Cryptanalysis of Reduced Round SPEEDY

Raghvendra Rohit[1] and Santanu Sarkar[2]

[1] Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
raghvendra.rohit@tii.ae
[2] Indian Institute of Technology Madras, Chennai, India santanu@iitm.ac.in

Abstract. SPEEDY is a family of ultra low latency block ciphers proposed by Leander, Moos, Moradi and Rasoolzadeh at TCHES 2021. Although the designers gave some differential/linear distinguishers for reduced rounds, a concrete cryptanalysis considering key recovery attacks on SPEEDY was completely missing. The latter is crucial to understand the security margin of designs like SPEEDY which typically use low number of rounds to have low latency. In this work, we present the first third-party cryptanalysis of SPEEDY-r-192, where $r \in \{5, 6, 7\}$ is the number of rounds and 192 is block and key size in bits. We identify cube distinguishers for 2 rounds with data complexities 2^{14} and 2^{13}, while the differential/linear distinguishers provided by designers has a complexity of 2^{39}. Notably, we show that there are several such cube distinguishers, and thus, we then provide a generic description of them. We also investigate the structural properties of 13-dimensional cubes and give experimental evidence that the partial algebraic normal form of certain state bits after two rounds is always the same. Next, we utilize the 2 rounds distinguishers to mount a key recovery attack on 3 rounds SPEEDY. Our attack require $2^{17.6}$ data, $2^{25.5}$ bits of memory and $2^{52.5}$ time. Our results show that the practical variant of SPEEDY, i.e., SPEEDY-5-192 has a security margin of only 2 rounds. We believe our work will bring new insights in understanding the security of SPEEDY.

Keywords: SPEEDY · Cube attacks · Block cipher

1 Introduction

Lightweight ciphers are designed with the aim of achieving implementation-specific properties such as low gate count, low latency, and low power and energy consumption. It is often difficult to obtain all these properties in a single design, and thus, the spectrum of lightweight ciphers (considering gate count, latency, power and energy) is too wide and continuously evolving. Some of the ciphers targeting low gate count are block ciphers, for example, PRESENT [10], KATAN [13], LED [17], Piccolo [26], SIMON [7] and GIFT [6], and stream ciphers such as Grain [21], Mickey [3] and Trivium [14].

The second key property is the latency which is defined as the time taken between the moment an input data is given to cipher and the corresponding output is obtained. Low latency is highly desirable in applications like encryption

of memory bus and storage systems where entire encryption and decryption should take place within the shortest possible delay. Since for many stream ciphers, the high number of clock cycles are required for the initialization phase, these are not suitable for low latency.

The first lightweight block cipher in literature which was aimed for low latency is PRINCE [11]. The design principles of PRINCE with slight variations were later adopted in QARMA [2] and PRINCEv2 [12]. Mantis is another family of low latency tweakable block ciphers [8]. Another block cipher, Midori [4], whose primary aim was low energy, also has relatively small latency.

Very recently, Leander *et al.* proposed SPEEDY [25]. It is a family of ultra low latency block ciphers that targets high-end CPUs and efficient hardware implementations (in terms of latency). In particular, one instance SPEEDY-6-192 consists of 6 rounds with 192-bit block and 192-bit key. The authors showed that its execution time is faster in hardware than any other known encryption primitives like Even-Mansour block cipher with Gimli as its core primitive [16,9] and Orthros pseudorandom function [5]. From security perspective, the authors claimed 128-bit security for SPEEDY-6-192. They also claimed that 7 rounds SPEEDY with 192-bit block and 192-bit key achieves full 192-bit security. Moreover, they proposed a 5-round variant SPEEDY-5-192 and mentioned that "SPEEDY-5-192 provides a decent security level that is sufficient for many practical applications ($\geq 2^{128}$ time complexity when data complexity is limited to $\leq 2^{64}$)".

In this paper, we investigate the security of SPEEDY for reduced rounds using cube attack. We unveil new distinguishers, their structural properties, and key recovery attacks on SPEEDY which were not reported before. Table 1 gives a summary of attacks on SPEEDY till date. In what follows, we summarize our contributions.

Our Contributions. We report the first third-party security evaluation of the SPEEDY family of block ciphers. In particular, we present practical distinguishers for 2 rounds, and key recovery attacks that can reach up to 3 rounds for all three instances of SPEEDY. We now list our contributions.

1. **Practial distinguishers for 2 rounds:** We identify generic 14-dimensional cubes whose cube-sum[3] in rows 1, 2 and 3 of state (arranged in 6 rows and 32 columns) after two full rounds is always zero. We also find 13-dimensional cubes for which cube-sum value of state bits after 2 rounds at indices $i, 31+i$ and $62+i$ is always equal, for all $32 \leq i \leq 63$. Moreover, we provide experimental evidence for the same and conjecture that the partial algebraic normal form of these bits ($i, 31+i$ and $62+i$) is always the same. In total, we find 32 such cubes for both cases.
   The source codes of the distinguishers are available on request for verification.
2. **Key recovery attack on 3 rounds:** We present a key recovery attack on 3-round SPEEDY with $2^{17.6}$ data, $2^{25.5}$ bits of memory and $2^{52.5}$ time. To

[3] XOR-ing the evaluation of a state bit at all possible 2^{14} values of cube variables.

Table 1: Summary of attacks on SPEEDY

Distinguishers					
Method	#Rounds	Data	Time	Memory (bits)	Source
Differential and linear†	2	2^{39}	2^{39}	-	[25]
	3	2^{69}	2^{69}	-	[25]
Cube	2	2^{14}	2^{14}	-	Section 3.2
Cube	2	2^{13}	2^{13}	-	Section 3.3
Key recovery					
Method	#Rounds	Data	Time	Memory (bits)	Source
Integral	3	$2^{17.6}$	$2^{52.5}$	$2^{25.2}$	Section 4.1

†: No exact trails are provided in the paper. The data in column 3, for instance, 2^{39} corresponds to the upper bound on the probability (2^{-39}) of a differential (linear) trail.

achieve this, we use the 2-round distinguisher (with cube size 14) and append one round (from decryption side) for key recovery. It is worth noting that a 2-round differential distinguisher (from designers) can be used to mount a key-recovery attack on 3 rounds. However, the attack complexities will be larger than our proposed 3-round cube attack.

