Information Processing Letters 116 (2016) 259–266

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Linear cryptanalysis of reduced-round SPECK

Yu Liu, Kai Fu, Wei Wang, Ling Sun, Meiqin Wang *

Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China

ARTICLE INFO

Article history:
Received 15 September 2015
Accepted 5 November 2015
Available online 2 December 2015
Communicated by S.M. Yiu

Keywords:
Lightweight block cipher
SPECK
Linear approximation
Linear cryptanalysis
Cryptography

ABSTRACT

SPECK is a family of lightweight block ciphers which was proposed by United States National Security Agency and designed for optimal performance in software. The paper gives the security of SPECK against linear cryptanalysis and introduces 9, 10, 12, 15 and 16 rounds linear approximations on SPECK for block sizes of 32, 48, 64, 96 and 128 bits, respectively. Partial linear mask table is used to speed up the search progress rather than the linear mask table. Using the structure of red-black tree to store the pLMT, we deduce the search time. Combining the Segment Searching with branch-and-bound method, the search time is further reduced. For 48-, 96- and 128-bit version the lengths of the linear approximations are 1, 9 and 10 rounds longer than the previous linear cryptanalytic. For SPECK64 the correlation of the linear approximation is twice as much as the previous linear cryptanalytic. As a result, we improve the previous linear cryptanalysis and gain more obvious advantage for block lengths of 96 and 128 bits. Especially, in aspect of SPECK96/144, SPECK128/192 and SPECK128/256 we can attack the same rounds as the best previous attacks.

1. Introduction

In June 2013 the United States National Security Agency published the SPECK family of lightweight block ciphers, which was optimized for software implementations [2]. Due to SPECK's wide application prospect, it's necessary to evaluate the security against various attacks and there are numerous cryptanalyses of SPECK, such as differential cryptanalysis [1,3,5], rectangle attack [1], differential fault analysis [13] and linear cryptanalysis [15]. Among them, the best attacks on round-reduced SPECK are differential attacks presented by Abed et al. [1] and Dinur [5], while for the linear cryptanalysis, recently the only results were proposed by Yao et al. [15]. They applied Wallén's numeration algorithm [14] to Matsui's branch-and-bound framework [9] in the searching algorithm for linear trails, however, there is still a gap compared with the differential cryptanalysis. Especially for 96- and 128-bit version. Therefore, we explore the security of SPECK against linear cryptanalysis, and match the same rounds with the differential attacks for 96- and 128-bit version. Summary of the attacks is presented in Table 2.

Linear cryptanalysis is a known-plaintext attack which was introduced by Matsui as a theoretical attack on the Data Encryption Standard (DES) [8] and later successfully led to a practical cryptanalysis of DES [7]. It exploits the correlation of linear approximations between input and output of a block cipher. Since SPECK is an ARX cipher, which consists of modular addition, bit rotation and XOR operations, the key step in searching the linear approximation is to calculate the correlation of linear approximation for modular addition. The linear approximations of addition modulo 2^n was investigated by Wallén [14], where an efficient algorithm to compute the probability of linear approximation of modular addition is presented. Nyberg and Wallén applied this algorithm to find the linear approximation of the FSM of the stream cipher SNOW 2.0 [10]. A more explicit formula for linear probabilities of addi-

* Corresponding author.
E-mail address: mqwang@sdu.edu.cn (M. Wang).

http://dx.doi.org/10.1016/j.ipl.2015.11.005
0020-0190/

tion modulo 2^n was given by Schulte-Geers [11]. Based on it, Dehnavi et al. exhibited [4] a better insight for these probabilities. We name their method State Conversion for convenience.

Our Contributions:

- **A new search method for linear approximations of the SPECK family.** In the search process for linear approximations we produce the Linear Mask Table (LMT) easily by State Conversion. However, the LMT is too large to search the linear approximations. Therefore we will use partial linear mask table (pLMT) rather than LMT, similarly to the partial differential distribution tables [3]. Moreover, since the pLMT is traversed many times, it will be stored using the structure of red-black tree. Then the time of traversing the pLMT will be $O(\log m)$, where m is the total number of elements in the tree. For the cipher with large block size, the search process is divided into several parts, and these parts will be joined together into the linear approximation. This method is called Segment Searching. Combining the methods above with [9] will show several approaches to produce linear approximations for SPECK family.
- **The best known linear approximations of the SPECK family.** We present the best known linear approximations for 9-round SPECK32 with correlation 2^{-14}, 10-round SPECK48 with correlation 2^{-22}, 12-round SPECK64 with correlation 2^{-30}, 15-round SPECK96 with correlation 2^{-45} and 16-round SPECK128 with correlation 2^{-61}. The different outcomes between [15] and this paper are illustrated in Table 1. Here is the only comparison of the maximum number of rounds and correlation of the linear approximations, because there are no specific linear approximations in [15].
- **Linear attack on the variants of SPECK96 and SPECK128.** Taking advantage of these linear approximations, the paper will present attacks on SPECK of 96 and 128 bits versions, which target up to the same rounds as the differential cryptanalysis. Moreover, it will improve the previous linear cryptanalysis significantly.

Table 1
Comparison of the linear approximations on SPECK.

