SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2019, Vol. 62 032108:1–032108:13
https://doi.org/10.1007/s11432-018-9527-8

New observation on the key schedule of RECTANGLE

Hailun YAN[1], Yiyuan LUO[2], Mo CHEN[1] & Xuejia LAI[1,3]*

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*School of Electronics and Information, Shanghai Dianji University, Shanghai* 201306*, China;*
[3]*Westone Cryptologic Research Center, Beijing* 100070*, China*

Received 17 May 2018/Accepted 16 July 2018/Published online 29 January 2019

Abstract We evaluate the security of RECTANGLE from the perspective of actual key information (AKI). Insufficient AKI permits the attackers to deduce some subkey bits from some other subkey bits, thereby lowering the overall attack complexity or getting more attacked rounds. By considering the interaction between the key schedule's diffusion and the round function's diffusion, we find there exists AKI insufficiency in 4 consecutive rounds for RECTANGLE-80 and 6 consecutive rounds for RECTANGLE-128, although the master key bits achieve complete diffusion in 2 and 4 rounds, respectively. With such weakness of the key schedule, we give a generic meet-in-the-middle attack on 12-round reduced RECTANGLE-128 with only 8 known plaintexts. Moreover, we calculate AKI of variants of RECTANGLE as well as PRESENT. Surprisingly we find that both RECTANGLE-128 and PRESENT-128 with no key schedule involve more AKI than the original one. Based on this finding, we slightly modify the key schedule of RECTANGLE-128. Compared with the original one, this new key schedule matches better with the round function in terms of maximizing AKI. Our work adds more insight to the design of block ciphers' key schedule.

Keywords lightweight block cipher, RECTANGLE, key schedule, round function, diffusion

Citation Yan H L, Luo Y Y, Chen M, et al. New observation on the key schedule of RECTANGLE. Sci China Inf Sci, 2019, 62(3): 032108, https://doi.org/10.1007/s11432-018-9527-8

1 Introduction

During recent years, there have been quite extensive application of lightweight block ciphers in constrained environments, such as RFIDs, sensor nodes, and smart cards. There have been a wide variety of lightweight algorithms up to now, including PRESENT [1], LED [2], Midori [3], SKINNY [4], QARMA [5], GIFT [6], Simon and Speck [7]. In addition to providing (relatively) strong cryptographic security, lightweight block ciphers are also known for a lower cost than standard block ciphers in terms of hardware and software implementation. The key schedule of lightweight block ciphers, in particular, is often highly simplified. Some key schedules use round-by-round iterations with low diffusion [1, 8, 9], some do linear operations or simple permutations on master keys [10, 11], some even have no key schedule at all [2, 12].

Scientifically designing the key schedule part of block ciphers is an important but not well-understood subject. It is not yet clear what properties a good key schedule should have. Particularly for lightweight block ciphers, it remains an open problem that how to guarantee sufficient diffusion of key bits by using limited operations of the key schedule. Another issue is that, the diffusion of key schedules and the diffusion of round functions are usually evaluated independently, while the poor distribution of key

* Corresponding author (email: laix@sjtu.edu.cn)

bits on the the diffusion path of round functions leads to many attacks, such as the meet-in-the-middle attack [13].

It is pointed out by Huang and Lai in [14] that the information leakage of key bits might be resulted from the the overlapping of the diffusion of round functions and the diffusion of key schedules, thereby deserving more attention. They discussed this relation in terms of the definition of actual key information (AKI). Insufficient AKI permits the adversary to deduce some subkey bits from some other subkey bits. The missing of those key bits will further reduce time complexity of attacks, or even leading to more attacked rounds. Until now, there are mainly two types of algorithm to calculate AKI. One is the algorithm proposed by Huang et al. [15] for iterated key schedules of lightweight ciphers to search calculation dependency paths involving insufficient AKI, which is based on greedy strategy. Another automatic tool to detect such key schedule weakness is the algorithm given by Lin et al. [16] for searching key bridges. The key-bridging technique was first used in the single-key attacks on AES, which exploits the weakness of AES key schedule [17]. The existence of key bridges can be interpreted as key bits leakage, namely AKI insufficiency.

RECTANGLE, which allows fast implementations for multiple platforms, is a lightweight block cipher proposed by Zhang et al. [18] using bit-slice techniques. It adopts a substitute-permutation network (SPN) structure with 25 rounds. It has a 64-bit block size and two versions of key size: 80-bit and 128-bit. There have been many analysis results of the security of RECTANGLE. In [18], the designers of RECTANGLE presented some security analyses in the single-key model, including differential cryptanalysis [19], impossible differential cryptanalysis [20], linear cryptanalysis [21], integral cryptanalysis [22,23] and statistical saturation attack [24]. Shan et al. [25] attacked 19-round reduced RECTANGLE-80 in a related-key setting. Kosuge et al. [26] presented the first integral attack on 10-round RECTANGLE-80 and 12-round RECTANGLE-128. Sun et al. [27] constructed an 8-round higher-order integral distinguisher for RECTANGLE based on division property. Subsequently, Xiang et al. [28] found 9-round integral distinguishers of RECTANGLE by using the so-called MILP method. Sasaki and Todo [29] give the complete list of impossible differential characteristics of RECTANGLE starting and ending with 1 active nibble. There are also some results about RECTANGLE S-box [30,31] and the implementation of RECTANGLE [32–34]. Despite all this, the key schedule of RECTANGLE has not been well analysed. When the designers of RECTANGLE consider the diffusion of key bits, they see the key schedule in an isolated way. However, as mentioned above, the interaction between the diffusion of key schedules and the diffusion of round functions should also be given attention. In this paper, we evaluate the diffusion of RECTANGLE's key schedule in combination with the diffusion of its round function and give a further research on its security. We hope our work add more insight to the design of the key schedule of RECTANGLE, even some other block ciphers.

Our contributions. First, we redefine AKI in a more general way. Compared with Huang and Lai's definition, our definition is more concise and illustrates the concept more accurately. Then, we calculate the AKI of key bits distributed in the diffusion path of the round function of RECTANGLE. Our results show that there exists information leakage of key bits in (2–4)-round paths for RECTANGLE-80 and (2–6)-round paths for RECTANGLE-128. One main reason is the overlapping between the diffusion of the key schedule and the diffusion of the round function. With such weakness of the key schedule, we give a generic meet-in-the-middle attack on 12-round RECTANGLE-128 with quite low data complexity, which is only 8 plaintexts. Moreover, we analyze several variants of RECTANGLE and PRESENT. Surprisingly we find that both PRESENT-128 and RECTANGLE-128 with no key schedule involve more AKI than the original one. Inspired by the experiment results, we slightly modify the key schedule of RECTANGLE-128. Compare with the original one, this new key schedule matches better with the round function in terms of maximizing AKI.