Our key recovery attack is applicable to all three instances of SPEEDY, i.e., SPEEDY-5-192, SPEEDY-6-192 and SPEEDY-7-192, reduced to 3 rounds. Interestingly, after our attack, the security margin of SPEEDY-5-192 is reduced to only 2 rounds.

Outline of the Paper. The rest of the paper is organized as follows. Section 2 gives the specification of SPEEDY and the basics of Boolean functions and cube attacks. In Section 3, we present our low data complexity distinguishers for 2 rounds of SPEEDY along with their structural properties. Section 4 gives a detailed analysis of key recovery attacks on 3 rounds SPEEDY. Finally, we conclude the paper with relevant research directions in Section 5.

2 Preliminaries

In this section, we first describe the specification of SPEEDY along with its instances and their security claims. We then briefly recall basic theory of Boolean functions and cube attacks which are required for our attacks on SPEEDY.

2.1 Specification of SPEEDY

SPEEDY is a family of ultra low latency block ciphers proposed by Leander *et al.* at TCHES 2021 [25]. SPEEDY-r-6ℓ denotes one instance of this family with block and key size 6ℓ and r rounds. It takes as inputs a 6ℓ-bit plaintext P and a 6ℓ-bit secret key K and outputs the 6ℓ-bit ciphertext C after applying the round function $\mathcal{R}_j$ sequentially r times for $j = 0, \cdots, r-1$.

We consider the 6ℓ-bit state as a $6 \times \ell$ binary matrix. In the original design specification, the authors considered the state as a $\ell \times 6$ matrix. However, for the simplicity of analysis and efficient software implementation[4], we choose to view the state as a $6 \times \ell$ matrix. The round function $\mathcal{R}_j$ and key schedule are then modified accordingly, and the test vectors are matched with the author's implementation to verify the correctness of our representation. [5] A high level overview of SPEEDY is shown in Figure 1 where the round function is given by

$$\mathcal{R}_j = \begin{cases} \mathsf{RK}^{r-1} \circ \mathsf{SB} \circ \mathsf{SR} \circ \mathsf{SB} \circ \mathsf{RK}^{r} & \text{for the last round,} \\ \mathsf{RK}^{j} \circ \mathsf{SB} \circ \mathsf{SR} \circ \mathsf{SB} \circ \mathsf{SR} \circ \mathsf{MR} \circ \mathsf{AC} & \text{otherwise.} \end{cases} \tag{1}$$

Note that to keep consistency between Figure 1 and Equation 1, we perform the operations from left to right for an input of 6ℓ-bit state.

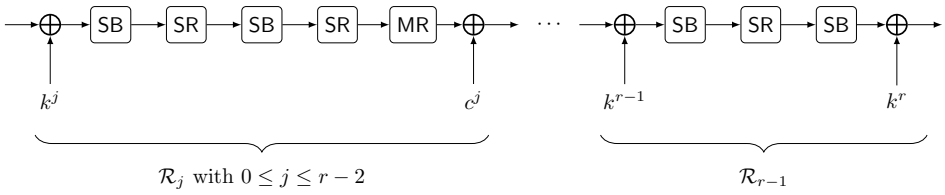

Fig. 1: r rounds of SPEEDY block cipher

We now describe the core components of the round function following our representation of SPEEDY. We use $X = x_0, \cdots, x_{6\ell-1}$ and $Y = y_0, \cdots, y_{6\ell-1}$ to represent intermediate states. We sometimes write $X = X_0 \| X_1 \| X_2 \| X_3 \| X_4 \| X_5$ and $Y = Y_0 \| Y_1 \| Y_2 \| Y_3 \| Y_4 \| Y_5$ where $X_i = (x_{\ell \cdot i}, \cdots, x_{\ell \cdot i + 31})$ and $Y_i = (y_{\ell \cdot i}, \cdots, y_{\ell \cdot i + 31})$ denote the i-th row of X and Y, respectively. The operations SB, SR, MR, AC and RK are explained in detail as follows.

SubBox (SB). A 6-bit Sbox is applied on each of the columns (see Figure 2). Let $(x_0, x_1, x_2, x_3, x_4, x_5)$ and $(y_0, y_1, y_2, y_3, y_4, y_5)$ denote the input and output of the Sbox, respectively. Then the Sbox is given in Table 2. Note that here x_i and y_i are the bits of row X_i and Y_i, respectively.

[4] From point of cryptanalysis.
[5] Test vectors are provided along with the codes.

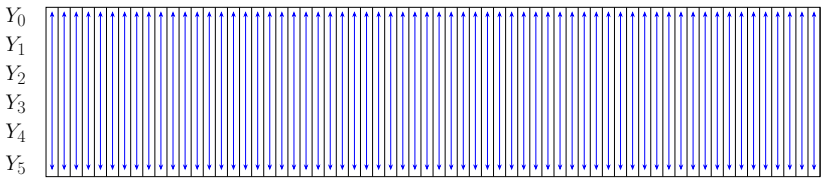

Fig. 2: SubBox SB

Table 2: SPEEDY SBox

x_0x_1	$x_2x_3x_4x_5$															
	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 (SR). As shown in Figure 3, the i-th row of state Y is rotated left by i bits. We have, $Y_i \leftarrow Y_i \lll i$ for $0 \leq i \leq 5$ where $\lll$ is a left cyclic shift operation.

Y_0
$Y_1 \lll 1$
$Y_2 \lll 2$
$Y_3 \lll 3$
$Y_4 \lll 4$
$Y_5 \lll 5$

Fig. 3: ShiftRows SR

MixRows (MR). A cyclic binary matrix is multiplied to each row of the state. When $\ell = 32$, we have the 192 version. For this version, given an input $(x_0, \cdots, x_{31}) \in \mathbb{F}_2^{32}$, MR computes the output $(y_0, \cdots, y_{31}) \in \mathbb{F}_2^{32}$ as follows.

$$y_i = x_i \oplus x_{i+1} \oplus x_{i+5} \oplus x_{i+9} \oplus x_{i+15} \oplus x_{i+21} \oplus x_{i+26}, \text{ for } 0 \leq i \leq 31, \quad (2)$$

where the subscripts are computed modulo 32.

Add Constant (AC). A 6ℓ-bit round constant c^j is XORed to the state, i.e., $Y = X \oplus c^j$. The round constants following our representation of state are given in Appendix A.

Add Round Key (RK). A 6ℓ-bit round key k^j is XORed to the state, i.e., $Y = X \oplus k^j$.