Block size	Maximum number of rounds		Correlation	
	[15]	this paper	[15]	this paper
32	9	9	2^{-14}	2^{-14}
48	9	10	2^{-20}	2^{-22}
64	12	12	2^{-31}	2^{-30}
96	6	15	2^{-11}	2^{-45}
128	6	16	2^{-11}	2^{-61}

A complete summary of all the best previous attacks on SPECK is described in [1,5,15] as seen in Table 2. All of the best previous attacks are based on differential cryptanalysis and related techniques.

The paper is organized as follows: Section 2 will list the notations used throughout this paper and present a brief description of SPECK family. Sections 3 and 4 present the linear approximations and the linear attacks on SPECK, respectively. Finally, we conclude the paper in Section 5.

2. Preliminaries

This section gives notations and presents a brief description of SPECK.

2.1. Notations

The following notations are used in this paper. For $a = (a[n-1], \ldots, a[0]) \in \mathbb{F}_2^n$, $b = (b[n-1], \ldots, b[0]) \in \mathbb{F}_2^n$, $c = (c[n-1], \ldots, c[0]) \in \mathbb{F}_2^n$:

$a \oplus b$	bitwise XOR of a and b,
$w(a)$	hamming weight of the binary vector a,
$\lll i$	left circular shift by i bits,
$\ggg j$	right circular shift by j bits,
$a[i]$	the i-th bit of a, $i \in \{0, \ldots, n-1\}$,
$a[i]b[j]$	bitwise AND on $a[i]$ and $b[j]$, $i,\ j \in \{0, \ldots, n-1\}$,
$a \cdot b$	the inner product of a and b, $a \cdot b = \bigoplus_{i=0}^{n-1} a[i]b[i]$,
$a\&b$	bitwise AND of a and b, $a\&b = (a[n-1]b[n-1], \ldots, a[0]b[0])$,
$a \boxplus b \ (mod\ m)$	a add to b modulo m, $m \in \mathbb{Z}^+$,

Table 2
Summary of attacks on SPECK family.

SPECK2n/k	Rounds attacked/Total rounds	Method	Time (en)	Data	Memory (block)	Reference
96/144	16/29	RC	$2^{135.9}$	$2^{90.9}$CP	$2^{90.92}$	[1]
	17/29	DC	2^{133}	2^{85}CP	$2^{18.42}$	[5]
	9/29	LC	$2^{122.90}$	$2^{27.65}$KP	–	[15]
	17/29	**LC**	2^{96}	2^{92}**KP**	2^{92}	**This paper**
128/192	18/33	RC	$2^{182.7}$	$2^{125.9}$CP	$2^{117.9}$	[1]
	18/33	DC	2^{177}	2^{113}CP	2^{18}	[5]
	9/33	LC	$2^{156.74}$	$2^{28.30}$KP	–	[15]
	18/33	**LC**	2^{128}	2^{124}**KP**	2^{124}	**This paper**
128/256	18/34	RC	$2^{182.7}$	$2^{125.9}$CP	$2^{117.9}$	[1]
	19/34	DC	2^{241}	2^{113}CP	2^{18}	[5]
	7/34	LC	$2^{220.74}$	$2^{28.30}$KP	–	[15]
	19/34	**LC**	2^{192}	2^{124}**KP**	2^{124}	**This paper**

DC: Differential Cryptanalysis. LC: Linear Cryptanalysis. RC: Rectangle Cryptanalysis. CP: Chosen Plaintexts. KP: Known Plaintexts.

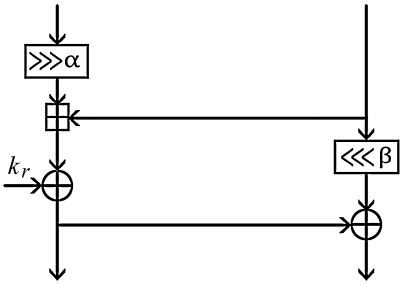

Fig. 1. SPECK round function.

Table 3
SPECK parameters.

Block size	Key size	n	k	a	b	T
32	64	16	4	7	2	22
48	72	24	3	8	3	22
	96		4			23
64	96	32	3	8	3	26
	128		4			27
96	96	48	2	8	3	28
	144		3			29
128	128	64	2	8	3	32
	192		3			33
	256		4			34

the sign	$a[i]+2b[i]+4c[i],\ i\in\{0,\dots,n-1\}$,
0-block	a block of sign 0,
7-block	a block of sign 7,
o-block	a block of signs 1, 2, 4,
e-block	a block of signs 3, 5, 6,
block $B[i]$	0-block, 7-block, o-block or e-block,
$\|B\|$	the number of signs in block B,
$a\|\|b$	concatenation of a and b,
$cor_{\Lambda_\alpha,\Lambda_\beta\Rightarrow\Lambda_\gamma}$	the correlation of $(\Lambda_\alpha, \Lambda_\beta \Rightarrow \Lambda_\gamma)$, where Λ_α, Λ_β and Λ_γ are the two input masks and output mask of modular addition, respectively.