Organization. The other parts of the paper are organized as the way in below. Section 2 gives a brief description of RECTANGLE. Section 3 introduces the notion of AKI. Section 4 presents a weakness of RECTANGLE key schedule from the perspective of AKI, which can be exploited to mount a meet-in-the-middle attack. This section also analyzes several variants of RECTANGLE with different key schedules. Section 5 proposes a new key schedule for RECTANGLE-128 by making a slight modification to the

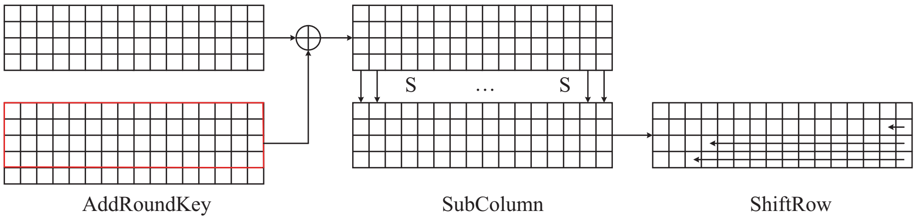

Figure 1 (Color online) The round transformation of RECTANGLE, taking the 80-bit version as an example.

original one. Section 6 concludes this paper.

2 Specification of RECTANGLE

RECTANGLE has a 64-bit block size and two versions of key size: 80-bit and 128-bit. It adopts an SPN structure with 25 rounds. The state of RECTANGLE, including the plaintext, ciphertext and intermediate value, is expressed as a rectangular array of 4 × 16 bits:

$$\begin{bmatrix} v_{15} & \cdots & v_2 & v_1 & v_0 \\ v_{31} & \cdots & v_{18} & v_{17} & v_{16} \\ v_{47} & \cdots & v_{34} & v_{33} & v_{32} \\ v_{63} & \cdots & v_{50} & v_{49} & v_{48} \end{bmatrix}.$$

The key state is expressed as a 5×16 rectangular array for the 80-bit key schedule and a 4×32 rectangular array for the 128-bit key schedule:

$$\begin{bmatrix} k_{15} & \cdots & k_1 & k_0 \\ k_{31} & \cdots & k_{17} & k_{16} \\ k_{47} & \cdots & k_{33} & k_{32} \\ k_{63} & \cdots & k_{49} & k_{48} \\ k_{79} & \cdots & k_{65} & k_{64} \end{bmatrix}, \qquad \begin{bmatrix} k_{31} & \ldots & k_2 & k_1 & k_0 \\ k_{63} & \ldots & k_{34} & k_{33} & k_{32} \\ k_{95} & \ldots & k_{66} & k_{65} & k_{64} \\ k_{127} & \ldots & k_{98} & k_{97} & k_{96} \end{bmatrix}.$$

The round transformation of RECTANGLE is a sequence of three steps: AddRoundkey, SubColumn and ShiftRow, which are illustrated in Figure 1. The final round contains an additional AddRoundkey step.

Key schedule of RECTANGLE-80. The 80-bit master key is used as the initial value WK_0 of an 80-bit key register, a rectangular array in size of 5×16 bits. The 64-bit round key RK_i $(i = 0, \ldots, 25)$ consists of the first 4 rows of the current state of the key register. After extracting RK_i, the key register WK_i is updated by three steps: Applying the S-box S at the 4 rightmost columns and the uppermost rows, employing a 1-round generalized Feistel transformation, and the rightmost 5 bits in the first row XORed with a round constant. The update of the key register is illustrated in Figure 2.

Key schedule of RECTANGLE-128. The 128-bit master key is used as the initial value WK_0 of a 128-bit key register, arranged as a rectangular array of 4×32 bits. The 64-bit round key RK_i consists of the 16 rightmost columns of the current contents of register. After extracting RK_i, the key register WK_i is updated by three steps, which is quite similar with that of RECTANGLE-80 key schedule (see Figure 3).

Refer to [18] for more details regarding the specification of RECTANGLE.

3 Actual key information

In this section, we give the definition of AKI, which was first proposed to depict the interaction between the key schedule and the round function of block ciphers by Huang and Lai in [14]. Their research is

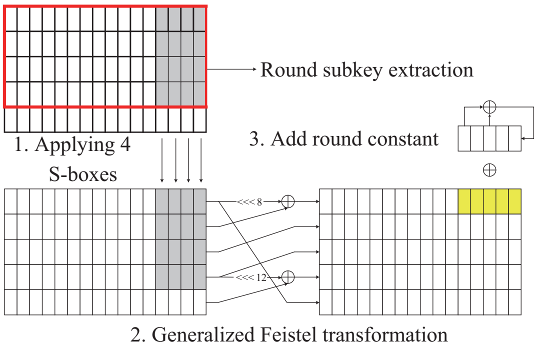

Figure 2 (Color online) Key schedule of RECTANGLE-80.

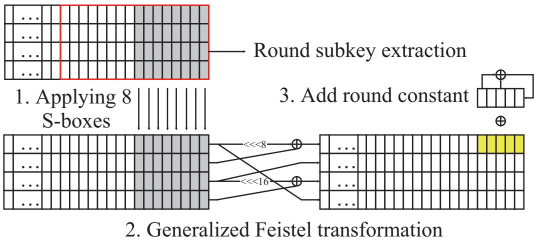

Figure 3 (Color online) Key schedule of RECTANGLE-128.

interesting and useful but has not got the deserving attention. Besides, the definition of AKI is a little complicated and hard to read. Therefore, we redefine AKI in a more general way, and explain the relative definitions by taking a toy cipher as an illustrative example. Then, we briefly introduce the algorithm to search paths involving insufficient AKI.

Given the key schedule of a block cipher, we denote the set of all key variables (both the master key and subkeys) by K, and denote the size of the set K by $|K|$.