Key Scheduling Algorithm. A 6ℓ-bit master key K is used to generate round keys k^j. The first round key k^0 is taken directly from K, i.e., $k^0 = K$. Other round keys k^j for $1 \leq j \leq r$ are generated by applying the bit-wise permutation P on k^j. We omit the details of the permutation P as this is not necessary for our attack. The reader may refer to [25] for more details on the key scheduling algorithm.

In the following, we denote $(x_0^j, \cdots, x_{191}^j)$ and $(k_0^j, \cdots, k_{191}^j)$ as the input state to j-th round and j-th round key, respectively.

2.2 SPEEDY Instances and Security Claims

The authors chose $\ell = 32$ and provided three instances of SPEEDY, namely SPEEDY-5-192, SPEEDY-6-192 and SPEEDY-7-192. They expect that SPEEDY-6-192 and SPEEDY-7-192 provide 128-bit security and 192-bit security, respectively. For SPEEDY-5-128, the claimed time complexity is at least 2^{128} when data is limited to 2^{64}.

2.3 Cube Attacks

It is well known that $\mathbb{F}_2^n$ is a vector space of dimension n over the field $\mathbb{F}_2 = \{0, 1\}$. A Boolean function f in n variables is a map from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $\mathbb{B}_n$ be the set of all n-variable Boolean functions, then we have $|\mathbb{B}_n| = 2^{2^n}$. A Boolean function $f \in \mathbb{B}_n$ can be expressed as a polynomial in n variables over $\mathbb{F}_2$ as

$$f(x_0, \cdots, x_{n-1}) = \sum_{a \in \mathbb{V}_n} C_a x_0^{a_0} \cdots x_{n-1}^{a_{n-1}} \ , \tag{3}$$

is called as algebraic normal form (ANF for short) of f, where $C_a \in \mathbb{F}_2$, $a = (a_0, \cdots, a_{n-1})$ and $\mathbb{V}_n$ is the set consisting of all possible values of a. The number of variables in the highest order monomial with non-zero coefficient is called the algebraic degree, or simply the degree of f. In the ANF form of any random element of $\mathbb{B}_n$, each monomial (and in particular, the highest degree monomial $x_0 \cdot x_1 \cdots x_{n-1}$) appears with probability $\frac{1}{2}$.

Let $v = (v_0, \cdots, v_{m-1})$ be m public variables and $k = (k_0, \cdots, k_{n-1})$ be n secret variables. Then, in the context of symmetric ciphers, each output bit can be regarded as a Boolean function $f : \mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{F}_2$ given by

$$f(v, k) = \sum_{u \in \mathbb{V}_m} \sum_{w \in \mathbb{V}_n} C_{u,w} v_0^{u_0} \cdots v_{m-1}^{u_{m-1}} k_0^{w_0} \cdots k_{n-1}^{w_{n-1}}, \tag{4}$$

where $u_i, w_j, \in \mathbb{F}_2$ for $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$ and $C_{u,w} \in \mathbb{F}_2$.

The cube attack proposed in [30,15] analyzes a keyed Boolean function as a black-box polynomial which is tweakable in public variables. Given a set of indices $\mathcal{I} = \{i_0, \cdots, i_{d-1}\} \subseteq \{0, \cdots, m-1\}$ and $\bar{\mathcal{I}} = \{0, \cdots, m-1\} \backslash \mathcal{I}$, Equation 4 can be viewed as

$$f(v, k) = v_{i_0} \cdots v_{i_{d-1}} \cdot t(\underset{i \in \bar{\mathcal{I}}}{v_i}; k_0, \cdots, k_{n-1}) + q(v_0, \cdots, v_{m-1}, k_0, \cdots, k_{n-1}), \tag{5}$$

where each monomial in the Boolean function q misses at least one variable from $v[\mathcal{I}] = \{v_i \mid i \in \mathcal{I}\}$. Following the terminology of cube attacks, we denote $\mathcal{I}$, $v[\mathcal{I}]$ and a Boolean function $t(\cdot)$ as the *cube indices* set, *cube variables* set, and the *superpoly* of cube monomial $\prod_{i\in\mathcal{I}} v_i$, respectively.

One can see that XOR-ing the evaluation of f at all possible 2^d values of $v_{i_0}, \cdots, v_{i_{d-1}}$ (called as *cube sum* and given by $\mathcal{C}_{v[\mathcal{I}]}$), we have

$$\bigoplus_{\mathcal{C}_{v[\mathcal{I}]}} f(v,k) \coloneqq \sum_{(v_{i_0},\ldots,v_{i_{d-1}})\in\mathbb{F}_2^d} f(v,k) = t(\underset{i\in\bar{\mathcal{I}}}{v_i}; k_0, \cdots, k_{n-1}). \tag{6}$$

Cube tester [1] is an algorithm which can distinguish a cipher from random source. The presence of monomials, balancedness, constantness, presence of linear variables, presence of neutral variables are some testing properties which can detect non-randomness in superpoly of a Boolean function. Recently, cube attacks have gained attention due to the introduction of the division property [27,29] based automated techniques which can provide information of a superpoly [28,18,19,23,22,20].

3 Practical Distinguishers for Two Rounds SPEEDY

In this section, we present (experimental) practical distinguishers for two full rounds of SPEEDY. We first explain our core observation behind the distinguishers. Next, we present two generic distinguishers with data complexities 2^{14} and 2^{13}. We also unveil some unexpected properties of the second distinguisher and show that for certain state bits, a part of the algebraic normal form of these state bits is always the same. In the end, we discuss the possibility (with current challenges) of their proof.

3.1 Core Idea of Distinguishers

Our main idea is to reduce the algebraic degree of the output bits after 1 full round, i.e., $\mathsf{SB}\circ\mathsf{SR}\circ\mathsf{SB}\circ\mathsf{SR}\circ\mathsf{MR}$. Note that the degree of the output bits in rows 0, 1, 2, 3, 4 and 5 after 1 round are 19, 15, 13, 13, 13 and 20, respectively. To reduce these degrees, we look at the ShiftRows property of the round function, i.e., row i is cyclically left shifted by i bits (for $0 \le i \le 5$).

