Cryptanalysis of the SPECK Family of Block Ciphers

Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel

Bauhaus-Universität Weimar, Germany
{farzaneh.abed, eik.list, stefan.lucks, jakob.wenzel}@uni-weimar.de

Abstract. SIMON and SPECK are two families of ultra-lightweight block ciphers which were proposed by the U.S. National Security Agency in June 2013. Yet, the specification paper discusses only the design and the performance of both cipher families, the task of analyzing their security has been left to the research community.
In this paper we present conventional differential as well as rectangle attacks for almost all members of the SPECK cipher family, where we target up to 11/22, 12/23, 14/16, 15/29, and 18/34 rounds of the 32-, 48-, 64-, 96-, and 128-bit version, respectively. In addition, we discuss rotational attacks, where we show that these attacks can be easily mounted for the full or almost the full number of rounds for large groups of weak keys.

Keywords: Differential cryptanalysis, block cipher, lightweight, SPECK

1 Introduction

Lightweight ciphers are optimized to operate on resource-constrained devices such as RFID tags, smartcards, or FPGAs that are limited with respect to their memory, battery supply, and computing power. In such environments, hardware and software efficiency is becoming more and more important. Besides ensuring efficiency, preserving a reasonable security is a main challenge in this area that getting a lot of attention and making it one of the ongoing research problem. During the last five years, many block ciphers have been developed to address this problem, including but not limited to mCrypton [18], HIGHT [13], PRESENT [6], KATAN [8], KLEIN [11], LED [12], and PRINCE [7].
In June 2013, the U.S. National Security Agency (NSA) contributed to this ongoing research by proposing two ARX-based families of ultra-lightweight block ciphers, called SIMON and SPECK, where the former is optimized for hardware (like PRESENT, LED, or KATAN), and the latter for software implementations (like KLEIN). Though, due to aggressive optimizations in their round function and the used rotation constants, both families perform well in hard- *and* software. The original paper of SIMON and SPECK presented only performance, specifications and implementation footprints [1,2], and was noticed by the cryptography research community in the work by Saarinen and Engels [19] in Summer 2012. The design team did not discuss any security assessment of these two ciphers regarding their resistance against common attacks and left the task of analyzing the security of their constructions to the research community.

Method	Cipher	Rounds		Data	Memory	Time	$\lvert\mathcal{KG}\rvert$
		Full	Att.	(CP)	(Bytes)		
Differential	Speck32/64	22	11	2^{31}	$2^{33.0}$	$2^{45.2}$	full
	Speck48/72	22	12	2^{44}	$2^{46.6}$	$2^{66.0}$	full
	Speck48/96	23	12	2^{44}	$2^{46.6}$	$2^{66.0}$	full
	Speck64/96	26	13	2^{55}	$2^{58.0}$	$2^{84.9}$	full
	Speck64/128	27	13	2^{55}	$2^{58.0}$	$2^{84.9}$	full
	Speck96/144	29	15	2^{87}	$2^{90.6}$	$2^{132.7}$	full
	Speck128/192	33	16	2^{121}	$2^{125.0}$	$2^{182.6}$	full
	Speck128/256	34	16	2^{121}	$2^{125.0}$	$2^{182.6}$	full
Rectangle	Speck32/64	22	11	2^{30}	$2^{33.6}$	$2^{61.1}$	full
	Speck48/72	22	11	2^{45}	$2^{45.0}$	$2^{67.0}$	full
	Speck48/96	23	12	2^{45}	$2^{48.2}$	$2^{91.0}$	full
	Speck64/96	26	13	2^{62}	$2^{62.0}$	$2^{91.9}$	full
	Speck64/128	27	14	2^{62}	$2^{64.3}$	$2^{123.7}$	full
	Speck96/144	29	15	2^{91}	$2^{91.0}$	$2^{136.0}$	full
	Speck128/192	33	17	2^{126}	$2^{126.0}$	$2^{186.9}$	full
	Speck128/256	34	18	2^{126}	$2^{128.3}$	$2^{251.4}$	full
Rotational	Speck32/64	22	21	$2^{30.7}$	$2^{32.7}$	$2^{30.7}$	$2^{35.7}$
Related-Key	Speck48/72	22	22	$2^{32.2}$	$2^{34.8}$	$2^{32.2}$	$2^{42.2}$
	Speck48/96	23	23	$2^{33.6}$	$2^{36.2}$	$2^{33.6}$	$2^{64.8}$
	Speck64/96	26	26	$2^{37.8}$	$2^{40.8}$	$2^{37.8}$	$2^{60.6}$
	Speck64/128	27	27	$2^{39.2}$	$2^{42.2}$	$2^{39.2}$	$2^{91.2}$
	Speck96/96	28	28	$2^{40.7}$	$2^{44.2}$	$2^{40.7}$	$2^{57.7}$
	Speck96/144	29	29	$2^{42.0}$	$2^{45.5}$	$2^{42.0}$	$2^{104.3}$
	Speck128/128	32	32	$2^{46.3}$	$2^{50.3}$	$2^{46.3}$	$2^{84.1}$
	Speck128/192	33	33	$2^{47.7}$	$2^{51.7}$	$2^{47.7}$	$2^{146.7}$
	Speck128/256	34	34	$2^{49.2}$	$2^{53.2}$	$2^{49.2}$	$2^{209.3}$