Definition 1. Let K_0, $K' \subseteq K$. We say that K' is a reduced set of K_0 if $|K'| \leqslant |K_0|$ and all key variables in K_0 can be derived from the key variables in K' according to the key schedule. We denote it by $K' \Rightarrow K_0$.

Definition 2 (Actual key information)**.** The AKI of a key set K_0 is the minimum size of its reduced sets, which is denoted by AKI_{K_0}. That is

$$\mathrm{AKI}_{K_0} = \min_{K' \Rightarrow K_0} \{|K'|\}.$$

We denote it by AKI for simplicity if the context specifically indicates the key set K_0.

In Huang and Lai's work [14], AKI is defined by key dependency path (KDP), while KDP is defined by calculation dependency path (CDP), which is a bit complicated. We keep the main idea of AKI and redefine it in a more compact way. What is more, in their definition, AKI is a reduced set. While in our definition, AKI is the size of reduced sets. Since for a given key set, there can be several different reduced sets with the same size, we think that our definition can illustrate the concept more accurately.

Definition 3. The theoretical key information (TKI) of the key-guessing set K_0 is

$$\mathrm{TKI}_{K_0} = \min\{|K_0|, m\},$$

where m is the master key length. If AKI < TKI, we say that AKI is insufficient or AKI insufficiency or key bits leakage.

Insufficient AKI permits the adversary to get some subkey bits from some other subkey bits for free. The missing of those key bits will further reduce time complexity of attacks, or even leading to more attacked rounds. More specifically, most attacks on block ciphers, e.g., meet-in-the-middle attacks and guess-and-determine attacks, can be divided into three consecutive parts of r_1, r and r_2 rounds, such that a certain property in the middle r rounds can be verified by guessing some key-bits in the first r_1 and last r_2 rounds combined with a particular set of messages (see Figure 4). If the AKI of the key-guessing sets is insufficient, then it is possible to lower the attack complexity. With a proper data complexity, it is also possible to get more attacked rounds. For example, the zero-correlation cryptanalysis and impossible differential cryptanalysis on LBlock [16], the single-key attacks on AES [17], and the meet-in-the-middle attacks on TWINE [15, 16, 35], all benefit from AKI insufficiency.

Example 1 (AKI insufficiency for TWINE-128)**.** In [35], Biryukov et al. gave a meet-in-the-middle attack on 25-round TWINE-128 by exploiting an 11-round distinguisher, appending 5 rounds at beginning

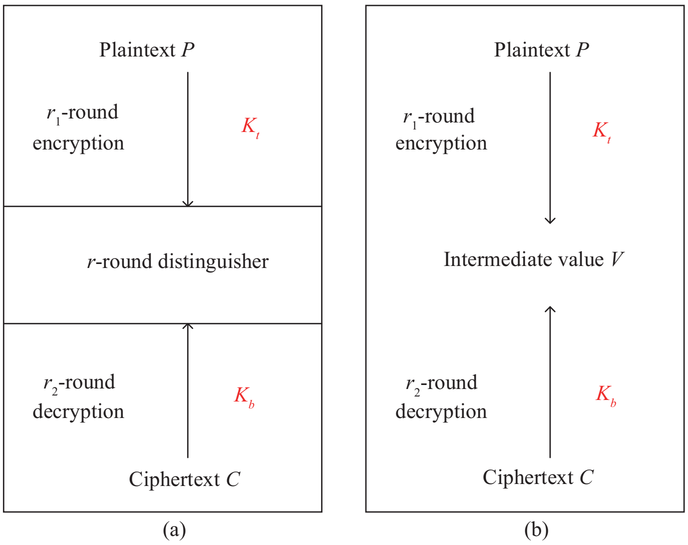

Figure 4 (Color online) Main idea of guess-and-determine attacks (a) and MITM attacks (b) on block ciphers.

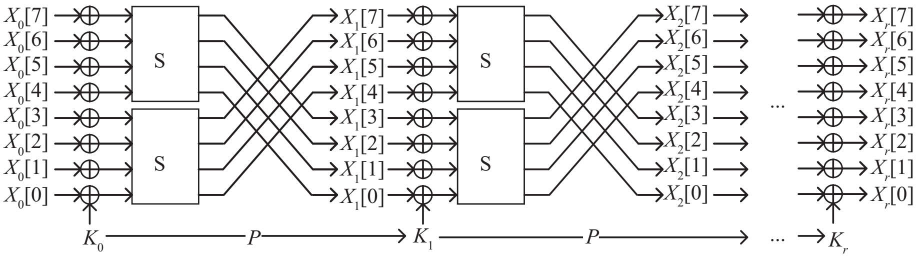

Figure 5 Cipher8: a toy block cipher with an SPN structure.

and 9 rounds at the end. The key-guessing set K_0 totally contains 232 key bits. However, $\mathrm{AKI}_{K_0} =$ 124 bits, which reduces the time complexity and makes the attack feasible.

3.1 An illustrative example: Cipher8

We constructed a toy cipher named "Cipher8" to help our readers better understand the above definitions[1]. Besides, we use three different key schedules to show different levels of interaction between the key schedule and the round function.

Assume that Cipher8 has an 8-bit block size with an SPN structure (see Figure 5). Every internal state X_i is XORed by the 8-bit subkey k_i. That is, the subkey is used as the round key. Each 8-bit subkey are generated iteratively by doing simple permutation P on the 8-bit master key k_0, i.e., $k_i = P(k_{i-1})$. Here, we consider three different key schedules with different permutations (see cases 1–3 in Figure 6). Next, we calculate the AKI by taking a 2-round encryption path as an example, as is depicted in Figure 6 (left part).

Obtaining the key-guessing set. In order to partially encrypt $X_2[7]$[2] in a 2-round path, we need 8 key bits: $K_0 = \{k_0[4–7], k_1[0–3]\}$. We can analyze it by using a backtracking method. In order to calculate $X_2[7]$, we need to know $X_1[0–3]$ and guess $k_1[0–3]$. In order to calculate $X_1[0–3]$, we need to know $X_0[4–7]$ and guess $k_0[4–7]$. Given all this, we get an 8-bit key-guessing set K_0[3], as is shown in Figure 6.

[1] Note that the structure of Cipher8 is totally insecure and it is just used for explaining and illustrating the above definitions.

[2] $X_i[j]$: the jth bit of X_i, where the right most bit is referred to as the zeroth bit.

