v_{i,0}} \vee v_{i,0} = 1 \\ \overline{u_i} \vee \overline{v_{i-1,j-1}} \vee v_{i,j} = 1 \\ \overline{v_{i-1,j}} \vee v_{i,j} = 1 \\ \overline{u_i} \vee \overline{v_{i-1,k-1}} = 1 \\ \overline{u_{n-1}} \vee \overline{v_{n-2,k-1}} = 1 \end{array}\right\} 1 \leq j \leq k-1 \Bigg\} 1 \leq i \leq n-2$$

Using the model shown above, we build the constraints of the SAT problem for searching differential characteristics, and we utilize CryptoMinisat5 [16] as the solver with parameters set as shown in Sect. 3.

3 Searching for Good Differential Trails for SPEEDY

It requires searching a space of exponential size in the number of Boolean variables to solve the SAT problem. We believe that the size of the problem needed to be solved is one of the most important factors affecting the runtime of the SAT based automatic search model. In this section, we try to build the automatic search model for the differential trails of SPEEDY-r-192 with as few variables as possible based on the SAT model and discuss how to solve the model with as few as possible running times.

3.1 Improved Automatic Searching Model for SPEEDY

For **SubSbox** operation of SPEEDY-r-192, the entries in the DDT of S-box has six possible evaluations, which are 0, 2, 4, 6, 8, and 16, with corresponding differential probabilities in the set $\{0, 2^{-5}, 2^{-4}, 2^{-3.415}, 2^{-3}, 1\}$. When we use the automatic search model proposed in [17,18], six auxiliary Boolean variables are required for each S-box, and $O((n-1) \cdot k)$ auxiliary Boolean variables are also needed according to the sequential encoding method in Sect. 2.4 to build the constraints for the upper bound of the probability for the distinguishers, where n is the number of variables which denote the probability for each S-box and k is the upper bound for the probability of the whole distinguisher. In order to descend the scale of the auxiliary variables, we introduce four Boolean variables $\rho_0, \rho_1, \rho_2, \rho_3$, let p denote the probability of the possible differential propagation,

then we build the constraints for the variables as follows:

$$\rho_0||\rho_1||\rho_2||\rho_3 = \begin{cases} 1110, & if\ p = 2^{-5} \\ 0110, & if\ p = 2^{-4} \\ 0011, & if\ p = 2^{-3.415} \\ 0010, & if\ p = 2^{-3} \\ 0000, & if\ p = 1 \end{cases}$$

In order to build the constraints for the upper bound of the probability of the whole distinguisher with as few auxiliary variables as possible, we depart the objective function of the SAT problem into three parts, which are:

$$\sum_{i=0}^{r-1}\sum_{j=0}^{31}\sum_{l=0}^{2}\rho_l^{(i,j)} \le \omega_I,\ \sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_2^{(i,j)} \le \omega_S \text{ and } \sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_3^{(i,j)} \le \omega_D.$$

Where ω_I, ω_S and ω_D are non-negative integers, and $0 \le i \le r-1, 0 \le j \le 31$. The prospective value for the weight of the trail ω can be represented by $\omega = \sum_{i=0}^{r-1}\sum_{j=0}^{31}\sum_{k=0}^{2}\rho_l^{(i,j)} + 2\cdot\sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_2^{(i,j)} + 0.415\cdot\sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_3^{(i,j)}$. It is obvious that the number of S-boxes in the trail can be represented as $\sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_2^{(i,j)}$, so we can follow the steps in Sect. 3.2 to solve the model.

The constraints for **ShiftRows**, **MixColumns** and **AddRoundKey** have nothing to do with the probability of the trail, so we do not need to make addtional constraints on these operations.

3.2 Process of Solving the Model

$p_1 = \sum_{i=0}^{r-1}\sum_{j=0}^{31}\sum_{k=0}^{2}\rho_k^{(i,j)}$, $p_2 = \sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_2^{(i,j)}$ and $p_3 = \sum_{i=0}^{r-1}\sum_{j=0}^{31}\rho_3^{(i,j)}$ denote the summation of partial weights respectively. Suppose the optimal trail we found has the prospective value for the weight of the probability $\omega = a + 2\cdot b + 0.415\cdot c$, *i.e.* $p_1 = a$, $p_2 = b$ and $p_3 = c$, if the number of active S-boxes is not less than this trail, the trails with higher probability must satisfy the conditions of the parameters as shown below, the trivial cases $p_1 \le a$ and $p_3 \le c$ are ruled out.