Table 1. Summary of our results on Speck. $\lvert\mathcal{KG}\rvert$ denotes the size of the weak-key group for which the given rotational attacks work.

Contribution. In this paper, we analyze Speck regarding to its resistance against differentials cryptanalysis. We show conventional key-recovery attacks on round-reduced versions of almost all family variants. Thereupon, we mount rectangle attacks where we use parts of our characteristics to extend the number of attacked rounds for the larger versions of the cipher. In addition, we consider rotational attacks in the related-key model where we follow the attacking principle from Khovratovich and Nikolić [15] for ThreeFish. We show that such attacks are possible on the full cipher (or almost the full cipher for Speck32/64) for large groups of weak keys. A complete summary of our results can be seen in Table 1.

Outline. In what follows, we first review the necessary details of SPECK in Section 2. The sections 3 and 4 present our differential and rectangle key-recovery attacks. We explain our rotational attacks in Section 5 before we conclude our paper in Section 6. Before, we list the notations used throughout this paper.

n	Word size.
$2n$	State size.
k	Size of the secret key in bits.
P_i, C_i	Plaintext-ciphertext pair.
(L^r, R^r)	Left (L) and right (R) halves of the state after encryption of Round r in a Feistel-cipher.
$L_{i,j}$	The i-th and j-th least-significant bit in L.
Δ_i	An n-bit (XOR) difference, where only the i-th bit is active. with $0 \leq i \leq n-1$ and Δ_0 denotes the least significant bit.
$\Delta_{i,[j]}$	An n-bit truncated difference, where only the i-th bit is active and the j-th bit is unknown.
Δ^r	Difference after Round r.
$\Delta^r \xrightarrow[E]{p} \Delta^s$	A differential characteristic which yields the output difference Δ^s with probability p when encrypting over a (sub-)cipher E and starting from an input difference Δ^r.

Table 2. Notations used throughout this paper.

2 SPECK

The SPECK$2n/k$ family is a simple ARX-based Feistel network, which processes the input as two words. At the beginning of a round, the left word of the state is rotated by α bits to the left, before the right word is added to it modulo 2^n. Next, a round key K^{i-1} is XORed to the left half. The right word is then rotated by β bits to the right, before the left word is XORed to the right. This procedure is depicted in Figure 1. The constants α and β are 8 and 3 for most versions of the cipher, except for SPECK$32/64$, which employs $\alpha = 7$ and $\beta = 2$.

Key Schedule. In contrast to the best-known ARX cipher ThreeFish, the designers of SPECK have applied a key addition in each round. To generate the round keys, the key schedule of SPECK re-uses the round transformation. At the beginning, m variables $K^0, \ell^0, \ldots, \ell^{m-2}$ are initialized with the words of the secret key: $(K^0, \ell^0, \ldots, \ell^{m-2}) \leftarrow (SK^0, SK^1, \ldots, SK^m)$ and the further round keys K^i are generated with the help of the following procedure:

$$\ell^{i+m-1} = K^i \boxplus (\ell^i \ggg \alpha) \oplus i,$$
$$K^{i+1} = (K^i \lll \beta) \oplus \ell^{i+m-1}.$$

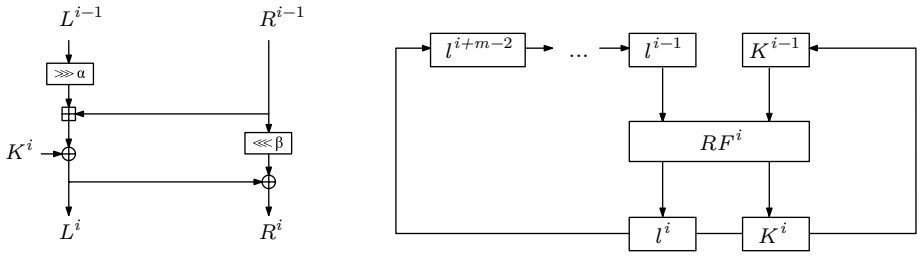

Fig. 1. Schematic views on the round function (left) and the key schedule (right) of SPECK. RF_i denotes the invocation of the round function, parametrized with i as the key.