[3] In [14], $X_2[7] \to X_1[0–3] \to X_0[4–7]$ is defined as a calculation dependency path. $k_0[4–7]$, $k_1[0–3]$ is defined as a key dependency path.

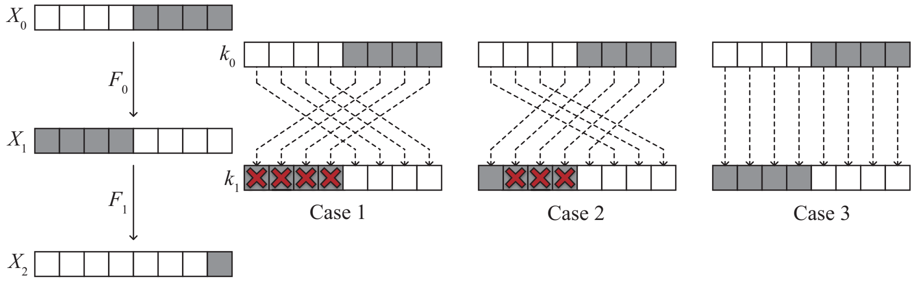

Figure 6 (Color online) The reduced sets of key-guessing set in three cases.

Calculating AKI. The AKI of the above key-guessing set varies with the key schedule candidate. The first case serves as a classical and also extreme case in terms of key bits leakage, where the diffusion of the key schedule and the diffusion of the round function are overlapped. More specifically, in the key-guessing set $\{K_0,\ k_1[0],\ k_1[1],\ k_1[2],\ k_1[3]\}$ can be deduced by $\{k_0[4],\ k_0[5],\ k_0[6],\ k_0[7]\}$. Therefore, we have that $\{k_0[4],\ k_0[5],\ k_0[6],\ k_0[7]\}\Rightarrow K_0$ and $\mathrm{AKI}_{K_0}=4$ bits (see Case 1 in Figure 6). For the second case, $\mathrm{AKI}_{K_0}=5$ bits (see Case 2 in Figure 6). For the third case, $\mathrm{AKI}_{K_0}=8$ bits (see Case 3 in Figure 6), which means that the whole key space is covered.

Through the same volume of the diffusion operations, severe key bits leakage can be caused while others are not, indicating that both the positions and the amount of the diffused bits would influence the actual key information.

3.2 The algorithm to calculate AKI

In 2014, Huang developed an effective and efficient tool for iterated key schedules of lightweight ciphers to search encryption/decryption paths involving insufficient AKI. We briefly review the main idea here and refer our readers to [15] for more details.

The search can be divided into two phases: obtaining the key-guessing set and calculating AKI. More specifically, by using the incidence matrix of the round function, we trace back round-by-round from one-bit active state on the r-th round to the plaintext to find all the related bits that have to be guessed in the intermediate states, which is defined as r-round forward paths. Then we can get all the subkey bits involved on the path, i.e., the key-guessing set K_0, according to the corresponding key schedule. Next, we search the possible reduced set of K_0 by using a greedy strategy to get AKI_{K_0}.

Remark 1. All r-round paths we searched start from one bit[4]. Suppose that the block size is n bits, then given a fixed r, we need to search n r-round paths to find the minimum AKI. The search covers different rounds until the AKI is larger than or equal to the master key length m.

Remark 2. All r-round paths we searched are forward, namely partial encryption path. Similar work can be done for partial decryption, which is omitted in our paper.

4 New observation on RECTANGLE key schedule

In this section, we search all 64 r-round $(r=1,2,3,\ldots)$ forward paths of RECTANGLE and calculate the least AKI involved by using Huang et al.'s tool [15]. We summarize the results in Table 1. Based on our observation of key bits leakage, we give a generic meet-in-the-middle attack on 12-round RECTANGLE-128 with quite low data complexity. Moreover, we analyze several variants of RECTANGLE.

4.1 Insufficient AKI for RECTANGLE

For RECTANGLE-80, there exists key bits leakage in 2, 3, 4-round forward paths. For RECTANGLE-128, there exists key bits leakage in r-round forward paths, where $2\leqslant r\leqslant 6$. See Table 1. We summarize the ratio of least AKI to its theoretical value on paths of different rounds for RECTANGLE in Figure 7.

[4] The reason why we do it this way is shown in [15] and omitted in our paper.

Table 1 Comparison between the least AKI and its theoretical value TKI on different rounds of forward paths for RECTANGLE-80, PRESENT-80, RECTANGLE-128 and PRESENT-128 (Unit: bit)a)

	RECTANGLE-80		PRESENT-80		RECTANGLE-128		PRESENT-128	
	TKI	AKI	TKI	AKI	TKI	AKI	TKI	AKI
$r = 1$	4	4	4	4	4	4	4	4
$r = 2$	20	**18**	20	20	20	**18**	20	20
$r = 3$	56	**42**	80	**64**	56	**44**	84	**77**
$r = 4$	80	**73**	80	80	120	**83**	128	**125**
$r = 5$	80	80	–	–	128	**103**	128	128
$r = 6$	–	–	–	–	128	**120**	–	–
$r = 7$	–	–	–	–	128	128	–	–

a) The bold numbers denote the insufficient AKI. Our search aborts when the AKI covers the whole key space.

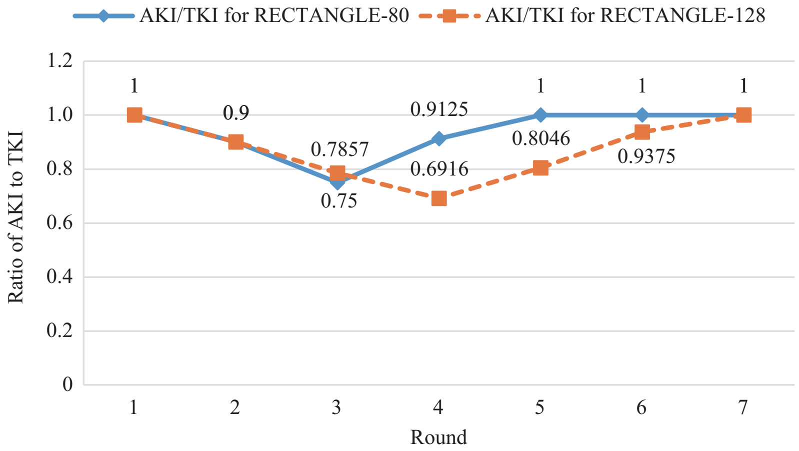

Figure 7 (Color online) The ratio of AKI to its theoretical value (on the y-axis) on paths of different rounds (on the x-axis) for RECTANGLE.