Table 3: Possible value combinations of p_1, p_2 and p_3

	p_1	p_2	p_3
1	$a+n$	b	$c - \lceil \frac{n}{0.415} \rceil$
2	$a-n$	b	$c + \lfloor \frac{n}{0.415} \rfloor$
3	$\le a-k+n$	$b+1$	$\le c - \lceil \frac{n-k+2}{0.415} \rceil$
4	$\le a-k-n$	$b+1$	$\le c + \lfloor \frac{n+k-2}{0.415} \rfloor$

The case where the number of active S-boxes is greater than $b+1$ can be dealt with inductively. And we notice that although there are many possible scenarios theoretically, we need not test all of them, because the parameter p_1 usually increases with the number of active S-boxes. So we proposed a heuristic method to search for the differential trail with optimal probability. Firstly we search for the minimized number of active S-boxes, *i.e.* we set the objective function to minimize the parameter p_2, suppose the obtained minimum is b. Secondly, we run the solver again with the objective function to minimize the parameter p_1 with the constraint $p_2 = b$ and suppose the minimize objective function is a, then with the constraints $p_1 = a$ and $p_2 = b$, we set the objective function to minimize the parameter p_3, and denote the value is c. Finally, we test the possible value combinations of p_1, p_2 and p_3 in Table 3 to ensure the probability of the trail we found is optimal, if it is not, repeat the test.

The minimum of the parameter p_1, p_2 and p_3 have already constrained the candidate of the test, so we just need to repeat the test few times to ensure the trail is optimal. The size of the auxiliary variables we introduced is $O(r \cdot (3\omega_I + \omega_S + \omega_D))$, which is several times less than the size of the problem that we build constraints with 6 auxiliary variables for each S-box. The improvement of the runtime is significant, the average time of solving our model to search for the 4.5-round distinguisher once is about 3 hours, as well the time for solving the model normally once to search for the 4.5-round distinguisher is over 24 hours.

4 Differential Cryptanalysis on 6-round SPEEDY

In this section, we give differential cryptanalysis of SPEEDY-7-192 to achieve the rounds as long as possible. According to the round function of SPEEDY, we first select the differential distinguisher with $N + 0.5$ rounds which are suitable for the key-recovering phase with the optimal probability and mount a $1 + N + 1$ key-recovery attack under chosen-ciphertext ability. In this section, we show that we can achieve a 6-round attack for SPEEDY-7-192 with the time complexity of $2^{158.06}$ and data complexity of $2^{158.04}$.

4.1 The 4.5-round Differential Distinguisher

Because of the rapid propagation of the truncated differential trails of SPEEDY, it will cost lots of time complexity on filtering out the wrong pairs which do not conform with the differential trail in the key recovery phase. However, there is just one **ShiftColumns(SC)** operation in the last round of round-reduced SPEEDY-7-192, the truncated propagation of the second **SubBox(SB)** operation can be easily handled because the 6-bit non-zero difference of each S-box only leads to 6 active bits in the state of ciphertexts. So we search for an optimal 4.5-round differential trail as the distinguisher and launch a 6-round key recovery attack on SPEEDY-7-192 under chosen-ciphertext ability.

According to the method in Sect. 3.2, firstly we find out that the minimum number of active S-boxes of 4.5-round differential trails for SPEEDY is 43, and then

we search for the optimal differential distinguishers with 43 active S-boxes and get the maximum probability of 4.5-round differential path is $2^{-150.15}$. Finally we resolve the automatic model several times with the constraints for adjusted parameters to ensure that there are no trails with 44 or 45 active S-boxes have probability higher than $2^{-150.15}$. The 4.5-round differential path we got from the SAT solver is shown in Figure 1.