Properties of our Differential Characteristics. We construct differential characteristics for SPECK by starting from a difference with a single active bit in the middle, and propagate towards start and end. To minimize the number of active bits, we build our trails on the events that the addition in each round will not produce any carry bits. Tables 5, 6, 7, and 8 (see Appendix A) list our characteristics for the individual versions of SPECK in detail.

3 Differential Attacks on SPECK

In the following, we describe our conventional differentials analysis of SPECK. Note that we describe only the attack on SPECK32/64 in detail since this version allows a simple practical verification. For attacks on the further versions of SPECK, we only provide the complexities and list the necessary details where these attacks differ from those in the smallest version.

3.1 Key-Recovery Attack on SPECK32/64

Here, we describe in brief an 11-round key-recovery attack on SPECK32/64. To do this, we use the characteristic from Table 5 over the rounds $4-11$ of the cipher:

$$\Delta^4 = (\Delta_{3,10,12}, \Delta_{3,6,12,13,14}) \xleftarrow[\text{rounds } 4-11]{p=2^{-25}} (\Delta_{1,3,8,10,15}, \Delta_{5,8,10,12,15}) = \Delta^{11}.$$

Then, later we guess key bits from the first rounds, which directly provide us with information about the secret key. We also know that ΔR^2 must be $\Delta_{4,8,11,12}$.

Attack Procedure. In the following, we simply denote by $\mathcal{A}$ a probabilistic algorithm or adversary which aims to recover the secret key for this cipher. The full attacking procedure can be split into a *collection phase*, and a *filtering phase*. The steps for the collection phase are as following:

1. Choose 2^{30} pairs (C_i, C_i') with $C_i \oplus C_i' = \Delta^{11}$.

2. Collect the corresponding plaintext pairs (P_i, P'_i) from a decryption oracle, where $P_i = E_K^{-1}(C_i)$ and $P'_i = E_K^{-1}(C'_i)$. Store all pairs (P_i, P'_i) in a list $\mathcal{P}$.

The filtering phase then consists of following steps:

3. For all key combinations K^0:
   3.1 Initialize $count \leftarrow 0$.
   3.2 For all pairs $(P_i, P'_i) \in \mathcal{P}$:
   - Partially encrypt (P_i, P'_i) to the state after the encryption of Round 2 and derive ΔR^2.
   - If $\Delta R^2 = \Delta_{4,8,11,12}$, then, for all values K^1, further encrypt (P_i, P'_i) to the state after the encryption of Round 3 and check if Δ^3 matches the expected difference. If this is the case, then increment *count*. Note that we do not guess any bits of the key in the third round, since the key addition does not affect our target difference.

   3.3 If $count > 11$, mark the current value K^0 as the (or one of few) potentially correct key candidates.

This attack works because of the following reasons: the probability that a pair follows our differential characteric is given by 2^{-25}. Hence, the probability that no more than eleven correct pairs occur when using SPECK can be approximated by

$$Pr[\textbf{false random}] := Pr^{Poisson}[n = 2^{30}, p = 2^{-25}, x \leq 11] \approx 1.70 \cdot 10^{-5}.$$

In this point, we also need to consider the probability of a false positive key. The probability that a pair produces the Δ^3 by random is 2^{-32}. So, for one specific value of the guessed keys, the probability that more than eleven false-positive pairs occur is

$$1 - Pr^{Poisson}[n = 2^{30}, p = 2^{-32}, x \leq 11] \approx 2^{-53.38}.$$

Since $\mathcal{A}$ guesses 32 key bits, the probability that any key candidate produces more than eleven false-positive pairs is about

$$Pr[\textbf{false real}] := 1 - Pr^{Poisson}[n = 2^{32}, p = 2^{-53.38}, x \leq 0] \approx 3.67 \cdot 10^{-7}.$$

Concluding, the error probability of $\mathcal{A}$ becomes very close to 0, if it interprets a key candidate as the secret key when at least eleven pairs satisfy Δ^3. At the end, $\mathcal{A}$ can use those correct text pairs for its found key candidate, and perform further partial encryptions over the rounds 4 and 5, to identify the correct values of K^2 and K^3.

