Linear Cryptanalysis of Reduced-Round Speck

Tomer Ashur Daniël Bodden
KU Leuven and iMinds
Dept. ESAT, Group COSIC
Address Kasteelpark Arenberg 10 bus 2452, B-3001 Leuven-Heverlee, Belgium
tomer.ashur-@-esat.kuleuven.be dbodden-@-esat.kuleuven.be

Abstract

Since DES became the first cryptographic standard, block ciphers have been a popular construction in cryptology. Speck is a recent block cipher developed by the NSA in 2013. It belongs to the cipher family known as ARX. ARX constructions are popular because of their efficiency in software. The security of the cipher is derived from using modular addition, bitwise rotation and xor. In this paper we employ linear cryptanalysis for variants of Speck with block sizes of 32, 48, 64, 96, and 128 bits. We illustrate that linear approximations with high bias exist in variants of Speck.

1 Introduction

A recent surge of new block cipher designs has urged the need for cryptanalysts to scrutinize their security. Lightweight ciphers are designed for resource-constrained devices. Designing ciphers for these devices means that one has to make design trade-offs keeping in mind the limitation that such an environment presents, such as limited memory, restricted computational and energy resources, etc.

A popular construction for block ciphers is Addition, Rotation and XOR, abbreviated as ARX. In this construction a block is split into 2 or more words, which are then added, XORed and rotated by the round function. The popularity of ARX construction stems from its good performance in software. Confusion is achieved by using modular addition in the round function, whereas diffusion is achieved by using cyclic rotation and xor.

In this paper we analyse the security of Speck, a block cipher that has been published by the NSA in 2013 [2]. Speck is a lightweight block cipher designed to achieve good performance in software. The block consists of two words, each 16/24/32/48, or 64 bits, which are processed a number of times by the round function. At the end of the last round a ciphertext is obtained. Speck comes in different variants for which different security and performance levels are provided; see Table 1.

Since the publication of the cipher in 2013, several papers have analyzed its security [1, 4, 7]. The best published attacks on Speck are differential cryptanalytic attacks described in [7]. In this paper we analyse the resistance of all variants of Speck against linear cryptanalysis.

Linear cryptanalysis is a known plaintext attack developed in 1993 by Matsui [9]. When using this method the adversary searches for possible correlations between bits of the input and bits of the output. Once such a correlation is found, the known plaintexts and ciphertexts can be used to recover bits of the secret key. In our analysis we find linear approximations covering 7, 8, 11, 10, and 11 rounds for versions of Speck with block size 32, 48, 64, 96, and 128, respectively. These approximations can be further extended to attack more rounds using methods such as those presented in [8].

This paper is structured as follow: In section 2 we describe Speck. In section 3 we discuss shortly the related work. In section 4 we explain the automatic search method that has been used to obtain linear approximations for Speck. In section 5

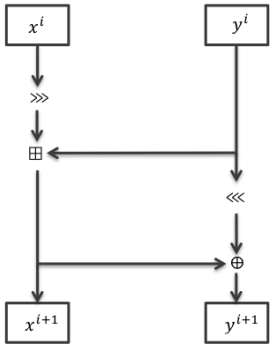

Figure 1: Speck round function

we present the approximations we found. In section 6 we explain how to obtain a linear distinguisher from the linear trails presented in section 5. Finally, in section 7 we conclude this paper.

2 A Brief Description of Speck

The cipher comes in several versions sharing the same Feistel-structure and round function. The different parameters of all Speck versions are presented in Table 1.

Table 1: Variants of the Speck-family [2]

Block Size n	Key Size	Word Size	α	β	Rounds
32	64	16	7	2	22
48	72	24	8	3	22
	96				23
64	96	32	8	3	26
	128				27
96	96	48	8	3	28
	144				29
128	128	64	8	3	32
	192				33
	256				34

The round function uses three operations:

- ◇ Modular addition, $\boxplus$.
- ◇ Left and right circular bit-shifts, by α and β.
- ◇ Bitwise XOR, $\oplus$.