2.2. Brief description of SPECK

The SPECK2n/k family is $2n$-bit block cipher with key length of k, which is a simple ARX-based Feistel network [2]. The input of the cipher is split up into two words L_0 and R_0. (The left word is rotated to the right by a bits. Then it will be modulo 2^n added to the right word and XOR to the round key. The right word will then be rotated to the left by b bits. At last the left word is XORed to the right.) For r from 1 to T, where T is the total number of rounds, the following is carried out:

$$L_r = (L_{r-1} \ggg \alpha) \boxplus R_{r-1} \oplus k_r,$$

$$R_r = (R_{r-1} \lll \beta) \oplus L_r.$$

The round function is depicted in Fig. 1. The values of n, k, a, b, T are shown in Table 3. See [2] for more details.

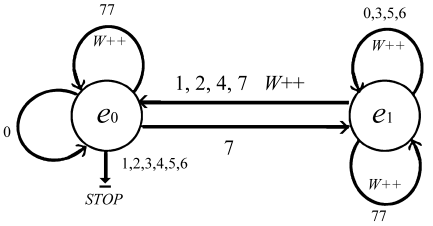

Fig. 2. State conversion.

3. Linear approximation of SPECK

3.1. Correlation of addition modulo 2^n

Let $G:\ \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$, $a=(a[n-1],\dots,a[0])$ and $b=(b[n-1],\dots,b[0])$ denote the input masks, $c=(c[n-1],\dots,c[0])$ denotes the output mask. The correlation is

$$\begin{aligned} cor_G(c,a,b) &= cor(c\cdot G(x_1,x_2)\oplus a\cdot x_1\oplus b\cdot x_2)\\ &= 2^{-2n}(\#\{x_1,x_2\in\mathbb{F}_2^n : c\cdot G(x_1,x_2)\\ &\quad \oplus a\cdot x_1\oplus b\cdot x_2=0\}-\#\{x_1,x_2\in\mathbb{F}_2^n :\\ &\quad c\cdot G(x_1,x_2)\oplus a\cdot x_1\oplus b\cdot x_2=1\}). \end{aligned} \tag{1}$$

Define r-round approximation $\Omega=(\Gamma_0,\Gamma_1,\dots,\Gamma_r)$, where Γ_0 is the input mask. Γ_i is the output mask of round i and the input mask of round $i+1 (i=1,\dots,r)$. Suppose the correlation of round i is cor_i and the correlation of the approximation is cor_Ω. According to the piling-up lemma, there is $cor_\Omega=\prod_{i=1}^n cor_i$. It's obvious that the only non-linear part of SPECK is addition modulo 2^n. So we take advantage of the State Conversion [4,10,11,14] to compute cor_g, where $g:(x,y)\mapsto z=x\boxplus y (mod\ 2^n)$. We just recall this method briefly. Define

$$|cor_g(c,b,a)| = |2p(a\cdot x\oplus b\cdot y=c\cdot z)-1|, \tag{2}$$

where $p(A)$ means the probability of occurrence of A. Let $s[i]=a[i]\oplus b[i]\oplus c[i], z[n-1]=0,\ z[i-1]=z[i]\oplus s[i]$, $0\le i<n$. The correlation in Eq. (2) is non-zero, if and only if for all i with $z[i]=0$ the equality $a[i]=b[i]=c[i]$ holds. In this case the absolute value of cor_g is $2^{-w(z)}$. The sign of the correlation is -1 if and only if $w((a\oplus c)\&(b\oplus c))$ is odd. Denote $\zeta[i]=4c[i]+2a[i]+b[i], 0\le i<n$. The sign $\zeta[i]$ is in $\{0,\dots,7\}$. Then Eq. (2) could be computed by the automaton of Fig. 2.

The search process begins by state e_0 and traverses the figure from ζ_{n-1} to ζ_0. When a sign comes, the state will convert to another state or itself. If you meet "STOP" then Eq. (2) is equal to 0. Otherwise, Eq. (2) is equal to 2^{-W+1}. In [4,10], there are some examples.

Theorem 1. *(See [4, Theorem 1].) Let* $\zeta[n-1],\dots,\zeta[0]=B[m],\dots,B[1]$. *Put* $\alpha[m]=0$ *and for* $1\le i<m$

$$\lambda[i]=\begin{cases}1 & A \text{ is odd}\\ 0 & A \text{ is even},\end{cases}$$

where $A=\#\{B[j]: i<j\le m, B[j] \text{ is 7-block of odd length}\}+\#\{B[j]: i<j\le m, B[j] \text{ is o-block}\}$. *Then (2) equals to* $\frac{q}{2^W}$, *where*

$$q = \prod_{i=1}^{m}(1 - \bar{\lambda_i}[B[i] \text{ is o-block or e-block}]),$$

$$W = \sum_{B[i] \text{ is o-block or e-block}} |B[i]| + \sum_{B[i] \text{ is 7-block}} \lfloor \frac{|B[i]|}{2} \rfloor + \sum_{B[i] \text{ is o-block}} \lambda_i |B[i]|.$$

According to Theorem 1, it is easy to see that the state conversion after state e_1 will increase w by 1. Here we get some corollaries in the following.

Corollary 1. *The most significant sign should be* 0 *or* 7*, i.e.* $\zeta[n-1]$ *is* 0 *or* 7*. Otherwise Eq. (2) equals to* 0*.*

Corollary 2. *The first nonzero sign should be* 7 *from the most significant bit to the least significant bit, i.e. from* $\zeta[n-1]$ *to* $\zeta[0]$*. Otherwise Eq. (2) equals to* 0*.*

(Note that the signs are "big-endian".)