Attack Complexity. The straight-forward application of our attack requires 2^{31} chosen ciphertexts. Concerning the memory complexity, $\mathcal{A}$ can store either a list of counters for all key candidates or a list of all plaintext pairs – the latter option gives us a memory complexity of $2^{31} \cdot 32/8 = 2^{33}$ bytes. The computational effort for the collection phase, C_{texts}, is equivalent to 2^{31} full decryptions performed by the oracle. The filtering effort, C_{filter}, is twofold. First, for 2^{16} values

K^0, we encrypt all pairs over the first two rounds. All pairs satisfying ΔR^2 and happening with probability 2^{-16} in average, are further encrypted over Round 3 for all values K^1. The brute-force effort to find the remaining bits of K^2 and K^3, $C_{\text{bruteforce}}$, can be overestimated by 2^{32} full encryptions. Summing up, we have

$$\underbrace{2 \cdot 2^{30}}_{C_{\text{texts}}} + \underbrace{\left(2^{16} \cdot \frac{2}{11} + 2^{-16} \cdot 2^{16+16} \cdot \frac{1}{11}\right) \cdot 2 \cdot 2^{30}}_{C_{\text{filter}}} + \underbrace{2^{32}}_{C_{\text{bruteforce}}} \approx 2^{45.2} \text{ encryptions.}$$

For the further versions of Speck, we can apply a similar procedure and get the following results which are summarized in Table 3.

State size	Key size	Rds.	Pr[diff.]	Prs.	Known bits at Δ^2	Known bits at Δ^3	Key bits	Thresh. prs.
32	64	11	2^{-25}	2^{30}	16	32	32	> 11
48	all	12	2^{-38}	2^{43}	24	48	48	> 11
64	all	13	2^{-49}	2^{54}	32	64	64	> 11
96	all	15	2^{-81}	2^{86}	48	96	96	> 11
128	all	16	2^{-115}	2^{120}	64	128	128	> 11

Table 3. Parameters of our differential attacks on Speck$2n/k$. Rds. = rounds, prs. = pairs.

4 Rectangle Attacks on Speck

4.1 Boomerang and Rectangle Attacks

Boomerangs [20] are differential-based attacks that allow an adversary to concatenate two "short" differential characteristics, which is beneficial for primitives where "long" characteristics would have a very low probability. Boomerang attacks have been first introduced by Wagner in 1999 [20], and were later transformed into a chosen-plaintext attack by Kelsey, Kohno, and Schneier [14], which they called it an *amplified boomerang*. In 2001, Biham, Dunkelman, and Keller added further improvements and renamed it to the *rectangle attack* [4]. In 2002, the same authors made more improvements for boomerang- and rectangle-based key-recovery attacks [5]. In 2010, Dunkelman, Keller, and Shamir [10] extended the technique by introducing the sandwich attack, where the adversary can insert a round between the two sub-ciphers if they have a differential with high characteristic probability.

Boomerang Attacks. In the basic setting of the attack, an adversary $\mathcal{A}$ first decomposes a given cipher E into two sub-ciphers $E = E_2 \circ E_1$, where it uses two differentials

$$\alpha \xrightarrow[E_1]{p} \beta \text{ and } \gamma \xrightarrow[E_2]{q} \delta,$$

with probability p and q, respectively. Then, $\mathcal{A}$ collects a pair (P, P') with $P \oplus P' = \alpha$ and asks an encryption oracle for their corresponding ciphertexts (C, C'). As a next, it derives two new ciphertexts $D = C \oplus \delta$ and $D' = C' \oplus \delta$, and asks the decryption oracle for their corresponding plaintexts (Q, Q'). If $Q \oplus Q' = \alpha$, then the adversary obtains a *correct quartet*. Each quartet (P, P', Q, Q'), has a probability of p^2, where their respective outputs after E_1, (R, R', S, S'), applies: $R \oplus R' = \beta$ and $S \oplus S' = \beta$. At this point, one is interested in the case when $R \oplus S = \gamma$ and automatically $R' \oplus S' = \gamma$, which is called the boomerang property. With probability q^2, the ciphertexts of such a quartet will produce the differences $C \oplus D = \delta$ and $C' \oplus D' = \delta$ and one obtains the correct quartet. Assuming that the adversary collects m pairs with difference α, then, the expected number of correct quartets is $m^2 \cdot 2^{-n} \cdot (pq)^2$.
For a random permutation, the number of correct quartets would be $m^2 \cdot 2^{-2n}$. So, in order to mount the attack, it must apply that $pq > 2^{-n/2}$. However, in this case, the adversary can count more correct quartets than the one would expect from a random permutation and it can distinguish E from random.

Amplified Boomerang/Rectangle Attacks. The standard boomerang procedure explained above represents an adaptive chosen plain-/ciphertext attack. Since this is a less practical scenario, Kelsey, Kohno, and Schneier developed amplified boomerangs which are pure chosen-plaintext attacks.
Following their method, the adversary chooses $\left(\frac{2^{n/2+2}}{pq}\right)$ plaintext pairs and let the oracle to encrypt them. Since any two pairs can be used to form a quartet, this gives the adversary $\left(\frac{2^{n+3}}{p^2q^2}\right)$ possible quartets. The difference γ holds with probability 2^{-n} after E_0, so one can expect a few correct quartets for which holds $C \oplus D = C' \oplus D' = \delta$.