The output of the round function is:
$x^{i+1} = (x^i \ggg \alpha) \boxplus (y^i)$
$y^{i+1} = ((x^i \ggg \alpha) \boxplus (y^i)) \oplus (y^i \lll \beta))$
The output words x^{i+1} and y^{i+1} are the input words for the next round, and the output of the last round is the ciphertext. In Figure 1 the round function of Speck is shown.

Table 2: Previous attacks concerning Speck

Variant n/key size	Rounds Attacked/ Total Rounds	Time	Data	Memory	Source
32/64	11/22	$2^{46.7}$	$2^{30.1}$	$2^{37.1}$	[1]
	14/22	2^{63}	2^{31}	2^{22}	[7]
48/72	12/22	2^{43}	2^{43}	na	[4]
	14/22	2^{65}	2^{41}	2^{22}	[7]
48/96	12/23	2^{43}	2^{43}	na	[4]
	15/23	2^{89}	2^{41}	2^{22}	[7]
64/96	16/26	2^{63}	2^{63}	na	[4]
	18/26	2^{93}	2^{61}	2^{22}	[7]
64/128	16/27	2^{63}	2^{63}	na	[4]
	19/27	2^{125}	2^{61}	2^{22}	[7]
96/96	15/28	$2^{89.1}$	2^{89}	2^{48}	[1]
	16/28	2^{85}	2^{85}	2^{22}	[7]
96/144	16/29	$2^{135.9}$	$2^{90.9}$	$2^{94.5}$	[1]
	17/29	2^{133}	2^{85}	2^{22}	[7]
128/128	16/32	2^{116}	2^{116}	2^{64}	[1]
	17/32	2^{113}	2^{113}	2^{22}	[7]
128/192	18/33	$2^{182.7}$	$2^{126.9}$	$2^{121.9}$	[1]
	18/33	2^{177}	2^{113}	2^{22}	[7]
128/256	18/34	$2^{182.7}$	$2^{126.9}$	$2^{121.9}$	[1]
	19/34	2^{241}	2^{113}	2^{22}	[7]

3 Related Work

Since the publication of Speck there has been a fair amount of analyzing done on the security and performance of the cipher. The best attacks to date are of the family of differential cryptanalysis [1, 4, 7]. Differential cryptanalysis has been developed by Biham and Shamir in 1990 [3]. When using differential cryptanalysis the adversary investigates how differences in the input affect the output, trying to discover non-random behavior. Several specialized techniques of differential cryptanalysis have been used to analyse Speck, such as the boomerang and the rectangle attacks [4, 7].

Results of previous attacks concerning Speck are presented in Table 2. From these results we gather that these methods are successful in attacking a large number of rounds in the chosen plaintext model. Looking at the design of Speck we might expect that linear cryptanalysis be effective as well. This paper complements previous work by evaluating the resistance of Speck against linear cryptanalysis.

3.1 Linear Cryptanalysis

Linear cryptanalysis [9] is a powerful cryptanalytic tool with regards to cryptanalysis of block ciphers. When using linear cryptanalysis, an adversary tries to find a linear expression that approximates a non-linear function with a probability different than $\frac{1}{2}$. When a good approximation, consisting of a relation between the plaintext and ciphertext, is found, the adversary gains information about the secret key. The approximation has the form:

$$P_i \oplus \ldots \oplus P_j \oplus C_k \ldots \oplus C_l = K_m \ldots \oplus K_n \tag{1}$$

with $P_i \dots P_j$ being plaintext bits, $C_k \dots C_l$ ciphertext bits and $K_m \dots K_n$ key bits. The approximation holds with some probability p, and its quality is usually measured by the bias which is defined as $\epsilon = |p - \frac{1}{2}|$.

Knowing the probability of the linear approximation, the adversary can determine the number of required known plaintexts, i.e., the data complexity ϵ^{-2}.