For instance, consider 6 cube variables in the 0-th Sbox as shown in Figure 4. After the SB operation, the output bits 0, 32, 64, 96, 128 and 160 have algebraic degrees of 5, 3, 3, 3, 4 and 5, respectively. Now, after the SR operation, these monomials will shift to Sboxes 0, 31, 30, 29, 28 and 27. Applying SB on these Sboxes will not change the algebraic degree as the monomials are in distinct Sboxes. Now, since $\mathsf{SR}\circ\mathsf{MR}\circ\mathsf{AC}$ is a linear operation, the algebraic degree of the state bits after 1 round is at most 5. The diffusion of these cube variables is shown in Figure 4.

To have further degrees of freedom, we select another 6 cube variables in the 6-th Sbox as the last shift offset is 5. Thus, after round 1, the algebraic degree of state bits in 12 cube variables is at most 5 (compared to 12).

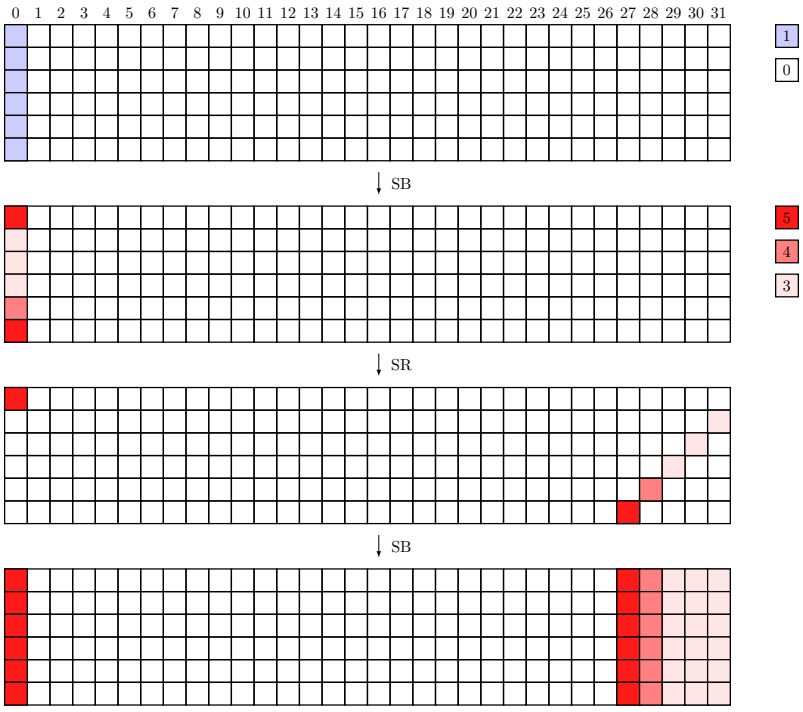

Fig. 4: Diffusion of cube variables for $\mathsf{SB} \circ \mathsf{SR} \circ \mathsf{SB}$. $\mathsf{SR} \circ \mathsf{MR} \circ \mathsf{AC}$ is omitted as it is linear and will not affect the degree. The colors represent the degree value as shown on the right side of the figure.

3.2 Distinguishers with 2^{14} Data

Recall that x_i^2 denotes the i-th bit of state after 2 rounds. We find multiple cube indices sets $\mathcal{I}$ with $|\mathcal{I}| = 14$ such that

$$\bigoplus_{\mathcal{C}_{v[\mathcal{I}]}} x_i^2 = 0, \quad \text{for all } i \in \{32, \cdots, 127\}. \tag{7}$$

We start with an example of one such $\mathcal{I}$ in Example 1.

Example 1. Consider $\mathcal{I} = \{0, 32, 64, 96, 128,\ 160, 6, 38, 70, 102, 134, 166, 12, 18\}$ as shown in Figure 5. Experimentally we checked the validity of $\mathcal{I}$ with 2^{16} random keys and for each key we set random non-cube variables. In all 2^{16} experiments, the superpolies at positions $\{32, \cdots, 127\}$ (green squares in Figure 5) after 2 rounds are always zero.

We now give a generic description of such $\mathcal{I}$'s in Observation 1.

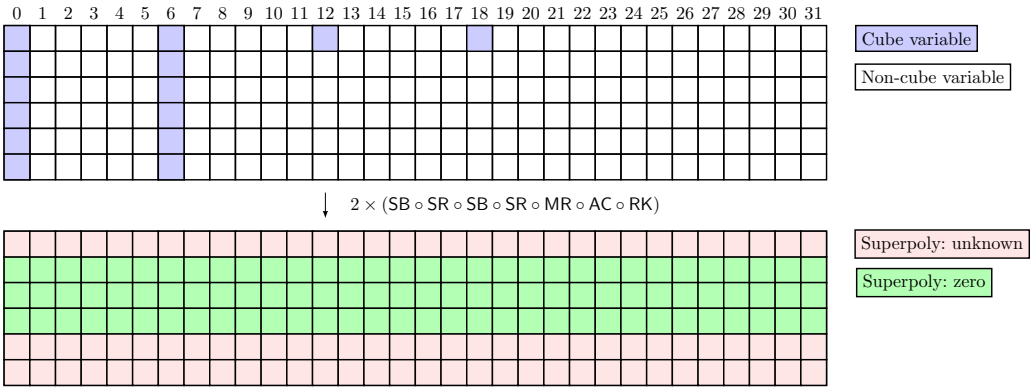

Fig. 5: A 2-round cube distinguisher with 2^{14} data

Observation 1 (Generic 14-dimensional cube) *Let* $0 \leq n \leq 31$. *Define*

$$\begin{aligned}
\mathcal{S}_n &\coloneqq \{n, 32+n, 64+n, 96+n, 128+n, 160+n\} \\
\mathcal{S}_{6+n \bmod 32} &\coloneqq \{i, 32+i, 64+i, 96+i, 128+i, 160+i \mid i \equiv 6+n \bmod 32\} \\
\mathcal{S}_{12+n \bmod 32} &\coloneqq \{12+n \bmod 32\} \\
\mathcal{S}_{18+n \bmod 32} &\coloneqq \{18+n \bmod 32\} \\
\mathcal{I}_n &\coloneqq \mathcal{S}_n \bigcup \mathcal{S}_{6+n \bmod 32} \bigcup \mathcal{S}_{12+n \bmod 32} \bigcup \mathcal{S}_{18+n \bmod 32}.
\end{aligned} \tag{8}$$

Then

$$\bigoplus_{\mathcal{C}_{v[\mathcal{I}_n]}} x_i^2 = 0, \ \textit{for all } i \in \{32, \cdots, 127\}. \tag{9}$$

Experimental Verification of Observation 1. For $0 \leq n \leq 31$, and for each $\mathcal{I}_n$, we take 2^{16} random keys and set non-cube variables as some random values. We then check the value of superpolies at positions $\{32, \cdots, 127\}$ after 2 rounds. In total, we have $2^{16} \cdot 2^5 \cdot (32 \times 3)$ superpolies. We observed that all superpolies are equal to zero.