4.2 Rectangle Attack on Speck32/64

In this section, we present rectangle attacks on round-reduced versions of Speck. For $\alpha \to \beta$ and $\gamma \to \delta$, we use only those parts of our characteristic in Appendix A which have a high probability.
In the following, we describe an 11-round rectangle attack on Speck32/64 in detail. Since our attacks on the further versions of Speck work similar, we only specify the used trails and their complexities here. For the smallest version we use the trails

$$\alpha = (\Delta_{11,12}, \Delta_4) \xrightarrow[E_0]{p=2^{-6}} (\Delta_{15}, \Delta_{1,3,10,15}) = \beta,$$

and

$$\gamma = (\Delta_{11,12}, \Delta_4) \xrightarrow[E_1]{q=2^{-6}} (\Delta_{15}, \Delta_{1,3,10,15}) = \delta.$$

Here, E_0 represents the rounds 4-7, and E_1 the rounds 8-11. The procedure for our attacks is as follows:

1. Choose $\left(\frac{2^{n/2+2}}{pq}\right) = (2^{32/2+2}/2^{-6-6}) = 2^{30}$ ciphertext pairs.
2. Initialize a set $\mathcal{K}$ of $2^{|K^0|+|K^1|} = 2^{16+16} = 2^{32}$ counters for all subkey bits in K^0 and K^1.
3. Ask an oracle for the decryption (P, Q) of all chosen ciphertext pairs and store them in a hash table.
4. For all possible values of the subkeys $K^0 \| K^1$:
   4.1 Encrypt all pairs (P, Q) over the first three rounds, and store the results as (S, T).
   4.2 For all combination of pairs (S, T), (S', T'), check whether their difference is equal to α, $(S \oplus S' = T \oplus T' = \alpha)$. If yes, then increment the counter for the current key candidate.
5. Output the key candidate with the maximal count in $\mathcal{K}$.

Attack Complexity. The attack requires 2^{31} chosen ciphertexts as a data complexity. Concerning the memory complexity, $\mathcal{A}$ need to store the ecncryption of all plaintext pairs beside the list of counters. So, it becomes $2^{31} \cdot 32/8 + 2^{32} \approx 2^{33.6}$ bytes. The computational effort for the collection phase, C_{texts}, is equivalent to 2^{31} full decryptions performed by the oracle. The filtering effort consists of encrypting 2^{30} pairs for 2^{32} key candidates over the first three rounds. The brute-force effort to find the remaining bits of K^2 and K^3, $C_{\text{bruteforce}}$, can be overestimated by 2^{32} full encryptions. Summing up, we have

$$\underbrace{2 \cdot 2^{30}}_{C_{\text{texts}}} + \underbrace{\left(2^{32} \cdot \frac{3}{11}\right) \cdot 2 \cdot 2^{30}}_{C_{\text{filter}}} + \underbrace{2^{32}}_{C_{\text{bruteforce}}} \approx 2^{61.1} \text{ encryptions.}$$

For the further versions of SPECK, we can apply a similar procedure. The parameters of our attacks with error probabilities of the adversary are summarized in Table 4.

State size	**Key size**	**Rounds**	$\mathbf{(pq)^2}$	$\mathbf{C_{data}}$	$\mathbf{C_{memory}}$	$\mathbf{C_{time}}$
32	64	11	2^{-24}	2^{30}	$2^{33.6}$	$2^{61.1}$
48	72	11	2^{-36}	2^{45}	$2^{45.0}$	$2^{67.0}$
48	96	12	2^{-36}	2^{45}	$2^{48.2}$	$2^{91.0}$
64	96	13	2^{-54}	2^{62}	$2^{62.0}$	$2^{91.9}$
64	128	14	2^{-54}	2^{62}	$2^{64.3}$	$2^{123.7}$
96	144	15	2^{-80}	2^{91}	$2^{91.0}$	$2^{136.0}$
128	192	17	2^{-118}	2^{126}	$2^{126.0}$	$2^{186.9}$
128	256	18	2^{-118}	2^{126}	$2^{128.3}$	$2^{251.4}$

Table 4. Parameters of our rectangle attacks on SPECK$2n/k$.