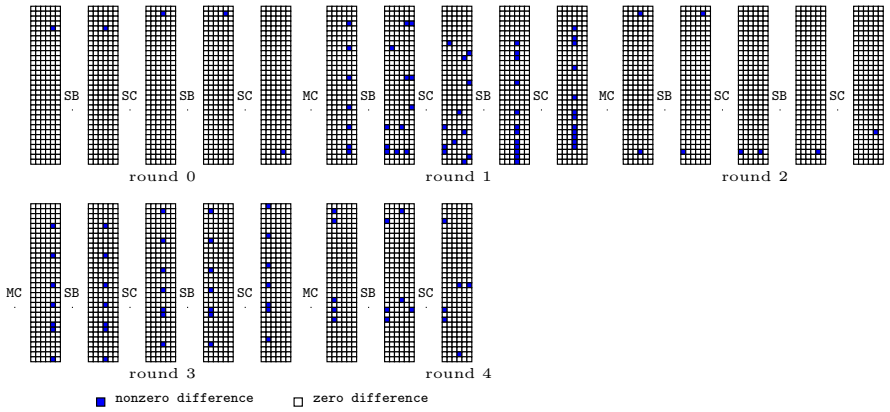

Fig. 1: The 4.5-round differential distinguisher for `SPEEDY-r-192`

4.2 Speed up Filtering Wrong Pairs by Optimizing the Distinguisher

We launch a 6-round key recovery attack based on the 4.5-round differential path by extending 1 round at the beginning and 0.5 round at the end.

The chosen-ciphertext attack with data structure is applied to reduce the time complexity. We choose 2^s structures of size 2^t (t denotes the number of active bits in the differences of ciphertext). There are about 2^{2t-1} pairs for each structure. Let p be the probability of the differential distinguisher we found. We choose enough ciphertexts such that there are about $2^{s+2t-1-t} \times p \geq 1$ pairs satisfying the output of the differential distinguisher. Hence, the data complexity is 2^{s+t}. We need to guess the subkeys for the 2^{s+2t-1} pairs, which plays a dominant role in the time complexity of the key recovery phase. Therefore, we adjust the input difference of the distinguisher and add some conditions to control the difference propagation in the extend round to make some bits in the plaintext with zero difference. The zero difference in the plaintext can remove some pairs not satisfying the truncated differential in the data collection phase, which reduces the time complexity of the key recovery phase.

Here, we describe the methods to make the differences of some bits of plaintext pairs become zero. The differential distinguisher with the maximum probability does not necessarily lead to the most effective key recovery attack, the

truncated differentials in the extended rounds also impact the complexity of the differential cryptanalysis. For partial decryption, the operation $\mathbf{MC}^{-1}$ can spread one active S-box at the beginning of the distinguisher to 19 active S-boxes, which can diffuse to the whole plaintext state (all the 32 S-boxes are active) after partial decryption through $\mathbf{SC}^{-1} \circ \mathbf{SB}^{-1}$. But according to the Observation 4, two active S-boxes at the beginning of the distinguisher can lead to less active S-boxes after propagation through $\mathbf{MC}^{-1}$, which make there are some bits with zero-difference can be used to filter wrong pairs in the data collection phase. Hence we proposed the trade-off strategy to balance the time complexity.

As shown in Figure 3, we adjust the differential propagation in round 1, such that the two active S-boxes at the beginning of the differential distinguisher can lead to 12 active S-boxes after propagation through $\mathbf{MC}^{-1}$. The probability of the altered differential characteristic is $2^{-155.735}$. And the probability of the differential distinguisher is $2^{-155.2}$, which is recalculated by multi differential trials. The altered distinguisher can generate 2 rows with zero difference in the state of plaintext, which are row 8 and row 9.

Meanwhile, considering the differences in rows 19 to 26 of the plaintexts, although these rows may be active after propagating through $\mathbf{SC}^{-1} \circ \mathbf{SB}^{-1}$, the actual number of active S-boxes of the first **SB** operation the rows 19 to 26 of round 0 depends on the output difference of the rows 18 and 24 of the second **SB** layer in round 0. So we exhaust all the possible differences of these active S-boxes propagating backward through the $\mathbf{SB}^{-1} \circ \mathbf{SC}^{-1} \circ \mathbf{SB}^{-1}$ operation, and find that the probability of the situations that the differences in rows 21 to 23 of the plaintext state are all zero is $2^{-1.415}$, which can be viewed as a part of the truncated differential.