Once a good approximation is found the adversary can retrieve one bit of the key. To combine linear approximations we use the Piling Up Lemma [9]. This will tell us the bias of the combined approximation, which is the amount the probability deviates from $\frac{1}{2}$. For two approximations, with ϵ_1 and ϵ_2 as their respective biases, combining them gives an overall bias of:

$$\epsilon = 2 \cdot \epsilon_1 \epsilon_2. \tag{2}$$

This can be generalized for σ approximations:

$$\epsilon = 2^{\sigma-1} \cdot \prod_{i=1}^{\sigma} \epsilon_i. \tag{3}$$

The combined approximations can be used to form a linear distinguisher ζ for the cipher. This ζ can be used to detect non-random behavior in supposed to be random signals. Meaning that ζ can be used to detect if a certain cipher is being used. When a good ζ is found, the input bits are masked by $\{\lambda_{x^1}; \lambda_{y^1}\}$, and the output bits are masked using $\{\lambda_{x^r}; \lambda_{y^r}\}$, resulting in:

$$T = \#\left\{\lambda_{x^1} \cdot x_1 \oplus \lambda_{y^1} \cdot y_1 \oplus \lambda_{x^r} \cdot x_r \oplus \lambda_{y^r} \cdot y_r = 0\right\} \tag{4}$$

with x_1 and y_1 being the plaintexts, x_r and y_r the ciphertexts, r the length of ζ, T a counter and $\cdot$ the dot product. This should be repeated for at least ϵ^{-2} messages. If T is around:

$$\epsilon^{-2} \cdot (\frac{1}{2} \pm \epsilon) \tag{5}$$

the cipher can be distinguished from a random permutation.

3.2 Properties of Modular Addition

The modular addition operation, which is used as the nonlinear operation of Speck, consists of an xor and a carry. A chain of carries means using the previous carry in the current operation. This makes the behavior of modular addition nonlinear. To approximate the behavior of Speck, we have to deal with modular addition in such a way that accounts for this nonlinear behavior. The property of modular addition that we use is exploring the correlation between two neighboring bits. Suppose we have 3 words x, y and z with $z = x \boxplus y$. According to Cho and Pieperzyk [5], each single $z_{(i)}$ bit written as function of $x_{(i)}$,$\cdots$,$x_{(0)}$ and $y_{(i)}$,$\cdots$,$y_{(0)}$ bits can be expressed as:

$$z_{(i)} = x_{(i)} \oplus y_{(i)} \oplus x_{(i-1)} \oplus y_{(i-1)} \oplus \sum_{j=0}^{i-2} x_{(j)} y_{(j)} \prod_{k=j+1}^{i-1} x_{(k)} \oplus y_{(k)}, \quad i = 1, \dots, n-1 \tag{6}$$

with x_i and y_i each representing 1 bit each. The carry is represented by $x_{i-1} \oplus y_{i-1} \oplus \sum_{j=0}^{i-2} x_j y_j \prod_{k=j+1}^{i-1} x_k \oplus y_k$ which we refer to as $R_i(x, y)$. The parity of two consecutive bits can be approximated as:

$$z_{(i)} \oplus z_{(i-1)} = x_{(i)} \oplus y_{(i)} \oplus x_{(i-1)} \oplus y_{(i-1)}, \quad p = \frac{3}{4} \tag{7}$$

In this expression, we use a property of modular addition mentioned in another paper by Cho and Pieperzyk [6] that removes the carry chain from Equation 6. Meaning that if a mask λ contains only two consecutive bits, one can write:

$$P\left[\lambda \cdot (x \boxplus y) = \lambda \cdot (x \oplus y)\right] = \frac{3}{4}. \tag{8}$$

This expression means that using a mask λ to mask the bits we want to throw away and keep the bits we are interested in (e.g two consecutive bits), we linearize the left side of the expression by replacing it with the right-side. This approximation holds with probability $\frac{3}{4}$ and has a bias $\frac{1}{4}$. For an ARX construction, Equation 8 remains valid as long as the following two conditions are avoided:

1. Bitwise rotation moves a pair of approximated bits to the MSB (most significant bit position) and LSB (least significant bit position) hence not adhering to the Cho and Pieperzyk framework; or
2. Xor creates a single bit hence not adhering to the Cho and Pieperzyk framework.

4 Automated Search for Linear Approximations in ARX Constructions

In order to automate the search for a good linear approximation, we have designed an ARX toolbox. The toolbox is implemented using a parallel programming language called OpenCL. The configuration on which the computation was run has been a 40 core Intel Xeon machine with a clock speed of 3.10GHz, exclusively using CPU's. The time it took to compute all pairs of consecutive bits for the 32-bit input variant of Speck was less than 0.01 seconds to less than a week for the largest version of Speck.