Remark 1. The distinguisher presented in Observation 1 is very unique. For instance, one may think of first choosing 4 Sboxes which are at a distance of 6, and then select 14 (out of 4×6) variables in these Sboxes as cube variables. But this approach does not give a similar distinguisher. A counter example is $\mathcal{I} = \{0, 32, 64, 96, 128,\ 160, 6, 38, 70, 102, 134, 166, 12, 30\}$.

3.3 Distinguishers with 2^{13} Data

The 14 size cube in the previous section gives a distinguisher with probability 1. Thus, it is normal to see if we decrease the cube dimension what is the impact on probability. Accordingly, we remove 1 variable from the 14 size cube and observe the behavior of superpolies. We start with an example of 13-dimensional cube and then provide the general description of such cubes.

Example 2. Consider $\mathcal{J} = \{0, 32, 64, 96, 128,\ 160, 6, 38, 70, 102, 134, 166, 12\}$ as shown in Figure 6. We computed the cube sum for $\mathcal{J}$ with 2^{16} random keys and for each key we set non-cube variables as random values. In all 2^{16} experiments, we observe patterns[6] similar to Figure 6. For instance, as shown in Figure 6, the superpolies of state bits (35, 66, 97), (40, 71, 102) and (60, 91, 122) are equal to (1, 1, 1). More precisely, for all $32 \leq i \leq 63$, the superpolies (after 2 rounds) at positions $i, (i-1)+32, (i-2)+64$ are always equal.

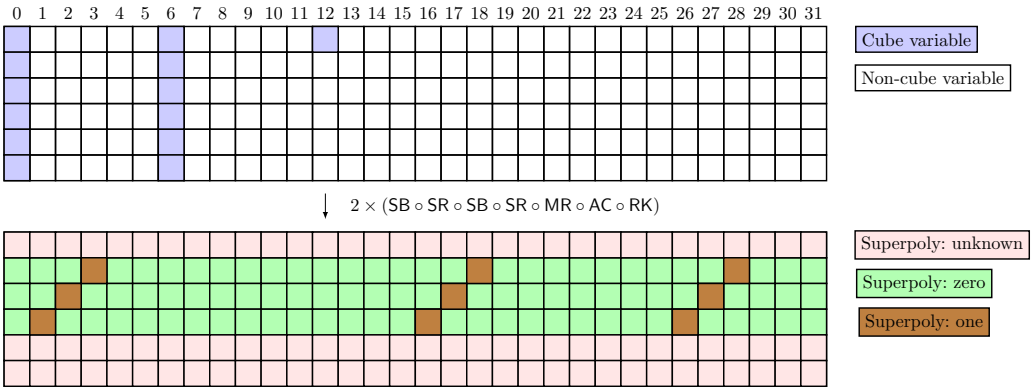

Fig. 6: A 2-round cube distinguisher with 2^{13} data

Now, analogous to Observation 1, we give a generic description of 13-dimensional cubes in Observation 2.

Observation 2 (Generic 13-dimensional cube) *Let* $0 \leq n \leq 31$. *Define*

$$\begin{aligned}
\mathcal{S}_n &\coloneqq \{n, 32+n, 64+n, 96+n, 128+n, 160+n\} \\
\mathcal{S}_{6+n \bmod 32} &\coloneqq \{i, 32+i, 64+i, 96+i, 128+i, 160+i \mid i \equiv 6+n \bmod 32\} \\
\mathcal{S}_{12+n \bmod 32} &\coloneqq \{12+n \bmod 32\} \\
\mathcal{J}_n &\coloneqq \mathcal{S}_n \bigcup \mathcal{S}_{6+n \bmod 32} \bigcup \mathcal{S}_{12+n \bmod 32}.
\end{aligned} \tag{10}$$

Then

$$\bigoplus_{\mathcal{C}_{v[\mathcal{J}_n]}} x_i^2 = \bigoplus_{\mathcal{C}_{v[\mathcal{J}_n]}} x_{i+31}^2 = \bigoplus_{\mathcal{C}_{v[\mathcal{J}_n]}} x_{i+62}^2, \text{ for all } i \in \{32, \cdots, 63\}. \tag{11}$$

Experimental Verification of Observation 2. For $0 \leq n \leq 31$, and for each $\mathcal{J}_n$, we take 2^{16} random keys and set non-cube variables as some random values. We then check the value of superpolies (after 2 rounds) corresponding to the triplet $(i, i+31, i+62)$ for $32 \leq i \leq 63$. In total, we have $2^{16} \cdot 2^5$ triplets. We observed that in each triplet, superpolies values are always equal.

[6] This is one of the example of a pattern.

Observations on the Distinguisher. It is somewhat unexpected that superpolies in the triplet $(i, i+31, i+62)$ are always equal. Our experimental results suggest that this behavior happens for almost all keys (we further checked Observation 2 for another 2^{20} keys). Since $(i, i+31, i+62)$ can be (0, 0, 0) or $(1, 1, 1)$, it can be argued that the superpolies are not constant. We believe this happens only if the partial algebraic normal (containing the cube monomial and superpoly) of these state bits after 2 rounds is always same. Since we can not prove this fact theoretically (albeit this holds experimentally), we present it as conjecture below.

Conjecture 1. Let $0 \leq n \leq 31$ and $\mathcal{J}_n$ as defined in Observation 2. Then for all $i = 32, \cdots, 63$, the ANF of state bits $i, i+31$ and $i+62$ is given by

$$\begin{aligned} x_i^2 &= f_i + (\prod_{j \in \mathcal{J}_n} v_j) \cdot t_i \\ x_{i+31}^2 &= f_{i+31} + (\prod_{j \in \mathcal{J}_n} v_j) \cdot t_i \\ x_{i+62}^2 &= f_{i+62} + (\prod_{j \in \mathcal{J}_n} v_j) \cdot t_i \end{aligned} \tag{12}$$

where t_i is the superpoly corresponding to cube indices $\mathcal{J}_n$ and f_i, f_{i+31}, f_{i+62} are Boolean functions similar to the Boolean function q in Equation 5.