Up to now, we get the altered distinguisher with zero difference in rows 8, 9, 21, 22 and 23 in plaintexts and zero difference in position 3 of the 6-th column in ciphertexts. The 4.5-round differential can generate plaintexts by partial decryption with zero differences in rows 9 to 10 and 21 to 23 of the plaintext state with the probability $2^{-155.2-1.415-0.42} = 2^{-157.04}$. And this distinguisher leads to 30-bit filter in the data collection phase.

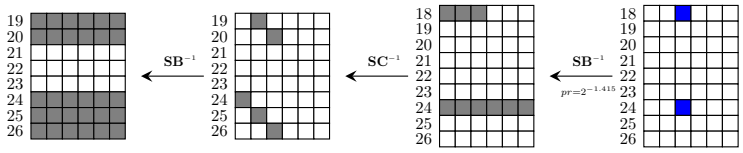

Fig. 2: Improved pre-filtering phase for the head of the distinguisher.

According to the truncated differential structure, we choose 2^s structures, each including 2^{29} ciphertexts by traversing the active bits with fixed random values for non-active bits and query the corresponding plaintexts. Let the bits

with the zero-difference of the plaintext as the index to obtain the pairs. There are about $2^{s+29\times 2-1-30} = 2^{s+27}$ pairs remaining.

4.3 Key Recovery of 6-round SPEEDY-192

Some precomputation tables are used to reduce the time complexity in the key recovery phase. Use the notations with the meaning in equation (1), we build a hash table H indexed by (α, β) to store the values $(x, SB(x))$. For given $\alpha = 4, 5, \texttt{0x20}$, there are about 21, 27 and 23 values of β, respectively. For each active row of the ciphertext pairs, we compute the output difference β of each pair, and removing the pairing which can not generate the given input difference α. There are about $2^{s+27} \times 2^{22.1-29} = 2^{s+20.1}$ pairs remaining. Then look up the table H to get the value $S(x)$ by the index, and deduce the key bits involved in this row. So in the last key addition, we deduce the key bits involved in rows 3, 16, 21, 23 and 30 of the state of ciphertexts by looking up tables. And we get $2^{s+20.1}$ pairs each corresponding to $2^{7.48}$ 30-bit keys.

As the key schedule used in SPEEDY family is linear, use the key bit in the set 0 to 191 of the zero-th round key k_0 to denote the obtained 30-bit key in the last round, seen in Table 4.

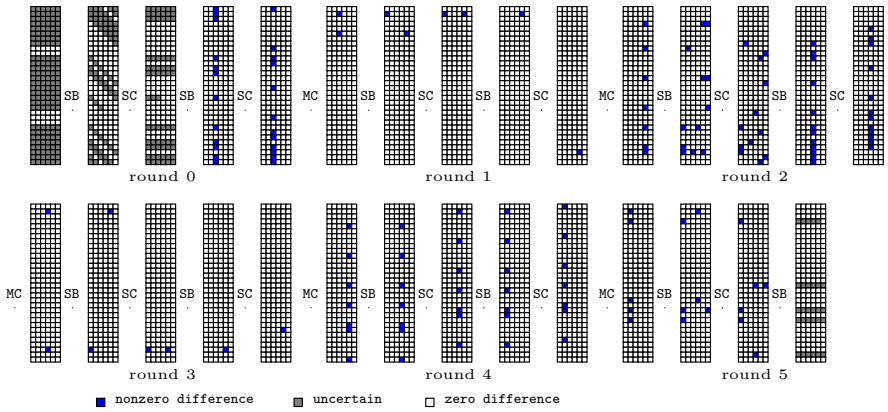

Fig. 3: 6-round attack for SPEEDY-7-192.

Then we deduce the key bits involved in row 18 after $\mathbf{SC} \circ \mathbf{SB}$ operation of round 0. We first deduce the key bits involved in row 20 in the plaintext state, because the key bit of position 120 is known, there are about $2^{s+20.1}$ pairs each corresponding to $2^{7.48}$ 35-bit keys after looking up the hash table H to deduce the 5 key bits left. Then compute the key bits involved in row 19 by looking up the hash table, we get $2^{s+20.1}$ pairs, which correspond to $2^{7.48}$ 40-bit keys. Deduce the key bits involved in the 18-th row by table lookups to obtain $2^{s+20.1}$

Table 4: The deduced key bits involved in the last key addition

row	key guess					
3	138	91	44	189	142	95
16	120	73	26	171	124	77
21	54	7	152	105	58	11
23	66	19	164	117	70	23
30	12	157	110	63	16	161

pairs, which are corresponding to $2^{8.48}$ 45-bit keys. Then we guess the unknown 3 bits values in row 18 of the state after the first **SC** and check the known output difference. Up to now we obtain $2^{s+20.1}$ pairs, which are corresponding to $2^{5.48}$ 46-bit key.