5 Rotational Cryptanalysis of SPECK

Since the round function of SPECK consists of only bit-wise operations, there is a chance of mounting attacks using rotational cryptanalysis. Thereby, instead of using pairs of texts with a XOR difference, an adversary collects pairs of the structure $(x, \overleftarrow{x})$, where $\overleftarrow{x}$ denotes x after rotation by a fixed value r. Note that this technique requires that the round keys used for the encryption to be rotated versions of each other. Thus, rotational cryptanalysis can be considered in less practical related-key model, or the usage of weak-key classes.

5.1 Rotational Cryptanalysis

Related-key attacks were introduced by Biham in [3], where he used a rotational pair of keys to attack LOKI and Lucifer. In 2009, Knudsen et al. employed a rotational pair of inputs to attack the compression function of Shabal [17]. In 2010, the term rotational cryptanalysis got more attention when Khovratovich, Nikolić, and Rechberger [15,16] analyzed the ARX-based ThreeFish cipher used by the SHA-3 finalist Skein.
In this work, we try to follow the notions of [15]. So, we denote a rotated variable by $\overleftarrow{x}$ and $\overrightarrow{x}$, where $\overleftarrow{x} = x \lll r$ and $\overrightarrow{x} = x \ggg r$, respectively. We call $(x, \overleftarrow{x})$ a *rotational pair* for a given fixed rotation amount r. It is easy to see that a rotational pair is preserved by any bitwise transformation, such as XOR or rotation:

$$\overleftarrow{x \oplus y} = \overleftarrow{x} \oplus \overleftarrow{y}, \qquad \overleftarrow{x} \ggg r = \overleftarrow{x \ggg r}.$$

In SPECK, the source of non-linearity in the round function w.r.t. rotational differences is the modulo addition 2^n which we denote that by '$+$'.

Lemma 1 (Daum, [9] after [15]). *The probability that a rotational pair (x, y) passes through an addition modulo 2^n is given by*

$$p_r = Pr[\overleftarrow{(x+y)} = \overleftarrow{x} + \overleftarrow{y}] = \frac{1}{4}\left(1 + 2^{r-n} + 2^{-r} + 2^{-n}\right).$$

For large n and small r, we obtain the following table as listed in [15]:

r	p_r	$log_2(p_r)$
1	0.375	-1.415
2	0.313	-1.676

Assuming that the inputs to all additions in the round function and key schedule of a cipher are independent and chosen at random from a uniform distribution. The probability that a rotational pair survives $|a|$ additions can be approximated by $(p_r)^{|a|}$. Thus, as stressed in [15], any ARX scheme which can be implemented with less than $n/1.415$ additions, can be considered vulnerable to the rotational cryptanalysis.

For SPECK, we can mount our rotational attacks on a high number of rounds for weak-key classes. Any version of SPECK consist of r rounds and employs the round function $r-1$ times in its key schedule to generate the round keys $K^1, \ldots, K^{r-1}$, which means $r-1$ additions in the key schedule. For rotational attacks, we are interested in such rotational pairs of keys that pass through all additions without rotational errors. This includes weak key groups with approximately $|\mathcal{KG}| = 2^{k-1.415(r-1)}$ elements. For such key groups, we can mount a related-key attack with the key $K, \hat{K}$, where $\hat{K} = \overleftarrow{K}$. In the following, let $\mathcal{A}$ be an adversary that chooses a text pair (P, P'), with $P' = \overleftarrow{P}$. There are r additions in the round function. Therefore, the probability for the resulting ciphertexts (C, C') to satisfy $C' = \overleftarrow{C}$ is given by $2^{-1.415r}$. Since every correct pair also provides information of the round keys (see [15]), we obtain a key-recovery attack with complexity of about $2 \cdot 2^{1.415r}$ encryptions. The parameters for attacks on the individual versions of full SPECK are given in Table 1.

6 Conclusion

In this work, we analyzed the security of the lightweight block cipher family SPECK by applying differential, rectangle, and rotational attacks as summarized in Table 1. To the best of our knowledge, our results are the first security analysis for SPECK, since the proposal did not include any form of security assessment. We could easily find conventional differentials for all versions of the cipher which helped us to mount differential and boomerang attacks on versions with up to half of the total number of rounds. More notably, we demonstrated that related-key rotational attacks can be applied to all full versions of SPECK for large groups of weak keys, which negates the designers' claim that there are no related-key attacks.
Since SPECK has a very simple ARX structure, any new attack on generalized ARX ciphers such as ThreeFish would be a threat to the security of SPECK. However, one positive security aspect of the NSA construction is the round-wise key addition and the simple, yet powerful key schedule, which protects very effectively against slide and meet-in-the-middle attacks over a reasonable number of rounds, as we noted during our studies. The security analysis in this paper can be seen as a starting point for upcoming research on the SPECK block cipher family. It would be interesting to see further investigation by using more sophisticated methods of cryptanalysis or improvements of our current results.