We have analyzed in our work all possible pairs of consecutive bits over the length of a word (e.g. starting with one pair of two consecutive bits up to $\frac{n}{2}$ pairs of two consecutive bits). We have done this analysis in three ways, forward direction, backward direction and combining forward and backward together. We have iterated over all pairs of consecutive bits until one of the two stop conditions was met. The first stop condition is when a mask containing non-consecutive bits is encountered in the round function entering into the modular addition. The other condition upon which the computation will be stopped is when the counter exceeds the maximum bias. In order to find approximations with good bias we restrict the allowable counter to $T \leq \frac{n}{2} + 1$. The bias for one pair of consecutive bits is $1 - P[R_i(x, y) = 0] = \frac{1}{4}$. Generalized to ℓ pairs of consecutive bits the bias can be calculated using the Piling Up Lemma:

$$2^{\ell-1} \cdot (1 - P[R_i(x, y) = 0])^{\ell} \tag{9}$$

with values for $\ell = 1, \ldots, \frac{n}{2}$. The bias is calculated with the value obtained by the hamming weight of the mask passing the modular addition in the round function divided by 2, plus the bias of the previous rounds, calculated on the approximated modular addition $\lambda \cdot (x \oplus y)$.

5 Linear Approximation of the Round Function of Speck

Linear cryptanalysis relies on collecting a large amount of input and output pairs in order to verify whether the approximation has been satisfied or not. In this work we

started with fixing one mask λ_{x^1} with one pair of consecutive bits (0x3, 0x6, ...) and keeping the other mask λ_{y^1} zero, checking how the masks evolve both in the forward and backward direction. After receiving promising results for one pair we extended the computation to all possible pairs of two consecutive bits given a block size.

An overview of the obtained results from our experiments is shown in Table 3. These results show that the distinguisher can cover up to 11-rounds for different versions of Speck. More details of linear cryptanalysis on Speck are given in Table 4.

A careful note to the reader when reviewing the results in Table 4 is that for 96-bit version of Speck we retrieved 10-rounds, whereas for the 64 and 128-bit versions we retrieved 11-rounds each. The reason for the different r between these versions is stop condition 1. For each version of Speck, the length of a word and mask vary according to n. Meaning that bitwise rotation operates over a different length (i.e., $\frac{n}{2}$).

We present in Table 5 the full linear trail in the forward and in the backward direction for the 32/64 variant. The information found in this table gives an idea why the computation broke in that particular round. Using this information in future research could extend the distinguisher further.

Table 3: Distinguisher length for different versions of Speck

Variant n	Distinguished rounds / Total rounds	Bias	Time Complexity	Data Complexity
32	7/22	2^{-14}	2^{28}	2^{28}
48	8/22	2^{-22}	2^{44}	2^{44}
64	11/26	2^{-32}	2^{64}	2^{64}
96	10/28	2^{-47}	2^{92}	2^{92}
128	11/32	2^{-63}	2^{144}	2^{144}

Table 4: Combined results for different versions of Speck

Variant n	Distinguished rounds / Total rounds	Bias ϵ	λ_{x^1}	λ_{y^1}	Number of rounds backward	Number of rounds forward
32	7/22	2^{-14}	0x0006	0x0000	3	4
48	8/22	2^{-22}	0x000003	0x000000	5	3
64	11/26	2^{-32}	0x00000003	0x00000000	8	3
96	10/28	2^{-47}	0x000000000000	0x01800000000c	5	5
128	11/32	2^{-63}	0x0000003000000003	0x0000000000000000	8	3