For the other rows, we just need to calculate the key bits that are not involved in the positions that we have obtained, the time complexity is much lower than computing the key bits involved in the first three rows as shown above. For example, we only need to guess the key bits involved in rows 13 to 17 when filtering with row 13 after the second **SB** layer. The time complexity of each looking up hash table is approximate to the looking up S-box. The time complexity is about $(2^{s+27} + 2^{s+20.1} \times (8 + 2^6)) \times 1/32 \times 1/12 \approx 2^{s+19}$.

Each guess determines a 168-bit key, and we exhaust the remaining key bits. By the complexity cryptanalysis in Sec. 2.3, we set $s = 129.04$. Under the right key guess, $2^{s+2\times29-1-29} \times 2^{-157.04} = 1$ pair is expected in content with the 4.5-round differential. About $2^{s+27-29-162} = 2^{-35.38}$ pairs will validate the input and output differences of the 4.5-round distinguisher under a wrong 168-bit key guess. According to the formulas, we have $\alpha < 0.2$ and $\beta < 2^{-40}$, hence the success probability is $P_S = 1 - \alpha > 80\%$ and the total time complexity of the 6-round attack is given by

$$2^{129.04+29} + 2^{129.04+19} + 2^{192} \cdot 2^{-40} \cdot (1 - 2^{-192}) = 2^{158.06}.$$

The data complexity of the 6-round attack is $2^{s+29} = 2^{158.04}$.

5 Differential Cryptanalysis of 5-round SPEEDY

5.1 Speed up Filtering Wrong Pairs with a 3.5-round Differential Distinguisher

In this section, we give an improved 3.5-round differential for SPEEDY-r-192 and mount a 5-round differential attack.

The searching method of a 3.5-round differential is the same as that used for searching the 4.5-round differential distinguisher of SPEEDY-r-192. Firstly we search for the differential characteristic with the minimum number of active S-boxes. Then we alter the constraints for other parameters and find the optimal differential trails with the maximum probability. We find out that the minimum number of active S-boxes of the 3.5-round is 31, and the optimal 3.5-round

differential trail with the probability of $2^{-104.83}$ got from the solver is shown in Figure 4.

The differential $\texttt{000010} \xrightarrow{SB} \texttt{000100}$ and $\texttt{000010} \xrightarrow{SB} \texttt{001000}$ for S-boxes of SPEEDY both has probability 2^{-3}. So we reduce the number of active S-boxes in the last **AK** operation from 8 to 5 without changing the probability. And following the idea in Sect 4.2, we just alter the differential propagation in the first round to get a distinguisher with two non-active S-boxes in rows 9 and 10 of the plaintexts. The detail of the altered differential distinguisher is shown in Figure 5, and the probability of the trail is $2^{-106.66}$. After searching for all the differential characteristics with the same input and output difference as well as no more than 35 active S-boxes, the probability of the differential trail is adjusted to $2^{-105.7}$.

In order to increase the number of zero-difference bits in the plaintexts which are used to remove more wrong pairs in the data collection, we made a few adjustments to the above distinguisher. Because the active bits in rows 8, 11 and 27 after the first **SB** layer originate from three different S-boxes in the second **SB** layer. According to Observation 3, the differential probability $\Pr\{\texttt{**0***} \xrightarrow{SB} \texttt{010000}\} = 2^{-0.54}$, $\Pr\{\texttt{0*****} \xrightarrow{SB} \texttt{010000}\} = 2^{-1}$, and $\Pr\{\texttt{*****0} \rightarrow \texttt{010000}\} = 2^{-0.67}$. And these three S-boxes are disjoint in the truncated differential. Such that the probability of the truncated differential that the difference in rows 8, 11 and 27 of plaintext state are all zero is $2^{-0.54-1-0.67} = 2^{-2.21}$.