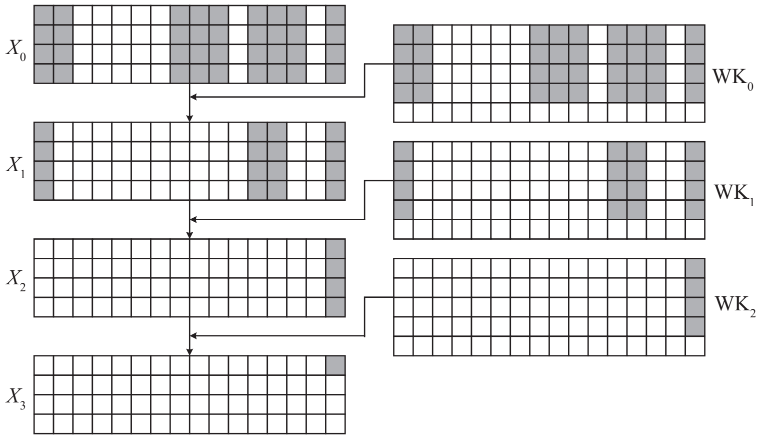

Figure 8 A 3-round path and the corresponding key-guessing set of RECTANGLE-80.

Example 2. For a 3-round path of RECTANGLE-80 from the plaintext X_0 to the state $X_3[0]$ (see Figure 8), the key-guessing set involves 56 key bits: $k_0[0, 2–4, 6–8, 14–16, 18–20, 22–24, 30–31, 34–36, 38–40, 46–48, 50–52, 54–56, 62, 63]$, $k_1[0, 3, 4, 15, 16, 19, 20, 31, 32, 35, 36, 47, 48, 51, 52, 63]$, $k_2[0, 16, 32, 48]$. However, the AKI is 42 bits, where $k_1[0, 15, 16, 19, 20, 31, 32, 35, 36, 47]$, $k_2[0, 16, 32, 48]$ are redundant.

When independently evaluating the diffusion of the key schedule, the master key bits achieve complete diffusion in 2 consecutive rounds for RECTANGLE-80 and 4 consecutive rounds for RECTANGLE-128, while the interaction between the diffusion of the key schedule and the diffusion of the round function results in AKI insufficiency in 4 consecutive rounds for RECTANGLE-80 and 6 consecutive rounds for

RECTANGLE-128, respectively. As is shown in [18], the 80-bit master key would determine the union of the subkey bits of any consecutive 2 rounds, while the 128-bit master key would determine the union of the subkey bits of the consecutive 4 rounds. Such diffusion of key bits are evaluated by considering the diffusion path of the key schedule in an isolated way, independent from the round function. However, in many existing attacks, it is the key bits distributed in the diffusion path of the round function that really matters. Both the positions and the amount of the diffused bits can have an impact on the actual key information. When considering the interaction between the diffusion of key schedule and the diffusion of round function, our results show that the key bits in 2, 3, 4-round paths do not cover the whole key space for RECTANGLE-80. In order to make sure that the key bits in each path depend on each of the 80 bits of the master key, the number of rounds should be at least 5. Similarly for RECTANGLE-128, our results show that there exists key bits leakage for 2, 3, 4, 5, 6-round paths. The number of rounds to make the key bits on the paths cover the whole 128-bit key space is at least 7. These three more rounds penalty benefits from the key bits leakage caused by the overlap between the round functions' diffusion and the key schedules' diffusion, which again demonstrates to us that their interaction should be attached with more attention.

As a comparison, we also put the corresponding results of AKI for PRESENT [1] in Table 1. In the design document of RECTANGLE [18], the authors has always made a comparison with PRESENT in many aspects when evaluating the security of RECTANGLE. Here, in terms of AKI, we can see that both RECTANGLE and PRESENT have varying degrees of key bits leakage, but the key-guessing sets of RECTANGLE involve less AKI than that of PRESENT.

4.2 A meet-in-the-middle attack on 12-round reduced RECTANGLE-128

Based on the aforementioned weakness of the key schedule, a generic meet-in-the-middle attack [13] on the 12-round reduced RECTANGLE-128 is given, which is composed by 12 rounds of transformation and a final AddRoundkey. As for the data complexity of the attack, there are only 8 known plaintexts. Besides, the time complexity and the memory complexity are $2^{126.32}$ encryptions and less than 2^{125} 128-bit blocks, respectively. Even though the attack does not serve as the best attack on RECTANGLE-128 in terms of the number of rounds, it is the first attack on RECTANGLE-128 in the type of meet-in-the-middle attack, with quite low data complexity. Truly, as illustrated in [36], it is of great importance to confirm the number of rounds which can be attacked through a few accessible data, for the better understanding about the security.

We set the intermediate value V as a 4-bit value in the 6-th round, i.e., $X_6[0,4,8,12]$, which can be computed in two different ways. In the encryption direction, V can be computed using only the plaintexts X_0 and the set K_t of key bits, and in the decryption direction, V can be computed using only the ciphertexts X_{12} and the set K_b of key bits. The forward and backward path and the key-guessing set K_t and K_b are depicted in Figure 9.

For the 6-round forward path, there are 288 bits of round key in the key-guessing set, and for the 6-round backward path, there are 292 bits of round key in the key-guessing set, i.e., $|K_t| = 288$ bits and $|K_b| = 292$ bits. That is to say, in order to calculate $X_6[0,4,8,12]$, we need to guess 288 key bits during the first 6-round encryption and 292 key bits during the last 6-round decryption, which should not have been feasible for RECTANGLE-128. However, the AKI is 124 bits on both forward path and backward path. Then, by exploiting such weakness of the key schedule, 124 key bits are enough to derive all the subkey bits in K_t or in K_b, which makes it possible to mount a MITM attack on 12-round RECTANGLE-128. For convenience, we denote the minimal reduced set of K_t (respectively K_b) by K'_t (respectively K'_b).