Table 5: Linear trail for Speck32/64

Direction	Cost	λ_{x^i}	λ_{y^i}	$\lambda_{x^{i+1}}$	$\lambda_{y^{i+1}}$	Reasons stopped
				0x0300	0xe0c7	broke consecutive for 0x300 $\oplus$ 0xe0c7
	1	0x0300	0xe0c7	0x8301	0x8307	
	2	0x8301	0x8307	0x0300	0x0006	
Backward	1	0x0300	0x0006	0x0006	0x0000	
Forward	1	0x0006	0x0000	0x3c00	0x3000	
	2	0x3c00	0x3000	0xc198	0xc1e0	
	3	0xc198	0xc1e0	0xf00c	0xc18f	
	3	0xf00c	0xc18f	0x785f	0x61bf	
		0x785f	0x61bf			broke consecutive for 0x785f $\ggg \alpha$

6 A Linear Distinguisher for Speck

Using the results from this paper we can build a linear distinguisher for each version of Speck. We present here a distinguisher for the 32-bit version. To build the linear distinguisher we use Equation 4 and fill in with information from Table 4, giving:

$$2^{28} \cdot (\frac{1}{2} \pm 2^{-14}) \tag{10}$$

using this distinguisher we can distinguish Speck32/64 from a random permutation. In Algorithm 1 a pseudo code of the distinguishing attack is shown.

Data: 2^{28} messages, input masks $\{\lambda_{x^1}; \lambda_{y^1}\}$, output masks $\{\lambda_{x^r}; \lambda_{y^r}\}$
set $T = 0$
for 2^{28} *messages* **do**
  if $\{\lambda_{x^1} \cdot x \oplus \lambda_{y^1} \cdot y \oplus \lambda_{x^r} \cdot x \oplus \lambda_{y^r} \cdot y = 0\}$ **then**
    $T = T + 1$
  end
end
if $T \approx 2^{28} \cdot (\frac{1}{2} \pm 2^{-14})$ **then**
  return 1
else
  return 0
end

Algorithm 1: A pseudo-code of the distinguishing attack

7 Conclusion

In this paper we investigated the linear behavior of Speck in the known plaintext model. We explained the theory behind our method, presented linear approximations for each version of Speck and gave one example a linear distinguisher for Speck32/64.

The analysis in this paper tested the strength of the Speck round function and demonstrated that the cipher offers sufficient resistance against linear cryptanalysis. Future work may find better linear approximations. Proven that linear cryptanalysis can achieve a meaningful distinguisher, we believe that future research on ARX constructions may deliver good linear approximations for these ciphers. Our tool can be used to test the resistance of such constructions performing the computation with the latest parallel computing techniques.

Acknowledgments. The authors wish to Thank Vincent Rijmen for his support, Mathy Vanhoef for reviewing an earlier version of this paper, and R. Tzach for some thought provoking discussions.

This work was partially supported by the Research Fund KU Leuven, OT/13/071. The first author was also supported by Google Europe Scholarship for Students with Disabilities.

References

[1] Abed, F., List, E., Wenzel, J., Lucks, S.: Differential cryptanalysis of round-reduced simon and speck. Preproceedings of Fast Software Encryption (FSE 2014)(2014, to appear) (2014)

[2] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The simon and speck families of lightweight block ciphers. IACR Cryptology ePrint Archive (2013) 404

[3] Biham, E., Shamir, A.: Differential cryptanalysis of the data encryption standard. Volume 28. Springer-Verlag New York (1993)

[4] Biryukov, A., Roy, A., Velichkov, V.: Differential analysis of block ciphers simon and speck. In: International Workshop on Fast Software Encryption-FSE. (2014)

[5] Cho, J.Y., Pieprzyk, J.: Algebraic attacks on sober-t32 and sober-t16 without stuttering. In: Fast Software Encryption, Springer (2004) 49–64

[6] Cho, J.Y., Pieprzyk, J.: Multiple modular additions and crossword puzzle attack on nlsv2. In: Information Security. Springer (2007) 230–248

[7] Dinur, I.: Improved differential cryptanalysis of round-reduced speck. In: Selected Areas in Cryptography–SAC 2014. Springer (2014) 147–164

[8] Knudsen, L.R., Mathiassen, J.E.: A chosen-plaintext linear attack on des. In: Fast Software Encryption, Springer (2001) 262–272

[9] Matsui, M.: Linear cryptanalysis method for des cipher. In: Advances in Cryptology—EUROCRYPT'93, Springer (1994) 386–397