References

1. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. Performance of the SIMON and SPECK Families of Lightweight Block Ciphers. Technical report, National Security Agency, May 2012.
2. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. `http://eprint.iacr.org/`.

3. Eli Biham. New Types of Cryptanalytic Attacks Using Related Keys. *J. Cryptology*, 7(4):229–246, 1994.
4. Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer, 2001.
5. Eli Biham, Orr Dunkelman, and Nathan Keller. New Results on Boomerang and Rectangle Attacks. In Joan Daemen and Vincent Rijmen, editors, *FSE*, volume 2365 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2002.
6. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
7. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalcin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Wang and Sako [21], pages 208–225.
8. Christophe De Cannière and Orr Dunkelman and Miroslav Knezevic. KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In *CHES*, pages 272–288, 2009.
9. Magnus Daum. *Cryptanalysis of Hash functions of the MD4-family*. PhD thesis, Ruhr-University Bochum, 2005.
10. Orr Dunkelman, Nathan Keller, and Adi Shamir. A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2010.
11. Zheng Gong, Svetla Nikova, and Yee Wei Law. KLEIN: A New Family of Lightweight Block Ciphers. In Ari Juels and Christof Paar, editors, *RFIDSec*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2011.
12. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED Block Cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011.
13. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.
14. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In *Fast Software Encryption*, pages 75–93, 2000.
15. Dmitry Khovratovich and Ivica Nikolić. Rotational Cryptanalysis of ARX. In *Proceedings of the 17th international conference on Fast software encryption*, FSE'10, pages 333–346, Berlin, Heidelberg, 2010. Springer-Verlag.
16. Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational Rebound Attacks on Reduced Skein. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–19, 2010.
17. Lars Knudsen, Krystian Matusiewicz, and Søren S Thomsen. Observations on the shabal keyed permutation, 2009.

18. Chae Hoon Lim and Tymur Korkishko. mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In JooSeok Song, Taekyoung Kwon, and Moti Yung, editors, *WISA*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2005.
19. Markku-Juhani O. Saarinen and Daniel Engels. A Do-It-All-Cipher for RFID: Design Requirements (Extended Abstract). Cryptology ePrint Archive, Report 2012/317, 2012. `http://eprint.iacr.org/`.
20. David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.
21. Xiaoyun Wang and Kazue Sako, editors. *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*. Springer, 2012.

A Differential Characteristics for Speck2n/k

Rd.	**Speck32**/k			**Speck48**/k		
	ΔL^i	ΔR^i	ℓ	ΔL^i	ΔR^i	ℓ
0	$\Delta_{10,11,15}$	$\Delta_{4,8,11,12}$	0	$\Delta_{0,3,8,9,11,20,22}$	$\Delta_{0,3,6,9,11,16}$	0
1	$\Delta_{3,10,12}$	$\Delta_{3,6,12,13,14}$	-6	$\Delta_{1,6,9,11,12,14,19}$	$\Delta_{1,3,11}$	-7
2	$\Delta_{5,15}$	$\Delta_{0,8,14}$	-4	$\Delta_{4,6,17,22}$	$\Delta_{14,17,22}$	-5
3	$\Delta_{0,9}$	$\Delta_{2,9,10}$	-5	$\Delta_{9,17,20}$	$\Delta_{1,9}$	-3
4	$\Delta_{11,12}$	Δ_4	-3	Δ_{12}	Δ_4	-1
5	Δ_6	0	0	0	Δ_7	-1
6	Δ_{15}	Δ_{15}	-1	Δ_7	$\Delta_{7,10}$	-2
7	$\Delta_{8,15}$	$\Delta_{1,8,15}$	-2	$\Delta_{7,10,23}$	$\Delta_{7,13,23}$	-4
8	Δ_{15}	$\Delta_{1,3,10,15}$	-4	$\Delta_{2,7,13,15}$	$\Delta_{7,10,13,15,16}$	-7
9	$\Delta_{1,3,8,10,15}$	$\Delta_{5,8,10,12,15}$		$\Delta_{5,10,13,15,16,18,23}$	$\Delta_{5,15,19,23}$	-8
10				$\Delta_{2,7,8,10,19,21,23}$	$\Delta_{7,10,18,19,21-23}$	
Σ			-25			-38

Table 5. Differential characteristics for the smaller variants of Speck2n/k. ℓ denotes $\log_2(\mathbf{Pr})$.