In the attack, for each guess of K'_t, which deduces K_t, we compute the value of V and stores it in a hash table. Then, for each guess of the subkey K'_b, which deduces K_b, we compute the value of V and search for a match in the Hash table. If the computation in the two different ways leads to the same value of V (which is always the case for the correct guess of (K_t, K_b)), then there is a match in the hash table and the corresponding key guess is regarded as a candidate key guess. Note that according to the

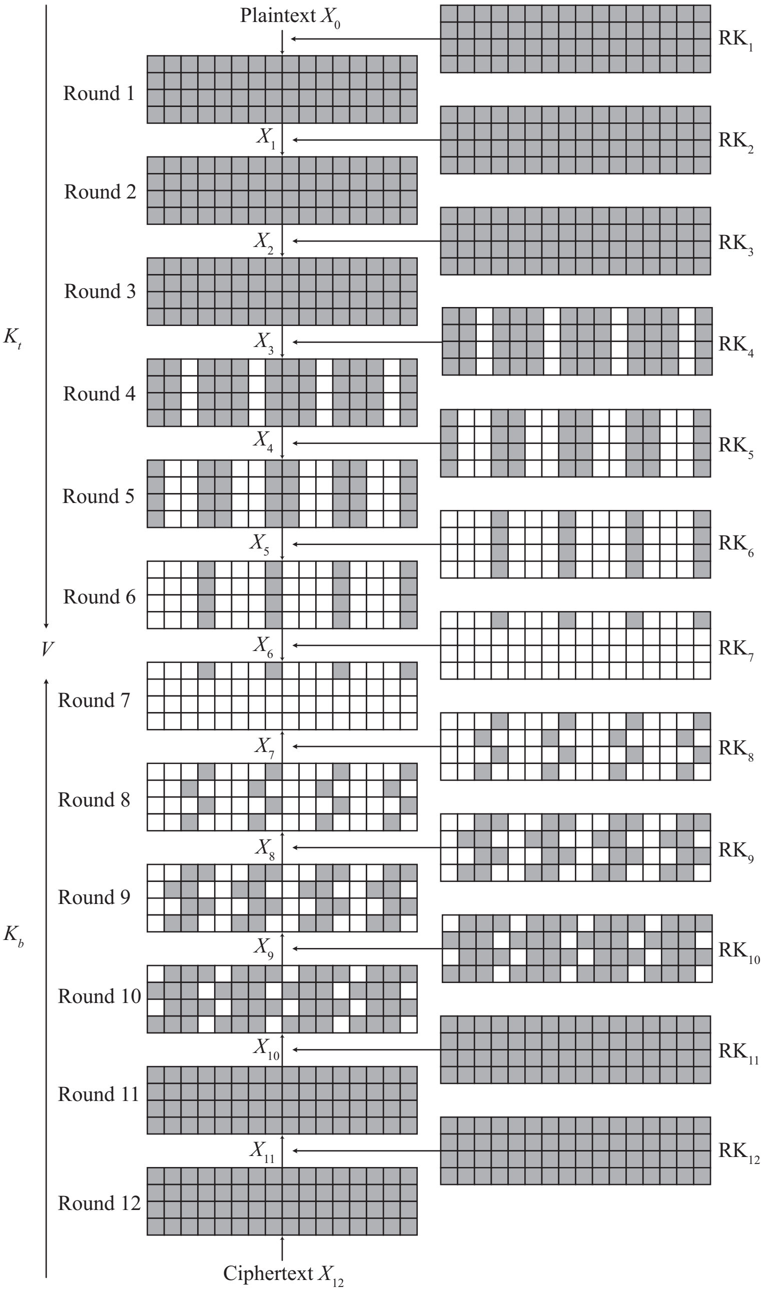

Figure 9 A 6-round forward path and backward path (left) and the corresponding key-guessing set (right) for RECTANGLE-128.

key schedule, after guessing all 124 key bits in K'_t, 92 key bits in K'_b can be known. Thus, these two paths share 92 bits of key information, and we can filter 92 bits of information from the key and reduce the time complexity in exhaustive search phase. The pseudo-code of the attack algorithm is given in Algorithm 1.

Complexity analysis. In the 12-round MITM attack, we use 8 known plaintext-ciphertext pairs. The time complexity is $\mathcal{O}(2^{124})$. More specifically, in the MITM phase, it is $2^{124} \cdot 8 \cdot 0.5 = 2^{126}$ 12-round encryptions. Meanwhile, the volume of the left candidate keys is $2^{124+124-92-4\cdot 8} = 2^{124}$. Thus, 2^{124} trivial encryptions are required in the exhaustive search phase. The entire time complexity is about $2^{126.32}$ encryptions of 12-round RECTANGLE-128. The memory complexity is $\mathcal{O}(2^{124})$. More specifically, the memory complexity is 2^{124} blocks of size $4 \cdot 8 + 124 = 156$ bits, which is less than 2^{125} 128-bit blocks.

Table 2 Comparison between the least AKI and its theoretical value TKI on different rounds of forward paths for RECTANGLE-128 and its three variants, PRESENT-128 and its variant without key schedule[a)b)]

	RECTANGLE-128		T_1-128	T_2-128	T_3-128	PRESENT-128		P-variant
	TKI	AKI	AKI	AKI	AKI	TKI	AKI	AKI
$r = 1$	4	4	4	4	4	4	4	4
$r = 2$	20	**18**	20	20	20	20	20	20
$r = 3$	56	**44**	**52**	**52**	**54**	84	**77**	**80**
$r = 4$	120	**83**	**68**	**100**	**104**	128	**125**	128
$r = 5$	128	**103**	**80**	128	128	128	128	–
$r = 6$	128	**120**	**92**	–	–	–	–	–
$r = 7$	128	128	**104**	–	–	–	–	–
$r = 8$	128	–	**116**	–	–	–	–	–
$r = 9$	128	–	**125**	–	–	–	–	–
$r = 10$	128	–	128	–	–	–	–	–

a) T_1-128 is RECTANGLE-128 variant with PRESENT-128 key schedule, T_2-128 is RECTANGLE-128 variant without key schedule, T_3-128 is RECTANGLE-128 variant with our new key schedule proposal.

b) The bold numbers denote the insufficient AKI. Our search aborts when the AKI covers the whole key space.