Corollary 3. *If the first nonzero sign is* 7*, suppose the sign is* $\zeta[i]$*,* w *should be added one no matter what value of* $\zeta[i-1]$ *is taken on.*
Case 1. If $\zeta[i-1] \in \{1, 2, 4, 7\}$*,* $\zeta[i-2]$ *should be* 0 *or* 7*. Otherwise Eq. (2) equals to* 0*.*
Case 2. If $\zeta[i-1] \in \{0, 3, 5, 6\}$*,* W *should be added one no matter what value of* $\zeta[i-2]$ *is taken on.*

Corollary 4. *If Eq. (2) is not zero, then the state after o-block is* e_0*.*

Using the above-mentioned information and the direct consequences of Theorem 1 in [4], the correlation of addition modulo 2^n can be computed quickly.

3.2. Partial linear mask table

Rather than the full pre-computed linear mask table (LMT), pLMT will be used which is similar to the partial pre-computed difference distribution table in [3].

Definition 1. A **partial linear mask table (pLMT)** T is a linear mask table that contains all linear masks $(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma)$ whose correlations are lager than or equal to a pre-defined threshold C, i.e.,

$$(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma) \in T \Leftrightarrow cor_{\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma} \geq C.$$

Proposition 1. *The size of LMT increases with the word size n of the linear mask* $(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma)$*.*

Proposition can be easily verified by Theorem 1. For instance, put $C = 2^{-4}$, if $n = 8$ the size of T is 364 KB, while if $n = 16$ the size of T is about 1 MB and $n = 24$ the size of T is about 153.6 MB. For the purpose of saving memory, the pLMT is divided into some small subtables in which all masks are 8-bit. The numeration of correlation starts from the sign's MSB and will have a final state. To link up two subtables, the final state of the previous table is the same as the initial state of the next table. There are two states, e_0 and e_1. Two subtables which have different initial states should be made. The pseudo-code in the appendix in [4] will then change into Algorithm 1.

Algorithm 1 Computation of a pLMT.

Require: $\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma; c$; initial state: S_{ini}
Ensure: pLMT T: $(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma) \in T, cor_{\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma} \geq C$; correlation: W; final state: S_{fia}.
compute the sign $\zeta[i] = \Lambda_\alpha[i] + 2\Lambda_\beta[i] + 4\Lambda_\gamma[i], (0 \leq i \leq 7)$
for $i = n-1$ to $i = 0$ **do**
  if $\zeta[i] == 7$ **then**
    $i = i - |7\text{-block}|$
    if $|7\text{-block}|$ is odd **then**
      State Transition
      if state is e_0 **then**
        $W = W \times 2^{-\lfloor |7\text{-block}|/2 \rfloor}$
      else
        $W = W \times 2^{-1-\lfloor |7\text{-block}|/2 \rfloor}$
      end if
    else
      $W = W \times 2^{-\lfloor |7\text{-block}|/2 \rfloor}$
    end if
  end if
  if $\zeta[i] == 0$ **then**
    $i = i - 1$
    if state is e_1 **then**
      $W = W \times \frac{1}{2}$
    end if
  end if
  if $\zeta[i] == 1,\ 2$ *or* 4 **then**
    $i = i - 1$
    if state is e_0 **then**
      STOP
    else
      $W = W \times \frac{1}{2}$ and State Transition
    end if
  end if
  if $\zeta[i] == 3,\ 5$ *or* 6 **then**
    $i = i - 1$
    if state is e_0 **then**
      STOP
    else
      $W = W \times \frac{1}{2}$
    end if
  end if
end for
if $W \geq C$ **then**
  Add $(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma), W, S_{fia}$ to T
  continue
end if

There are two tables named $pLMT_0$ with initial state e_0 and $pLMT_1$ with initial state e_1. Theorem 1 justifies Algorithm 1. The complexity of this algorithm is $\Theta(log\ n)$.

3.3. Application to SPECK

The key point of linear attack is to find a high-probability linear approximation covering as many rounds as possible for a block cipher, i.e. the most important thing is to guarantee each round with high correlation. Due to $cor_\Omega = \prod_{i=1}^{n} cor_i$, the correlation of each round will be high enough. Suppose the input mask and the output mask of round i are Γ_{i-1} and Γ_i, respectively. Let the mask of addition modulo 2^n be $(\Lambda_\alpha, \Lambda_\beta, \Lambda_\gamma)$ as shown in Fig. 3, the following equations can be deduced.

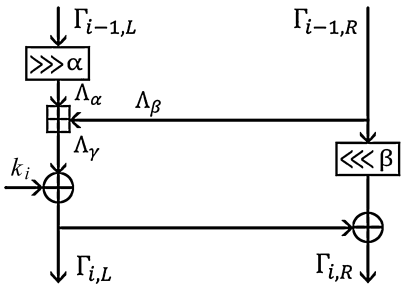

Fig. 3. Linear mask of the round function.

Algorithm 2 Search for linear approximation using $pLMT_0$ and $pLMT_1$.