The 3.5-round differential can generate plaintexts by partial decryption with zero differences in rows 8 to 11 and 27 of the plaintexts, having the probability $2^{-105.7-2.21} = 2^{-107.91}$.

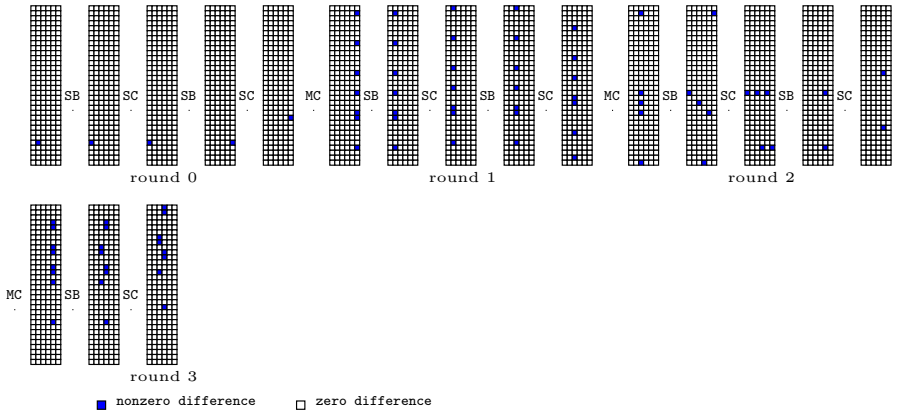

Fig. 4: The 3.5-round differential distinguisher for SPEEDY-r-192.

5.2 Key Recovery of 5-round SPEEDY-r-192

We launch a 5-round differential attack by extending 1 round at the beginning of the 3.5-round differential and appending 0.5 round. According to Observation 2, we choose the ciphertexts with output differences in the form `0*****` instead of `******` at rows 13 and 21, which can generate the input difference 001000 effectively. There are 28 active bits seen in Figure 5.

Use the same method in Sect. 4.3, we build a hash table indexed by input and output differences (α, β) to store the values $(x, SB(x))$ for S-box. For given $\alpha = 8, 12$, there are about 17, 27 values of β, respectively.

We choose 2^s structures, each including 2^{28} ciphertexts by traversing the active bits with fixed random values for non-active bits and query the corresponding plaintexts. Let the bits with the zero-difference of the plaintext as the index obtain the pairs. There are about $2^{s+28\times 2-1-30} = 2^{s+25}$ pairs remaining. For each active row of the ciphertext pairs, we compute the output difference β of each pair, and remove the pairing which can not generate the given input difference α. There are about $2^{s+25} \times 2^{22.44-28} = 2^{s+19.44}$ pairs remaining. Then look up the table H to get the value $S(x)$ by the index, and deduce the key bits involved in these rows. Hence, we deduce the key bits of k_6 involved in rows 1, 6, 10, 13 and 21. There are about $2^{s+19.44}$ pairs each corresponding to $2^{5.56}$ 30-bit keys.

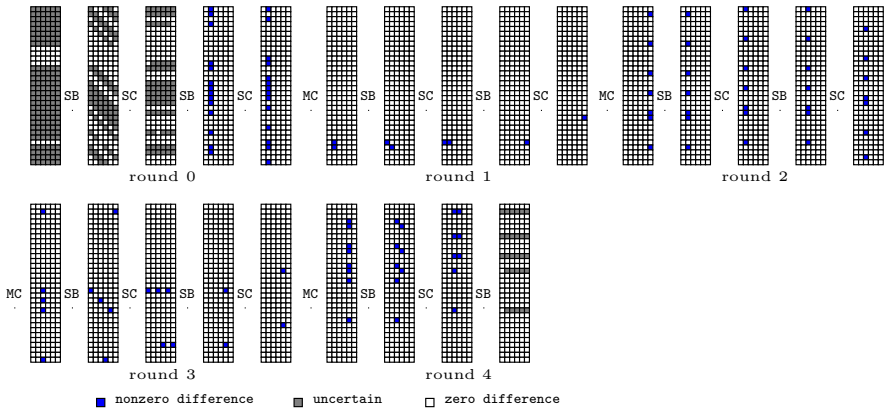

Fig. 5: 5-round attack for SPEEDY-r-192.