3.4 Discussion on the Proofs of Distinguishers

In all our experimental results related to Observation 1 and 2, we did not find a counter example, i.e., a key for which these two observations do not hold. Thus, we expected that they could be proved mathematically. As such, we tried the following two approaches for the proofs.

SAGE based Proof. We set the cube variables and 192 key bits as symbolic variables. Then, we checked the maximum degree in cube variables after round 2. Because of the high algebraic degree (including key variables), our SAGE code always ran out of memory. Thus, we chose to find the degree by selecting a random key and setting non-cube variables as zero. We find that for 14-dimensional cube, the degree is at most 13 in rows 1, 2 and 3 of the state. For 13-dimensional cube, we find that the algebraic degree is at most 12 in majority of the state bits. This provides another evidence for our experimental distinguishers.

Division property based Proof. We modeled the three subset bit based division property [31,19] propagation of one round SPEEDY using MILP. We find that even for a single round, the superpolies of a 5-dimensional cube are too dense. Since the algebraic degree of 1 round is at most 20, we expect that this tool may become slow for two consecutive rounds.

The source codes of the SAGE implementation and the division property models are also available to readers on request.

4 Key Recovery Attacks

In this section, we present a 3-round key recovery attack that is applicable to SPEEDY-5-192, SPEEDY-6-192 and SPEEDY-7-192. Our attack is based on the principles of integral cryptanalysis [24] and utilize the 2-round distinguishers as described before. Before proceeding to the attack, we first recall some notations that will be used throughout this section.

The vectors $(x_0^j, \ldots, x_{191}^j)$ and $(k_0^j, \ldots, k_{191}^j)$ denote the input state at j-th round and j-th round key, respectively. Also, $(x_0^0, \ldots, x_{191}^0)$ and $(x_0^r, \ldots, x_{191}^r)$ represent the plaintext and the ciphertext, respectively. Further, note that recovering a round key is equivalent to recovering the master key. In our attacks, we aim to recover the last round key k^r which is also the post-whitening key.

4.1 3-Round Key Recovery Attack

Figure 7 shows the high level overview of the 3-round key recovery attack on SPEEDY. We use a 2-round cube distinguisher (cube size 14, Example 1) and append 1-round for the key recovery. In our attack, we use the fact that each state bit after $\mathsf{SB}^{-1} \circ \mathsf{SR}^{-1} \circ \mathsf{SB}^{-1}$ depends only on 36 bits of key and 36 bits of the ciphertext. For instance, the bits in column 0 depends on the ciphertext and last round key bits from columns 0, 31, 30, 29, 28 and 27. More precisely, a column i after $\mathsf{SB}^{-1} \circ \mathsf{SR}^{-1} \circ \mathsf{SB}^{-1}$ depends on columns $i, i-1, \cdots, i-5$ of ciphertext and key k^3. [7] Thus, in order to do partial decryption with mutually disjoint subkey bits (see Equation 13), we choose columns 0, 6, 12, 18 and 24. We match the decrypted value of a state bit with the cube sum value in bits 1, 2 and 3 for each of these columns (see green squares in Figure 7).

We now explain the detailed attack steps along with their respective complexities. For $i = 0, 6, 12, 18$ and 24, we first define

$$\begin{aligned}\mathsf{SK}[i] \coloneqq \{\, & k_i^3, k_{32+i}^3, k_{64+i}^3, k_{96+i}^3, k_{128+i}^3, k_{160+i}^3, \\ & k_{i-1}^3, k_{31+i}^3, k_{63+i}^3, k_{95+i}^3, k_{127+i}^3, k_{159+i}^3, \\ & k_{i-2}^3, k_{30+i}^3, k_{62+i}^3, k_{94+i}^3, k_{126+i}^2, k_{158+i}^3, \\ & k_{i-3}^3, k_{29+i}^3, k_{61+i}^3, k_{93+i}^3, k_{125+i}^3, k_{157+i}^3, \\ & k_{i-4}^3, k_{28+i}^3, k_{60+i}^3, k_{92+i}^3, k_{124+i}^3, k_{156+i}^3, \\ & k_{i-5}^3, k_{27+i}^3, k_{59+i}^3, k_{91+i}^3, k_{123+i}^3, k_{155+i}^3 \,\}\end{aligned} \tag{13}$$

as partial bits of k^3. While computing $\mathsf{SK}[i]$, the subscripts of key bits are taken modulo 192. Note that $\mathsf{SK}[i]$'s are mutually disjoint. Similarly, we define mutually disjoint sets for the ciphertext bits as follows.

[7] Column numbers taken modulo 32.

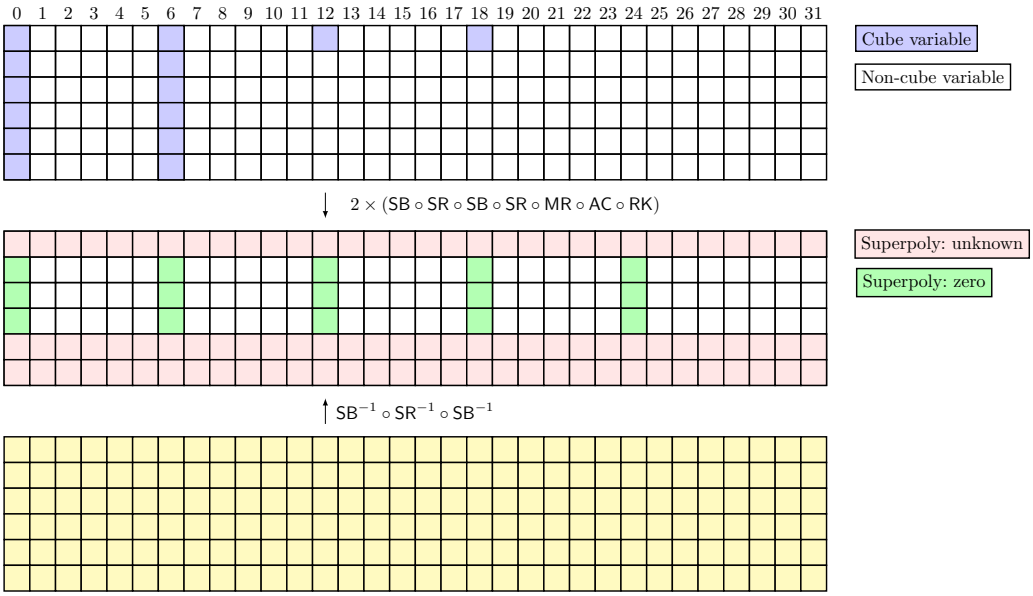

Fig. 7: An overview of the 3-round key recovery attack. After partial decryption, a matching is done at positions as shown by green squares (after 2 rounds from the encryption side).