Require: input mask Γ_0; correlation threshold C_{lc}; word size $n == 8m$; number of rounds r; current round *round*; correlation of current round W

Ensure: $LC \leftarrow \Gamma_0, \Gamma_1, \ldots, \Gamma_r$: trail for r-round with correlation $\hat{C}$

procedure search($\Gamma_{j,L}, \Gamma_{j,R}, round, W$)

$\Lambda_{\alpha j} \leftarrow \Gamma_{j,L} \ggg \alpha$

set $\Lambda_{\alpha j} = \Lambda_{\alpha j,m-1}||\Lambda_{\alpha j,m-2}||\ldots||\Lambda_{\alpha j,0}$, where $\Lambda_{\alpha j,i}$ is a byte.

find records indexed by $\Lambda_{\alpha j,m-1}$ from $pLMT_0$, denominate them $record_{j,m-1}$, otherwise continue

for these records **do**

  for $i = m-1$ to $i = 0$ **do**

    if $S_{fia} == 0$ in $record_{j,i+1}$ **then**

      find records indexed by $\Lambda_{\alpha j,i}$ from $pLMT_0$, denominate them $record_{j,i} \Leftrightarrow (\Lambda_{\alpha j,i}, \Lambda_{\beta j,i}, \Lambda_{\gamma j,i}, W_{j,i}, S_{fia})$, otherwise go to step 5

    else

      find records indexed by $\Lambda_{\alpha j,i}$ from $pLMT_1$, denominate them $record_{j,i}$, otherwise go to step 5

    end if

    $i = i - 1$

  end for

  product all $W_{j,i}$, $0 \le i < m$, named W_j, $\Lambda_{\beta j} \leftarrow \Lambda_{\beta j,m-1}||\Lambda_{\beta j,m-2}||\ldots||\Lambda_{\beta j,0}$, $\Lambda_{\gamma j} \leftarrow \Lambda_{\gamma j,m-1}||\Lambda_{\gamma j,m-2}||\ldots||\Lambda_{\gamma j,0}$, compute $\Gamma_{j+1,R} \leftarrow \Lambda_\beta \lll \beta$

end for

compute $W = W \times W_j$

if $w \ge C_{lc}$ **then**

  add $LC \leftarrow (\Gamma_j)$, $round++$

  call search($\Gamma_{j+1,L}, \Gamma_{j+1,R}, round, W$)

else

  $\hat{C} \leftarrow W$

  if $round \ge r$ **then**

    return $\hat{C}$, LC

  end if

end if

$$\Lambda_\alpha = \Gamma_{i-1,L} \ggg \alpha,$$

$$\Lambda_\beta = (\Gamma_{i,R} \ggg \beta) \oplus \Gamma_{i-1,R},$$

$$\Lambda_\gamma = \Gamma_{i,L} \oplus \Gamma_{i,R}.$$

Then $cor_i = cor_{\Lambda_\alpha, \Lambda_\beta \Rightarrow \Lambda_\gamma}$. Algorithm 2 illustrates how to search the linear approximation of SPECK.

Algorithm 2 can be speeded up by the branch-and-bound algorithm [9], in this way, set the best correlation for the first $(r-1)$-round as $C_1, C_2, \ldots, C_{r-1}$ and change the condition judgement on the 16-th line into $W \ge C_{lc}/C_{r-j}$.

The reason we divide the masks into byte in Algorithm 2 is to deduce the time complexity. The complexity of Algorithm 2 is about $O((n/8)^r (logm)^r)$, where n is the branch size, r is the search round and m is the total number of elements in the $pLMT_0$ or $pLMT_1$. When the sizes of $pLMT_0$ and $pLMT_1$ are small, the complexity will be not high. With threshold $C = 2^{-3}$ the sizes of the two tables used in analyzing SPECK32/SPECK48 are 90 KB and 50 KB, respectively. The following observation is obtained by experiments.

Table 4
Linear approximations for SPECK32, SPECK48.

r	SPECK32			SPECK48		
	Γ_L	Γ_R	w_r	Γ_L	Γ_R	W_r
0	a0	629	0	800501	e10625	0
1	78a0	18a1	2^{-1}	13100	3107	2^{-2}
2	90	6021	2^{-4}	18001	181b0	2^{-2}
3	6080	4081	2^{-1}	10000	180	2^{-2}
4	80	1	2^{-1}	100	0	2^{-1}
5	1	0	0	9	8	0
6	e00	c00	2^{-1}	618040	680040	2^{-2}
7	3040	3058	2^{-3}	24d00	420c00	2^{-4}
8	82	c0e2	2^{-2}	107813	107a7a	2^{-4}
9	1f8e	1b8f	2^{-1}	100	1b1150	2^{-5}
10				d88a89	d88a88	0
W	2^{-14}			2^{-22}		

Observation 1. Denote the linear mask of some middle round by Γ_{mid}, set $w(\Gamma_{mid}) = 1$. By extending it from Γ_{mid}, the linear approximation would cover the most rounds under the restriction of a reasonable correlation. The correlation for the same rounds of the linear approximation in other cases will be lower.

In order to search the backward linear approximation, the round function is replaced by the inverse function. Connecting the forward and backward linear trails, the linear approximation is obtained. And their splicing is the linear trail. The approximations for SPECK32/SPECK48 are illustrated in Table 4, where the mask is represented in hexadecimal.