For the first key addition, we first deduce the key bits involved in row 25 of the state after the **SC** $\circ$ **SB** in round 0. According to the linear key schedule of SPEEDY-r-192, the key bits in positions 155, 163, 169, 173, 183, 185 and 177 have been guessed. First, we deduce the key bits involved in row 28 of the plaintext state, because the key bits in positions 169 and 173 are known, there are about $2^{s+19.44}$ pairs each corresponding to $2^{5.56}$ 34-bit keys after looking up

the hash table to deduce the 4 key bits left. Then compute the key bits involved in row 30 by looking up the hash table, because the key bits in positions 183 and 183 are known, we get $2^{s+19.44}$ pairs each corresponding to $2^{6.56}$ 38-bit keys. Deduce the key bits involved in row 25 and get $2^{s+19.44}$ pairs with $2^{7.56}$ 43-bit keys remaining with the known key bit 155, and deduce the key bits involved in row 29 and get $2^{19.44}$ pairs with $2^{9.56}$ 48-bit keys with the known key bits 177. Finally, we guess the key bits involved in row 26, get $2^{19.44}$ pairs with $2^{11.56}$ 54-bit keys remaining, and guess the unknown key bits in row 25 of the state after the first **SC**, check the output difference to get a 6-bit filter. Up to now, we obtain $2^{s+19.44}$ pairs each corresponding to $2^{6.56}$ 55-bit keys.

For the other rows, we just need to compute the unknown key bits involved in the row, the complexity is much lower than the process we stated above. The time complexity of guessing the key bits involved in the first key addition is about $(2^{s+25} + 2^{19.44} \times (11 + 2^6)) \times 1/32 \times 1/10 \approx 2^{s+18}$.

Each guess determines a 166-bit key, and we exhaust the remaining key bits. In order to get one right pair under the right key guess, we expect $2^{s+2\times 28-1-28} \times 2^{-107.91} \geq 1$, and set $s = 80.91$. For the wrong key guess, about $2^{108.91-28-162} = 2^{-81.09}$ pairs will validate the input and output differences of the 3.5-round distinguisher. According to the formulas in [4], $\alpha < 0.15$ and $\beta < 2^{-100}$, hence the success probability of the attack is $P_S > 85\%$ and the total time complexity of the 5-round attack is given by

$$2^{80.91+28} + 2^{80.91+18} + 2^{192} \cdot 2^{-100} \cdot (1 - 2^{-192}) \approx 2^{108.95}.$$

The data used in the attack is about $2^{80.91+28} = 2^{108.91}$.

6 Conclusion

In this paper, an accelerated automatic search model for `SPEEDY-r-192` based on SAT method is proposed, the automatic search model is practical to give the optimal probability of the differential trail for `SPEEDY`. A 4.5-round differential characteristic with the probability of $2^{-150.15}$ and a 3.5-round differential characteristic with the probability of $2^{-104.83}$ are found by the solver. Furthermore, we propose a 5-round and a 6-round key-recovery attack for `SPEEDY-r-192` utilizing the modified differential distinguisher. These are the attacks that covered the longest rounds for `SPEEDY-r-192` in our knowledge. Our 6-round attack, with $2^{158.06}$ time complexity and $2^{158.04}$ data complexity, can be viewed as a valid attack under the security claim for the round-reduced version of `SPEEDY-7-192`, which covers 6/7 rounds of the block cipher. And our 5-round attack with $2^{108.91}$ data complexity and $2^{108.95}$ time complexity can be viewed as a valid attack for the round-reduced version of `SPEEDY-6-192`.

Acknowledgements We would like to thank the anonymous reviewers for their valuable comments to improve the quality of this paper. This paper is supported by the National Key Research and Development Program of China

(Grant Nos. 2018YFA0704701), the National Natural Science Foundation of China (Grant Nos. 62072270), Shandong Province Key Research and Development Project(Grant Nos. 2020ZLYS09 and 2019JZZY010133).