$$\begin{aligned}\mathsf{CT}[i] \coloneqq \{\, & x_i^3, x_{32+i}^3, x_{64+i}^3, x_{96+i}^3, x_{128+i}^3, x_{160+i}^3, \\ & x_{i-1}^3, x_{31+i}^3, x_{63+i}^3, x_{95+i}^3, x_{127+i}^3, x_{159+i}^3, \\ & x_{i-2}^3, x_{30+i}^3, x_{62+i}^3, x_{94+i}^3, x_{126+i}^3, x_{158+i}^3, \\ & x_{i-3}^3, x_{29+i}^3, x_{61+i}^3, x_{93+i}^3, x_{125+i}^3, x_{157+i}^3, \\ & x_{i-4}^3, x_{28+i}^3, x_{60+i}^3, x_{92+i}^3, x_{124+i}^3, x_{156+i}^3, \\ & x_{i-5}^3, x_{27+i}^3, x_{59+i}^3, x_{91+i}^3, x_{123+i}^3, x_{155+i}^3 \,\}\end{aligned} \tag{14}$$

The attack steps proceed as follows.

Step 1: Setting cube and non-cube variables. For $\mathcal{I} = \{0, 32, 64, 96, 128, 160, 6, 38, 70, 102, 134, 166, 12, 18\}$, set $x_i^0 = v_i$ for $i \in \mathcal{I}$, and set x_i^0 as a random bit, for $i \in \{0, \cdots, 191\} \setminus \mathcal{I}$.

Step 2. Querying SPEEDY oracle and storing ciphertexts. Let $v = (v_0, v_{32}, \cdots, v_{12}, v_{18})$. For $v = 0$ to $2^{14} - 1$, query 3-round SPEEDY oracle and store the ciphertexts in the set $\mathcal{C}$. This step requires 2^{14} encryption queries (1 query = 3-round SPEEDY) and $2^{14} \cdot 192$ bits of memory.

Step 3. Key recovery phase. For $i = 0, 6, 12, 18, 24$, we recover key bits as follows.

3.1 For each guess sk_i of $\mathsf{SK}[i]$, we compute the values $\bigoplus x^2_{32+i}, \bigoplus x^2_{64+i}$ and $\bigoplus x^2_{96+i}$ by partially decrypting all 2^{14} ciphertexts in $\mathcal{C}$. Note that while doing the partial decryption, we only need the information of 36 bits of each ciphertext. The latter is captured by the set $\mathsf{CT}[i]$ (see Equation 14).
3.2 If only $\bigoplus x^2_{32+i} = 0, \bigoplus x^2_{64+i} = 0$ and $\bigoplus x^2_{96+i} = 0$, then we add sk_i as a possible 36-bit key candidate.

Step 3.1 and 3.2 require $5 \cdot 2^{36} \cdot 2^{14}$ 1-round decryption. Since we are checking the values of superpolies at 3 positions, this will reduce the key space of each $\mathsf{SK}[i]$ by 3 bits.

Step 4: Further filtering. We repeat Steps 1-3 with the reduced key space 11 more times to obtain the correct $(\mathsf{SK}[0], \mathsf{SK}[6], \mathsf{SK}[12], \mathsf{SK}[18], \mathsf{SK}[24])$. In total, Steps 1-2 require $2^{14} \cdot 12$ encryption queries and $2^{14} \cdot 192 \cdot 12$ bits of memory. However, for each iteration $j = 12, \cdots, 1$, the time complexity of Step 3 is given by $5 \cdot 2^{3 \cdot j} \cdot 2^{14}$ 1-round decryption. This is because after each iteration, the key space is reduced by 3 bits. Thus, the overall time complexity of Step 3 is given by $\sum_{j=1}^{12} 5 \cdot 2^{3 \cdot j} \cdot 2^{14} \approx 2^{52.52}$.

Step 5: Exhaustive search. Till now, we have recovered 180 bits of k^3. The remaining 12 bits can be obtained by performing an exhaustive search. This requires 2^{12} time.

Combining Steps 1-5, the entire 3-round attack has the following complexities.

$$\begin{aligned} \text{Data} &= 2^{14} \cdot 12 \approx 2^{17.58} \\ \text{Memory} &= 2^{14} \cdot 192 \cdot 12 \approx 2^{25.16} \text{ bits} \\ \text{Time} &= 2^{52.52} + 2^{12} \approx 2^{52.52} \end{aligned} \tag{15}$$

4.2 On Improving Number of Rounds for Key Recovery

It is natural to ask whether we can attack 4-round SPEEDY. Based on our current analysis, we do not see a direct way to attack 4 rounds.

The reasons are as follows: (1) We are unaware of the existence of a 2.5 and 3 round distinguisher with a complexity at most 2^{64}, and (2) the exact ANF of 1 and 1.5 rounds in forward and backward directions is extremely complicated and of high degree.

5 Conclusion

In this work, we have presented the first third-party cryptanalysis of SPEEDY family of block ciphers. We identified multiple distinguishers (in total 32+32)

for 2 rounds with data complexities 2^{14} and 2^{13}. Our second distinguisher (13-dimensional cubes) revealed an unexpected property that the partial algebraic normal form of certain state bits after 2 rounds is always equal, for which we also provided the experimental evidence. We then gave a key recovery attack on 3-round SPEEDY which requires $2^{17.6}$ data, $2^{25.5}$ bits of memory and $2^{52.5}$ time.

Although our findings may not appear to be novel, they did cover 60% and 50% rounds of SPEEDY-5-192 and SPEEDY-6-192 for the first time in the literature. We expect many more unidentified distinguishers for 2 rounds. To find them, it is important to investigate and understand the theoretical properties of the current 2-round distinguishers. Furthermore, it would be interesting to see if there are any 2.5 or 3-round cube distinguishers. Our initial analysis shows that this may require a non-trivial effort because of the high growth in algebraic degree. Overall, we believe there are lot of unanswered questions and this work (being the first one apart from designers) will provide new insights to the community in further understanding the security of SPEEDY.