Algorithm 1 Meet-in-the-middle attack on 12-round RECTANGLE-128

```
for each possible value of 124 bits for $K'_t$ do
    Deduce the corresponding subkey bits in $K_t$;
    Encrypt $P_i$ to get $X_6^i[0,4,8,12]$, $i = 1, \ldots, 8$;
    Store corresponding key values for $K'_t$ in $T_1$ indexed by $X_6^1[0,4,8,12]||\ldots||X_6^8[0,4,8,12]$;
end for
for each possible value of 124 bits for $K'_b$ do
    Deduce the corresponding subkey bits in $K_b$;
    Decrypt $C_i$ to get $X_6^i[0,4,8,12]$, $i = 1, \ldots, 9$;
    if this $X_6^1[0,4,8,12]||\ldots||X_6^8[0,4,8,12]$ exists in $T_1$ then
        Take the corresponding key values for $K'_t$ as well as current $K'_b$ value as candidate key materials;
    end if
end for
Exhaustively search each remaining candidate key.
```

4.3 Experiments on variants of RECTANGLE key schedule

Nowadays, some trends exist in terms of the lightweight cryptography such as the increasing popularity of "lighter" key schedules. Some lightweight block ciphers, such as LED [2], do not even have the key schedule. They just directly apply master keys in each round. More specifically, in the case that the key size is equal to the block size, the master key is used in each round as the round key. In the case that the key size double the block size, the first half of the master key and the second half are used repeatedly in each round. A natural question is that, how AKI changes if RECTANGLE does away with the key schedule? And since the paths of PRESENT involve more AKI than that of RECTANGLE, what if RECTANGLE uses PRESENT key schedule?

We conduct extensive experiments on these two types of RECTANGLE-128 variants, where we keep the cipher algorithm but change the key schedule. We denote by T_1-128 the first type of RECTANGLE variant with PRESENT-128 key schedule, and by T_2-128 the second type of RECTANGLE variant which uses the master key directly. Incidentally, we also change the key schedule of PRESENT-128 by using the master key directly, which we denote by P-variant. Following the line of the above experiments, we search all 64 forward paths of different rounds for T_1-128 and T_2-128 and calculate the least AKI of the corresponding key-guessing set, as shown in Table 2.

Surprisingly, we find that RECTANGLE-128 without key schedule behaves better from the perspective of AKI. We get similar results for PRESENT-128 variant without key schedule (see Table 2). This issue indicates to some extent that a more complicated key schedule does not always means more security, which is a little counterintuitive.

For T_1-128, when replacing the key schedule of RECTANGLE-128 with that of PRESENT-128, there

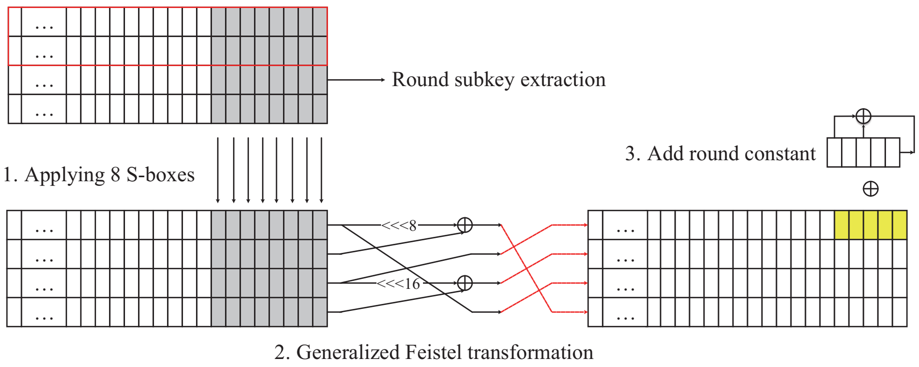

Figure 10 (Color online) A new key schedule proposal for RECTANGLE-128.

Row_0	→	Row_0
Row_1	→	$(\mathrm{Row}_2 \lll 16) \oplus \mathrm{Row}_3$
Row_2	→	Row_2
Row_3	→	$(\mathrm{Row}_0 \lll 8) \oplus \mathrm{Row}_1$

$$\xrightarrow{\lll \text{ one row}}$$

Row_0	→	$(\mathrm{Row}_2 \lll 16) \oplus \mathrm{Row}_3$
Row_1	→	Row_2
Row_2	→	$(\mathrm{Row}_0 \lll 8) \oplus \mathrm{Row}_1$
Row_3	→	Row_0

Figure 11 (Color online) The generalized Feistel transformation of the original RECTANGLE-128 key schedule (left) and our new key schedule proposal (right).

exists key bits leakage for more rounds. This reminds us, once more, that the interaction between the diffusion of round functions and the diffusion of key schedules should get more attention.

5 A new key schedule proposal for RECTANGLE-128

Inspired by T_2-128, a new key schedule for RECTANGLE-128 is proposed (see Figure 10). Our proposal makes a slight modification to the original one:

(1) We slightly modify the generalized Feistel transformation step (see Figure 11);

(2) We extract the 64-bit round key from the first 2 rows instead of the 16 rightmost columns of the current state of the key register.

After making these modifications to the key schedule of RECTANGLE-128, the two 64-bit parts of the 128-bit master key can be used alternately in each round. Our results show that RECTANGLE-128 with the modified key schedule, which is denoted by T_3-128, involves more actual key information than both RECTANGLE-128 and T_2-128 for the same round (see Table 2). Our results remind us that, besides the interaction between the key schedule's diffusion and the round function's diffusion, the key extraction phase of the key schedule should be given more attention.

We emphasize that our results mainly focus on resisting attacks resulting from key information leakage. Other security considerations should be taken into account when designing the key schedule of a lightweight block cipher, such as the related-key attacks (which is not taken in consideration in this paper and left as our future work), as the key schedule impacts many types of attacks.

6 Conclusion and further work

In this paper, we analyzed the key schedule of RECTANGLE from the perspective of actual key information. We pointed out that the key schedule of RECTANGLE poorly distribute key bits in the diffusion path of the round function, leading to AKI insufficiency of different rounds. Compared with PRESENT, the information leakage of key bits for RECTANGLE is more serious. Moreover, we analyze several variants of RECTANGLE and PRESENT. Surprisingly we find that both PRESENT-128 and RECTANGLE-128 without key schedule involve more AKI than the original one. Inspired by the experiment results, we propose a new key schedule for RECTANGLE-128 by making a slight modification to the original one. Actually, AKI reveals the interaction between key schedules and round functions, not merely the property of key schedules. We hope our work add more insight to the design of (lightweight) block ciphers.