Since the pLMTs are invoked many times, they will be stored using the red-black tree structure. The time of traversing the pLMT will be reduced to $O(log\ m)$, where m is the total number of elements in the tree. In the small block size setting, the search runs fast, but reversely for SPECK with state sizes 64, 96 and 128 bits. It is impracticable to search the approximation by the *pLMT* with the threshold $C = 2^{-3}$ on this occasion. The threshold should be reduced, but the search will take a long time. So we study the linear approximations which were already got through the search. It's easy to see that the correlation is inversely proportional to the hamming weight of the masks. We search the approximation from the middle round with hamming wight 1. The hamming weight increases first and then decreases from the middle to both ends. Then we used the Segment Searching method to search the forward or backward linear approximation. The forward (backward) search process is divided into three parts. Firstly, we search several rounds from the middle round with hamming wight 1 using the small *pLMT*. The hamming weight of the last round mask will be large. Secondly, in the next three rounds we search the characteristic using the *pLMT* whose threshold $C = 2^{-4}, 2^{-4}, 2^{-5}$ (for

Table 5
Linear approximations for SPECK64, SPECK96.

r	SPECK64			SPECK96		
	Γ_L	Γ_R	w_r	Γ_L	Γ_R	W_r
0	1b	10400003	0	1000120	140000010021	0
1	18	12000018	2^{-2}	18100	200000000101	2^{-2}
2	c0	100000c0	2^{-1}	100	1	2^{-1}
3	604	80000604	2^{-1}	1	0	0
4	26003000	20003004	2^{-2}	d0000000000	c0000000000	2^{-1}
5	1078101	1318121	2^{-3}	604500000000	604c00000000	2^{-2}
6	12401	1802100	2^{-4}	224d000003	6228000003	2^{-4}
7	1010000	10100	2^{-7}	181070001018	1b105a680018	2^{-12}
8	18000	100	2^{-4}	1200000010	180210400000	2^{-6}
9	100	0	2^{-1}	101000000000	101a000000	2^{-3}
10	9	8	0	1800000000	10000000	2^{-2}
11	10000044	c0000044	2^{-2}	10000000	0	2^{-1}
12	46980224	20800224	2^{-3}	d00000	c00000	2^{-1}
13				6041800	6048000	2^{-2}
14				30003490	30043080	2^{-3}
15				180181910500	800181a10526	2^{-5}
W	2^{-30}			2^{-45}		

Table 6
Linear approximation for SPECK128.

r	SPECK128		
	Γ_L	Γ_R	W_r
0	1200012000008040	6a11403194008010	0
1	1001018000040600	501a018120040680	2^{-6}
2	8000010000201134	8010000100201432	2^{-6}
3	181019104	268000000101a11d	2^{-6}
4	1000130	400000000010021	2^{-5}
5	18100	2000000000000101	2^{-2}
6	100	1	2^{-1}
7	1	0	0
8	d00000000000000	c00000000000000	2^{-1}
9	6045000000000000	604c000000000000	2^{-2}
10	236d0000000003	63280000000003	2^{-4}
11	1818600000000018	1b18526c00000018	2^{-9}
12	1a0000000000c0	18021060000000c0	2^{-4}
13	10000000000606	c010130000000606	2^{-3}
14	3680000000003000	3080180000003006	2^{-3}
15	8500000000018181	8524c000000181b1	2^{-4}
16	e01f0000000c0001	211a0000000c0180	2^{-5}
W	2^{-61}		

the 64, 96, 128-bit version, respectively). So that the hamming weight of the last round is low. Thirdly, the several remaining rounds of the trail will be searched using the small *pLMT*. Table 5 and Table 6 list the best linear approximations obtained by the Algorithm 2 for 12, 15, 16 rounds of SPECK for 64, 96, 128-bits respectively.

4. Linear cryptanalysis of SPECK

Using the r-round linear approximation from round 2 to round $r+1$ described in Tables 4–6, the paper provides partial key recovery attacks on SPECK2n/k. The 2n-bit plaintext, ciphertext and the input of the r-th round are denoted as $P=(P_L,P_R)$, $C=(C_L,C_R)$ and $(X_{r-1,L},X_{r-1,R})$, where $(P_L,P_R)=(X_{0,L},X_{0,R})$, respectively. Since the first subkey k_0 does not change the absolute value of the correlation, it can be moved to the linear approximation. The linear approximation can be extended such that $\Gamma'_{0,L}=\Gamma_{0,L}\oplus\Gamma_{0,R}$, $\Gamma'_{0,R}=\Gamma_{0,R}$, $\Gamma'_{r+1,L}=(\Gamma_{r+1,L}\ggg\alpha)$, $\Gamma'_{r+1,R}=\Gamma_{r+1,R}$. The key recovery attack is illustrated in Fig. 4. Assuming that N known plaintexts are used, then it describes the details of the attacks on SPECK 96 and SPECK128, due to the space limitation.