6 Acknowledgements

The authors would like to thank the reviewers of Africacrypt 2022 for providing us with insightful comments to improve the quality of the paper.

A SPEEDY Round Constants

In Table 3, we list the first 6 round constants of SPEEDY.

Table 3: Round constants of SPEEDY

Round j	c^j
0	0x3903501c, 0x22145a05, 0xb46705b0, 0x2269408a, 0x5b9954ce, 0xe150791e
1	0x3a21067b, 0x32801fbe, 0x35c8cee9, 0x0d33c971, 0xfd8f9408, 0x22b25e82
2	0xbf3984a2, 0xa5b365cd, 0x5d54b65f, 0x0ff7e9ee, 0x4012012d, 0x1a5d9cd5
3	0x8eb8aff6, 0xc16d9463, 0x1ddb3cda, 0xa19c9865, 0x535f36d7, 0x5f9f7fac
4	0xe17adece, 0x3cc44c83, 0x85ccd8e4, 0xc7b3b8d5, 0xe481006d, 0x4cc7691c
5	0x7873963c, 0xc98a9bb3, 0x8006f8e7, 0x6f7cbba0, 0x4def0a1c, 0x0785d9ae

References

1. Aumasson, J., Dinur, I., Meier, W., Shamir, A.: Cube testers and key recovery attacks on reduced-round MD6 and trivium. In: Dunkelman, O. (ed.) Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5665, pp. 1–22. Springer (2009), https://doi.org/10.1007/978-3-642-03317-9_1

2. Avanzi, R.: The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. IACR Trans. Symmetric Cryptol. **2017**(1), 4–44 (2017), https://doi.org/10.13154/tosc.v2017.i1.4-44
3. Babbage, S., Dodd, M.: The MICKEY stream ciphers. In: Robshaw, M.J.B., Billet, O. (eds.) New Stream Cipher Designs - The eSTREAM Finalists, Lecture Notes in Computer Science, vol. 4986, pp. 191–209. Springer (2008), https://doi.org/10.1007/978-3-540-68351-3_15
4. 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.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015), https://doi.org/10.1007/978-3-662-48800-3_17
5. Banik, S., Isobe, T., Liu, F., Minematsu, K., Sakamoto, K.: Orthros: A low-latency PRF. IACR Trans. Symmetric Cryptol. **2021**(1), 37–77 (2021), https://doi.org/10.46586/tosc.v2021.i1.37-77
6. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 321–345. Springer (2017), https://doi.org/10.1007/978-3-319-66787-4_16
7. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. ePrint Arch. p. 404 (2013), http://eprint.iacr.org/2013/404
8. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016), https://doi.org/10.1007/978-3-662-53008-5_5
9. Bernstein, D.J., Kölbl, S., Lucks, S., Massolino, P.M.C., Mendel, F., Nawaz, K., Schneider, T., Schwabe, P., Standaert, F., Todo, Y., Viguier, B.: Gimli : A cross-platform permutation. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 299–320. Springer (2017), https://doi.org/10.1007/978-3-319-66787-4_15
10. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007), https://doi.org/10.1007/978-3-540-74735-2_31
11. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen,

S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer (2012), https://doi.org/10.1007/978-3-642-34961-4_14

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., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 483–511. Springer (2020), https://doi.org/10.1007/978-3-030-81652-0_19
13. Cannière, 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.) Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings. Lecture Notes in Computer Science, vol. 5747, pp. 272–288. Springer (2009), https://doi.org/10.1007/978-3-642-04138-9_20
14. Cannière, C.D., Preneel, B.: Trivium. In: Robshaw, M.J.B., Billet, O. (eds.) New Stream Cipher Designs - The eSTREAM Finalists, Lecture Notes in Computer Science, vol. 4986, pp. 244–266. Springer (2008), https://doi.org/10.1007/978-3-540-68351-3_18
15. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings. pp. 278–299 (2009)
16. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptol. **10**(3), 151–162 (1997), https://doi.org/10.1007/s001459900025
17. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 326–341. Springer (2011), https://doi.org/10.1007/978-3-642-23951-9_22
18. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128aead. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 466–495. Springer (2020), https://doi.org/10.1007/978-3-030-45721-1_17
19. Hebborn, P., Lambin, B., Leander, G., Todo, Y.: Lower bounds on the degree of block ciphers. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 537–566. Springer (2020), https://doi.org/10.1007/978-3-030-64837-4_18

20. Hebborn, P., Lambin, B., Leander, G., Todo, Y.: Strong and tight security guarantees against integral distinguishers. Cryptology ePrint Archive, Report 2021/1502 (2021), https://ia.cr/2021/1502
21. Hell, M., Johansson, T., Maximov, A., Meier, W.: The grain family of stream ciphers. In: Robshaw, M.J.B., Billet, O. (eds.) New Stream Cipher Designs - The eSTREAM Finalists, Lecture Notes in Computer Science, vol. 4986, pp. 179–190. Springer (2008), https://doi.org/10.1007/978-3-540-68351-3_14
22. Hu, K., Sun, S., Todo, Y., Wang, M., Wang, Q.: Massive superpoly recovery with nested monomial predictions. IACR Cryptol. ePrint Arch. p. 1225 (2021), https://eprint.iacr.org/2021/1225
23. Hu, K., Sun, S., Wang, M., Wang, Q.: An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums. In: Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. pp. 446–476 (2020)
24. Knudsen, L.R., Wagner, D.A.: Integral cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2365, pp. 112–127. Springer (2002), https://doi.org/10.1007/3-540-45661-9_9
25. 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
26. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 342–357. Springer (2011), https://doi.org/10.1007/978-3-642-23951-9_23
27. Todo, Y.: Structural evaluation by generalized integral property. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 287–314. Springer (2015), https://doi.org/10.1007/978-3-662-46800-5_12
28. Todo, Y., Isobe, T., Hao, Y., Meier, W.: Cube attacks on non-blackbox polynomials based on division property. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 250–279. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_9
29. Todo, Y., Morii, M.: Bit-based division property and application to Simon family. In: Peyrin, T. (ed.) Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9783, pp. 357–377. Springer (2016), https://doi.org/10.1007/978-3-662-52993-5_18
30. Vielhaber, M.: Breaking one.fivium by aida an algebraic iv differential attack. Cryptology ePrint Archive, Report 2007/413 (2007), https://ia.cr/2007/413

31. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 648–678 (2016). https://doi.org/10.1007/978-3-662-53887-6_24