Our future work will be done in the following directions. One is developing effective automatic tool based on graph theory to get the exact value of AKI (basically completed). Huang et al. [15] developed an efficient tool to search the computation chains involving insufficient AKI for iterated key schedules of lightweight ciphers, which has been used in our paper. However, what they obtained is in fact an upper bound of real AKI. The algorithm for searching key-bridging technique given by Lin et al. [16] can also be seen as an tool to calculate AKI, as the existence of key bridges is actually key bits leakage. But they did not consider the key-guessing set obtaining phase in their work. Complete and effective automatic tool in needed. Meanwhile, it is significant to think about the design criteria of practical guiding significance for avoiding AKI insufficiency, not just in a “detecting-then-modifying” way. Another work is to further investigate the new key schedule proposals will/will not affect the related-key attack resistance (in progress). It is interesting to evaluate and redesign RECTANGLE variant without key schedule, since our experiments show that RECTANGLE-128 without key schedule involves more AKI than the original version. In this case, the design will be carried out in a much more compact way by concentrated only on the round function part, no key schedule and therefore no need to consider the interaction between key schedules and round functions.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61702331, 61472251, 61402280, U1536101), China Postdoctoral Science Foundation (Grant No. 2017M621471), National Cryptography Development Fund (Grant No. MMJJ20170105), and Science and Technology on Communication Security Laboratory.

References

1 Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2007. 450–466

2 Guo J, Peyrin T, Poschmann A. The LED block cipher. In: Proceedings of Cryptographic Hardware and Embedded Systems-CHES 2011, 2011

3 Banik S, Bogdanov A, Isobe T, et al. Midori: a block cipher for low energy. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2014. 411–436

4 Beierle C, Jean J, Kölbl S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Proceedings of Annual Cryptology Conference. Berlin: Springer, 2016. 123–153

5 Avanzi R. The QARMA block cipher family. IACR Trans Symmetric Cryptol, 2017, 2017: 4–44

6 Banik S, Pandey S K, Peyrin T, et al. GIFT: a small PRESENT. In: Proceedings of International Conference on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2017. 321–345

7 Beaulieu R, Treatman-Clark S, Shors D, et al. The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015. 1–6

8 Wu W, Zhang L. LBlock: a lightweight block cipher. In: Proceedings of International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2011. 327–344

9 Suzaki T, Minematsu K, Morioka S, et al. TWINE: a lightweight block cipher for multiple platforms. In: Proceedings of International Conference on Selected Areas in Cryptography. Berlin: Springer, 2012. 339–354

10 Hong D, Sung J, Hong S, et al. HIGHT: a new block cipher suitable for low-resource device. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2006. 46–59

11 Needham R M, Wheeler D J. Tea Extensions. Technical Report. Cambridge: Cambridge University, 1997

12 Knudsen L, Leander G, Poschmann A, et al. PRINTcipher: a block cipher for IC-printing. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2010. 16–32

13 Diffie W, Hellman M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard. Computer, 1977, 10: 74–84

14 Huang J, Lai X. Revisiting key schedule’s diffusion in relation with round function’s diffusion. Des Codes Cryptogr, 2014, 73: 85–103

15 Huang J, Vaudenay S, Lai X. On the key schedule of lightweight block ciphers. In: Proceedings of International Conference in Cryptology in India. Berlin: Springer, 2014. 124–142

16 Lin L, Wu W, Zheng Y. Automatic search for key-bridging technique: applications to LBlock and TWINE. In: Proceedings of International Conference on Fast Software Encryption. Berlin: Springer, 2016. 247–267

17 Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round AES-192 and AES-256. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2010. 158–176

18 Zhang W T, Bao Z Z, Lin D D, et al. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci China Inf Sci, 2015, 58: 1–15

19 Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptology, 1991, 4: 3–72

20 Biham E, Biryukov A, Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999. 12–23

21 Matsui M. Linear cryptanalysis method for DES cipher. In: Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques. Berlin: Springer, 1993. 386–397

22 Daemen J, Knudsen L, Rijmen V. The block cipher Square. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 1997. 149–165

23 Knudsen L, Wagner D. Integral cryptanalysis. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2002. 112–127

24 Collard B, Standaert F X. A statistical saturation attack against the block cipher PRESENT. In: Proceedings of Cryptographers' Track at the RSA Conference. Berlin: Springer, 2009. 195–210

25 Shan J Y, Hu L, Song L, et al. Related-key differential attack on 19-round reduced RECTANGLE-80 (in Chinese). J Cryptologic Reseatch, 2015, 2: 54–65

26 Kosuge H, Tanaka H, Iwai K, et al. Integral attack on reduced-round Rectangle. In: Proceedings of IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), 2015. 68–73

27 Sun L, Wang M Q. Toward a further understanding of bit-based division property. Sci China Inf Sci, 2017, 60: 128101

28 Xiang Z, Zhang W, Bao Z, et al. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2016. 648–678

29 Sasaki Y, Todo Y. New impossible differential search tool from design and cryptanalysis aspects. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2017. 185–215

30 Zhang W, Bao Z, Rijmen V, et al. A new classification of 4-bit optimal s-boxes and its application to PRESENT, RECTANGLE and SPONGENT. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2015. 494–515

31 Stoffelen K. Optimizing s-box implementations for several criteria using SAT solvers. In: Proceedings of International Conference on Fast Software Encryption. Berlin: Springer, 2016. 140–160

32 Bao Z, Luo P, Lin D. Bitsliced implementations of the PRINCE, LED and RECTANGLE block ciphers on AVR 8-bit microcontrollers. In: Proceedings of International Conference on Information and Communications Security. Berlin: Springer, 2015. 18–36

33 Maene P, Verbauwhede I. Single-cycle implementations of block ciphers. In: Proceedings of International Workshop on Lightweight Cryptography for Security and Privacy. Berlin: Springer, 2015. 131–147

34 Feizi S, Nemati A, Ahmadi A, et al. A high-speed FPGA implementation of a Bit-slice Ultra-Lightweight block cipher, RECTANGLE. In: Proceedings of the 5th International Conference on Computer and Knowledge Engineering (ICCKE), 2015. 206–211

35 Biryukov A, Derbez P, Perrin L. Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2015. 3–27

36 Bouillaguet C, Derbez P, Fouque P A. Automatic search of attacks on round-reduced AES and applications. In: Proceedings of Annual Cryptology Conference. Berlin: Springer, 2011. 169–187