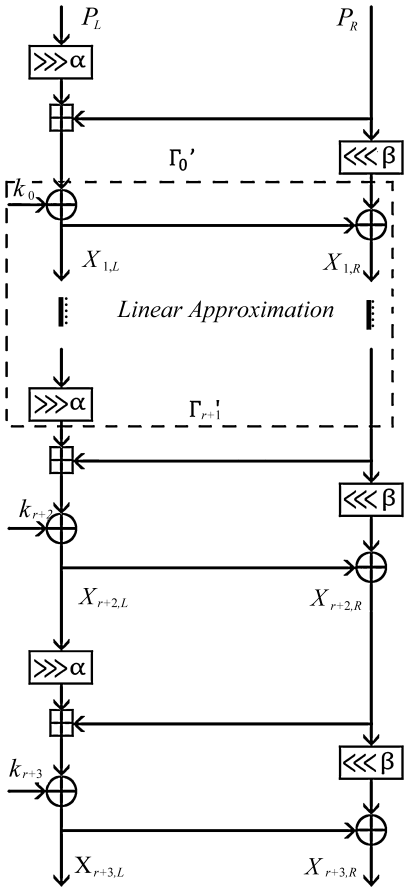

Fig. 4. Key recovery of SPECK.

4.1. Attacks on round reduced SPECK96 and SPECK128

The partial sum technique proposed by Ferguson et al. [6] will be used in the partial decryption procedures. With this method the time complexity will be reduced. The shortcoming is the increasing cost of memory. The details of the attack procedure (the case $k = 4n$) are as follows.

1. Compute $O \Leftarrow [\Gamma'_{0,L} \cdot ((P_L \ggg \alpha) \boxplus P_R)] \oplus [\Gamma'_{0,R} \cdot (P_L \lll \beta)]$. Ask for the encryption of the plaintexts, and store them in Table T_0.
2. Guess the subkey k_{r+3}, and decrypt the ciphertexts to get $X_{r+2,L}$ and $X_{r+2,R}$. Then we have $X_{r+1,R}$. Compute $O \Leftarrow X_{r+1,R} \cdot \Gamma'_{r+1,R} \oplus O$. Store them in Table T_1. Suppose the highest nonzero bit of $\Gamma'_{r+1,L}$ is j.
3. Allocate 2^{2n-5} counters V_3 for $X_{r+2,L}[3, 4, \ldots, n-1]$, $X_{r+1,R}[3, 4, \ldots, n-1]$ and O.
4. Guess the low 3 bits of subkey $k_{r+2}[0, 1, 2]$, and get the value of $[(X_{r+2,L}[0, 1, 2] \oplus k_{r+2}[0, 1, 2]) \boxminus X_{r+1,R}[0, 1, 2]] \cdot \Gamma'_{r+1,L}[0, 1, 2]$. Suppose this value is o, $O \Leftarrow O \oplus o$. The borrow is set to be c_2, $X_{r+1,R}[3, 4, \ldots, n-1] \Leftarrow X_{r+1,R}[3, 4, \ldots, n-1] \boxplus c_2$.
5. From 3 to $j-1$, do the following substeps.
   (a) $i \Leftarrow 3$.
   (b) Allocate $2^{2n-2i-1}$ counters V_i for $X_{r+2,L}[i+1, i+2, \ldots, n-1]$, $X_{r+1,R}[i+1, i+2, \ldots, n-1]$ and O.
   (c) Guess subkey $k_{r+2}[i]$ and get $(X_{r+2,L}[i] \oplus k_{r+2}[i]) \boxminus X_{r+1,R}[i]$. The borrow is set to be c_i, $X_{r+1,R}[i+1, i+2, \ldots, n-1] \Leftarrow X_{r+1,R}[i+1, i+2, \ldots, n-1] \boxplus c_i$. If $\Gamma'_{r+1,L}[i] = 1$, Compute $O \Leftarrow (X_{r+2,L}[i] \oplus k_{r+2}[i]) \boxminus X_{r+1,R}[i] \oplus O$. Else, keep the value of O.
   (d) Add V_{i-1} to V_i.
   (e) Add one to i. Jump to step (b).
6. Allocate one counter for each guessed subkey. Calculate the items whose $O = 0$ in V_j. This value is the value of the corresponding subkey counter.
7. Suppose the number of those counter is N_c. Compute $\eta = |2N_c/N - 1|$.
8. Sort η by value in descending order. For the first 2^{n+j-a} values of η, output the corresponding values of the guessed subkey bits as candidate right subkeys.

If $k = 3n$, the step 2 is eliminated. The number of candidate right subkeys in step 8 will be changed to 2^{j-a}.

Estimation of Complexity Selçuk gave the method to estimate the success probability as follows [12],

$$\begin{aligned} Ps &= \Phi(2\sqrt{N}|p - 1/2| - \Phi^{-1}(1 - 2^{-a-1})) \\ &= \Phi(\sqrt{N}|c| - \Phi^{-1}(1 - 2^{-a-1})) \end{aligned} \tag{3}$$

where Ps is the success probability of attack, N is the number of known plaintexts, p is the probability that the linear approximation holds, c is the correlation of the linear approximation, a is the number of advantage bits we can get, Φ and Φ^{-1} are the normal distribution and its inverse, respectively.

Table 7
Estimation of complexity.