Rd.	SPECK64/k		
	ΔL^i	ΔR^i	ℓ
0	$\Delta_{6,17,22,27,28}$	$\Delta_{14,17,27}$	0
1	$\Delta_{9,17,19,20,27,30}$	$\Delta_{9,19,27}$	-7
2	$\Delta_{1,11,12,22,27}$	$\Delta_{1,11,27,30}$	-9
3	$\Delta_{1,3,4,11,14,19,25,27,30}$	$\Delta_{3,11,19,25,27}$	-9
4	$\Delta_{6,17,22,28}$	$\Delta_{14,17,30}$	-5
5	$\Delta_{9,17,20}$	$\Delta_{1,9}$	-3
6	Δ_{12}	Δ_{4}	-1
7	0	Δ_{7}	-1
8	Δ_{7}	$\Delta_{7,10}$	-2
9	$\Delta_{7,10,31}$	$\Delta_{7,13,31}$	-4
10	$\Delta_{2,7,13,23}$	$\Delta_{7,10,13,16,23}$	-8
11	$\Delta_{5,7,10,13,15,16,23,26,31}$	$\Delta_{5,7,15,19,23,31}$	
Σ			-49

Table 6. Differential characteristics for SPECK64/k. ℓ denotes $\log_2(\mathbf{Pr})$.

Rd.	SPECK96/k		
	ΔL^i	ΔR^i	ℓ
0	$\Delta_{0,1,5,6,10,12,16,26,27,37,38}$	$\Delta_{1,2,5,18,27,37,46}$	0
1	$\Delta_{1,4,5,8,19,27,29,30,37,40,41,45}$	$\Delta_{19,21,27,29,37,41,45}$	-13
2	$\Delta_{0,11,22,27,32,33,44}$	$\Delta_{11,24,27,30,33,40}$	-11
3	$\Delta_{3,11,14,19,25,27,30,33,36}$	$\Delta_{3,11,19,25,43}$	-9
4	$\Delta_{6,17,22,28}$	$\Delta_{14,17,46}$	-5
5	$\Delta_{9,17,20}$	$\Delta_{1,9}$	-3
6	Δ_{12}	Δ_{4}	-1
7	0	Δ_{7}	-1
8	Δ_{7}	$\Delta_{7,10}$	-2
9	$\Delta_{7,10,47}$	$\Delta_{7,13,47}$	-4
10	$\Delta_{2,7,13,39}$	$\Delta_{7,10,13,16,39}$	-8
11	$\Delta_{5,7,10,13,16,31,39,42,47}$	$\Delta_{5,7,19,31,39,47}$	-10
12	$\Delta_{2,7,8,19,23,34,45}$	$\Delta_{7,10,19,22,23,42,45}$	-12
13	$\Delta_{0,7,10,11,15,19,22,23,26,37,45,47}$	$\Delta_{7,11,13,15,19,23,25,37,47}$	
Σ			-81

Table 7. Differential characteristics for SPECK96/k. ℓ denotes $\log_2(\mathbf{Pr})$.

Rd.	**SPECK128**/k		
	ΔL^i	ΔR^i	ℓ
0	$\Delta_{5,10,16,26,27,37,38,42,48,49,54,58,60}$	$\Delta_{5,18,27,34,37,46,49,50,2}$	0
1	$\Delta_{5,8,19,27,29,30,37,40,41,49,52,61}$	$\Delta_{19,21,27,29,41,53,61}$	-13
2	$\Delta_{11,22,27,32,33,44,0}$	$\Delta_{11,24,27,30,33,56}$	-11
3	$\Delta_{11,14,19,25,27,30,33,36,3}$	$\Delta_{11,19,25,59,3}$	-9
4	$\Delta_{6,17,22,28}$	$\Delta_{14,17,62}$	-5
5	$\Delta_{20,17,9}$	$\Delta_{9,1}$	-3
6	Δ_{12}	Δ_{4}	-1
7	0	Δ_{7}	-1
8	Δ_{7}	$\Delta_{7,10}$	-3
9	$\Delta_{7,10,63}$	$\Delta_{7,13,63}$	-4
10	$\Delta_{7,13,55,2}$	$\Delta_{7,10,13,16,55}$	-8
11	$\Delta_{5,7,10,13,16,47,55,58,63}$	$\Delta_{5,7,19,47,55,63}$	-11
12	$\Delta_{7,8,19,39,50,61,2}$	$\Delta_{7,10,19,22,39,58,61}$	-12
13	$\Delta_{7,10,11,19,22,31,39,42,53,61,63,0}$	$\Delta_{7,11,13,19,25,31,39,53,63}$	-17
14	$\Delta_{7,13,14,19,23,25,34,39,45,55,56,2,3}$	$\Delta_{7,10,13,16,19,22,23,25,28,39,42,45,55,3}$	
Σ			-115

Table 8. Differential characteristics for SPECK128/k. ℓ denotes $\log_2(\mathbf{Pr})$.