References

1. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 12–23. Springer (1999), https://doi.org/10.1007/3-540-48910-X_2
2. Biham, E., Dunkelman, O., Keller, N.: The rectangle attack - rectangling the serpent. In: Pfitzmann, B. (ed.) Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding. Lecture Notes in Computer Science, vol. 2045, pp. 340–357. Springer (2001), https://doi.org/10.1007/3-540-44987-6_21
3. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1990), https://doi.org/10.1007/3-540-38424-3_1
4. Blondeau, C., Gérard, B., Tillich, J.: Accurate estimates of the data complexity and success probability for various cryptanalyses. Des. Codes Cryptogr. **59**(1-3), 3–34 (2011), https://doi.org/10.1007/s10623-010-9452-2
5. Dong, X., Qin, L., Sun, S., Wang, X.: Key guessing strategies for linear key-schedule algorithms in rectangle attacks. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 3–33. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_1
6. Knudsen, L.: Deal - a 128-bit block cipher. In: NIST AES Proposal (1998)
7. Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (ed.) Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings. Lecture Notes in Computer Science, vol. 1008, pp. 196–211. Springer (1994), https://doi.org/10.1007/3-540-60590-8_16
8. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings. Lecture Notes in Computer Science, vol. 547, pp. 17–38. Springer (1991), https://doi.org/10.1007/3-540-46416-6_2
9. Leander, G., Moos, T., Moradi, A., Rasoolzadeh, S.: The SPEEDY family of block ciphers engineering an ultra low-latency cipher from gate level for secure processor architectures. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(4), 510–545 (2021), https://doi.org/10.46586/tches.v2021.i4.510-545
10. Liu, Y., Wang, Q., Rijmen, V.: Automatic search of linear trails in ARX with applications to SPECK and chaskey. In: Manulis, M., Sadeghi, A., Schneider,

S.A. (eds.) Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9696, pp. 485–499. Springer (2016), https://doi.org/10.1007/978-3-319-39555-5_26
11. Matsui, M.: On correlation between the order of s-boxes and the strength of DES. In: Santis, A.D. (ed.) Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings. Lecture Notes in Computer Science, vol. 950, pp. 366–375. Springer (1994), https://doi.org/10.1007/BFb0053451
12. Qin, L., Dong, X., Wang, X., Jia, K., Liu, Y.: Automated search oriented to key recovery on ciphers with linear key schedule applications to boomerangs in SKINNY and forkskinny. IACR Trans. Symmetric Cryptol. **2021**(2), 249–291 (2021), https://doi.org/10.46586/tosc.v2021.i2.249-291
13. Rohit, R., Sarkar, S.: Cryptanalysis of reduced round SPEEDY. IACR Cryptol. ePrint Arch. p. 612 (2022), https://eprint.iacr.org/2022/612
14. Sasaki, Y., Todo, Y.: New algorithm for modeling s-box in MILP based differential and division trail search. In: Farshim, P., Simion, E. (eds.) Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10543, pp. 150–165. Springer (2017), https://doi.org/10.1007/978-3-319-69284-5_11
15. Sinz, C.: Towards an optimal CNF encoding of boolean cardinality constraints. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3709, pp. 827–831. Springer (2005), https://doi.org/10.1007/11564751_73
16. Soos, M., Nohl, K., Castelluccia, C.: Extending sat solvers to cryptographic problems. In: Kullmann, O. (ed.) Theory and Applications of Satisfiability Testing - SAT 2009. pp. 244–257. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
17. Sun, L., Wang, W., Wang, M.: Accelerating the search of differential and linear characteristics with the SAT method. IACR Trans. Symmetric Cryptol. **2021**(1), 269–315 (2021), https://doi.org/10.46586/tosc.v2021.i1.269-315
18. Sun, L., Wang, W., Wang, M.: Improved attacks on GIFT-64. In: AlTawy, R., Hülsing, A. (eds.) Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13203, pp. 246–265. Springer (2021), https://doi.org/10.1007/978-3-030-99277-4_12
19. Wagner, D.A.: The boomerang attack. In: Knudsen, L.R. (ed.) Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1636, pp. 156–170. Springer (1999), https://doi.org/10.1007/3-540-48519-8_12
20. Zhao, B., Dong, X., Jia, K.: New related-tweakey boomerang and rectangle attacks on deoxys-bc including BDT effect. IACR Trans. Symmetric Cryptol. **2019**(3), 121–151 (2019), https://doi.org/10.13154/tosc.v2019.i3.121-151