SPECK	Rounds	Ps	a	N	Time	Memory
96/144	17	0.80	2	2^{92}	2^{96}	2^{92}
128/192	18	0.80	2	2^{124}	2^{128}	2^{124}
128/256	19	0.80	2	2^{124}	2^{192}	2^{124}

If $k = 4n$, the time complexity of step 3 is $N \cdot 2^{n+3}$ $(r+3)$-round decryptions, since we assume the complexity of one memory access to T_1 to be about $(r+3)$-round decryptions. This is the dominant term of time complexity. In step 5, the time complexity is about $2^{n+3}[2 \times 2^{2n-5} + 2^2 \times 2^{2n-7} + \ldots + 2^{j-2} \times 2^{2n-2j+1}) = 2^{3n}(1 - 2^{2-j})$ single bit one-round decryptions. So the total time complexity is about $2 \cdot N \cdot 2^{n+3}$ $(r+3)$-round decryptions. Out of conservative estimation, we multiply a coefficient 2 for the time complexity of the rest steps does not exceed $N \cdot 2^{n+3}$ $(r+3)$-round decryptions. The memory complexity is about $max(2^{2n-5}, N)$, which is used to store all counters. The unit of memory is the block size. The counter could be reused for every guess of subkeys.

If $k = 3n$, the time complexity of step 3 is $N \cdot 2^3$ $(r+2)$-round decryptions. In step 5, the time complexity is about $2^{2n}(1 - 2^{2-j})$ single bit one-round decryptions. The total time complexity is about $2 \cdot N \cdot 2^3$ $(r+2)$-round decryptions.

The values of Ps, a, N, the number of rounds attacked, time complexity and memory complexity are shown in Table 7.

5. Conclusion

The paper analyzed the security of SPECK family against linear cryptanalysis and presented linear approximations for different variants of SPECK. It introduced a 15-round linear approximation which was much longer than the previous 6-round linear approximation to attack 17-round SPECK96/144. It gave a 16-round linear approximation which was much longer than the previous 6-round linear approximation to target up to 18-, 19-round of SPECK128/192, SPECK128/256, respectively. In those variants we can attack the same rounds as the best previous attacks.

Acknowledgements

The work has been supported by 973 program (No. 2013CB834205), NSFC Projects (No. 61133013, No. 61103237), Program for New Century Excellent Talents in University of China (NCET-13-0350).

References

[1] F. Abed, E. List, S. Lucks, J. Wenzel, Differential cryptanalysis of round-reduced SIMON and SPECK, in: FSE2014, in: LNCS, vol. 8540, Springer, Berlin, Heidelberg, 2014, pp. 525–545.

[2] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, The SIMON and SPECK lightweight block ciphers, in: Proceedings of the 52nd Annual Design Automation Conference, ACM, June 2015.

[3] A. Biryukov, A. Roy, V. Velichkov, Differential analysis of block ciphers SIMON and SPECK, in: FSE2014, in: LNCS, vol. 8540, Springer, Berlin, Heidelberg, March 2014, pp. 546–557.
[4] S.M. Dehnavi, A.M. Rishakani, M.M. Shamsabad, A more explicit formula for linear probabilities of modular addition modulo a power of two, Cryptology ePrint archive, report 2015/026, 2015, http://eprint.iacr.org/.
[5] Dinur, I. Improved, Differential cryptanalysis of round-reduced SPECK, in: SAC 2014, in: LNCS, vol. 8781, Springer International Publishing, 2014, pp. 147–164.
[6] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, D. Whiting, Improved Cryptanalysis of Rijndael, in: FSE2000, in: LNCS, vol. 1978, Springer, Berlin, Heidelberg, January 2001, pp. 213–230.
[7] M. Matsui, The first experimental cryptanalysis of the data encryption standard, in: Advances in Cryptology, Crypto'94, Springer, Berlin, Heidelberg, January 1994, pp. 1–11.
[8] M. Matsui, Linear cryptanalysis method for DES cipher, in: Advances in Cryptology, EUROCRYPT'93, in: LNCS, vol. 1978, Springer, Berlin, Heidelberg, January 1994, pp. 386–397.
[9] M. Matsui, On correlation between the order of S-boxes and the strength of DES, in: Advances in Cryptology, EUROCRYPT'94, in: LNCS, vol. 1978, Springer, Berlin, Heidelberg, January 1995, pp. 366–375.
[10] K. Nyberg, J. Wallén, Improved linear distinguishers for SNOW 2.0, in: FSE2006, in: LNCS, vol. 4047, Springer, Berlin, Heidelberg, January 2006, pp. 144–162.
[11] E. Schulte-Geers, On CCZ-equivalence of addition mod 2^n, Des. Codes Cryptogr. 66 (1–3) (2013) 111–127.
[12] A.A. Selçuk, On probability of success in linear and differential cryptanalysis, J. Cryptol. 21 (1) (2008) 131–147.
[13] H. Tupsamudre, S. Bisht, D. Mukhopadhyay, Differential fault analysis on the families of SIMON and SPECK ciphers, in: FDTC2014, IEEE, September 2014, pp. 40–48.
[14] Wallén, J. Linear, Approximations of addition modulo 2^n, in: FSE2003, in: LNCS, vol. 2887, Springer, Berlin, Heidelberg, January 2003, pp. 261–273.
[15] Y. Yao, B. Zhang, W. Wu, Automatic search for linear trails of the SPECK family, in: ISC 2015, in: LNCS, vol. 9290, Springer, August 2015, pp. 158–176.