Cryptanalysis of the Speck Family of Block Ciphers
Revision From October 9, 2013

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 announced by the U.S. National Security Agency in June 2013. This paper presents differential and rectangle attacks for almost all members of the Speck family of ciphers, where we target up to 11/22, 12/23, 15/27, 15/29, and 18/34 rounds of the 32-, 48-, 64-, 96-, and 128-bit version, respectively.

Keywords: Differential cryptanalysis, block cipher, lightweight cipher.

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, hard- and software efficiency is becoming more and more important. Besides ensuring efficiency, preserving a reasonable security is a major challenge in this area that gets a lot of attention and making it one of the ongoing research problem. During the past five years, many block ciphers have been developed to address this problem, including but not limited to mCrypton [14], HIGHT [12], PRESENT [6], KATAN [8], KLEIN [10], LED [11], 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 intensive optimizations in their round function and the use of rotation constants, both families perform well in hard- *and* software. The original paper of Simon and Speck presented only performance, specifications and implementation footprints [2,3], and was noticed by the cryptography research community in the work by Saarinen and Engels [15] 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	Ref.
		Full	Att.	(CP)	(Bytes)		
Differential	Speck32/64	22	10	2^{29}	$2^{16.0}$	$2^{29.2}$	Sec. 3
	Speck48/72	22	12	2^{45}	$2^{24.0}$	$2^{45.3}$	Sec. 3
	Speck48/96	23	12	2^{45}	$2^{24.0}$	$2^{45.3}$	Sec. 3
	Speck64/96	26	15	2^{61}	$2^{32.0}$	$2^{61.1}$	Sec. 3
	Speck64/128	27	15	2^{61}	$2^{32.0}$	$2^{61.1}$	Sec. 3
	Speck96/96	28	15	2^{89}	$2^{48.0}$	$2^{89.1}$	Sec. 3
	Speck96/144	29	15	2^{89}	$2^{48.0}$	$2^{89.1}$	Sec. 3
	Speck128/128	32	17	2^{122}	$2^{64.0}$	$2^{122.1}$	Sec. 3
	Speck128/192	33	17	2^{122}	$2^{64.0}$	$2^{122.1}$	Sec. 3
	Speck128/256	34	17	2^{122}	$2^{64.0}$	$2^{122.1}$	Sec. 3
Rectangle	Speck32/64	22	11	$2^{31.1}$	$2^{33.4}$	$2^{40.7}$	Sec. 4
	Speck48/72	22	12	$2^{43.2}$	$2^{45.8}$	$2^{58.8}$	Sec. 4
	Speck48/96	23	12	$2^{43.2}$	$2^{45.8}$	$2^{58.8}$	Sec. 4
	Speck64/96	26	14	$2^{63.6}$	$2^{65.6}$	$2^{89.4}$	Sec. 4
	Speck64/128	27	14	$2^{63.6}$	$2^{65.6}$	$2^{89.4}$	Sec. 4
	Speck96/144	29	16	$2^{90.9}$	$2^{94.5}$	$2^{135.9}$	Sec. 4
	Speck128/192	33	18	$2^{121.9}$	$2^{125.9}$	$2^{182.7}$	Sec. 4
	Speck128/256	34	18	$2^{121.9}$	$2^{125.9}$	$2^{182.7}$	Sec. 4

Table 1. Summary of our results on Speck. Att. = attacked, CP = chosen plaintexts, Ref. = reference.

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. 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. As a last part, we conclude our paper in Section 5. Prior, 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	The i-th (least-significant bit) in L, where $i = 0$ denotes the least-significant bit.
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.

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 each round, the left word of the state is rotated by α bits to the left, before the right word is added modulo 2^n to that. 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 Speck32/64, which employs $\alpha = 7$ and $\beta = 2$.

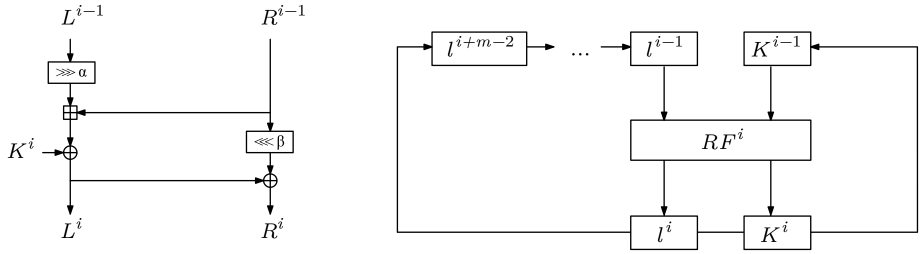

Fig. 1. Schematic views on the round function (left) and the key schedule (right) of Speck. RF_i denotes the invocation of the round function, parametrized with i as the key.

Differential Characteristics. We constructed differential characteristics for Speck by using a branch-and-bound algorithm, as inspired by the work by Alkhzaimi and Lauridsen [1]. Therefore, we started from differences with a single active bit in the middle, and generate all possible output differences after processing one round. In the following, we used the best output differences as inputs to the next round to effectively prune the search tree. While this approach can not consider all possible characteristics, it tries to regard those which are most probable. Consequently, it does not necessarily deliver the exact probabilites for our characteristics; however, it delivers useful lower bounds. Tables 5, 6, 7, 8, and 9 (see Appendix A) list our best found characteristics for the individual versions of Speck in detail.

3 Differential Attacks on Speck

In the following, we describe our 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 our attacks on the further versions of Speck, we provide only the complexities.

3.1 Key-Recovery Attack on Speck32/64

In the following, we describe in brief an 10-round key-recovery attack on Speck 32/64. There, we use the characteristic from Table 5 over rounds $2-9$:

$$\varDelta^2 = (\varDelta_{5,6,9,11}, \varDelta_{0,2,9,14}) \xleftrightarrow[\text{rounds } 2-9]{p \approx 2^{-24}} (\varDelta_{1,3,5,15}, \varDelta_{3,5,7,10,12,14,15}) = \varDelta^9.$$

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*, a *key-guessing*, and a *brute-force phase*:

Collection phase:

1. Choose 2^{28} pairs (P_i, P'_i) such that their difference after the first round is $P_i \oplus P'_i = \varDelta^2$.
2. Collect the corresponding ciphertext pairs (C_i, C'_i) from a decryption oracle, where $C_i = E_K(P_i)$ and $C'_i = E_K(P'_i)$. Derive $\varDelta L^9_{0-3}, \varDelta R^9$ and store all pairs (C_i, C'_i) with $\varDelta L^9_{0-3} = \varDelta_3$ and $\varDelta R^9 = \varDelta_{3,5,7,10,12,14,15}$ in a list $\mathcal{C}$.

Key-guessing phase:

3. Initialize a list of 2^{12} counters.
4. For all possible values of the 12 key bits K^9_{4-15}:
   - For all pairs $(C_i, C'_i) \in \mathcal{C}$:
     - Partially decrypt (C_i, C'_i) to the state after the encryption of Round 9, and derive $\varDelta L^9$. If $\varDelta L^9 = \varDelta_{1,3,5,15}$, then increment the counter for the current key candidate.
5. Output all keys as potentially correct which have a counter of at least four associated to them.
6. Mark all pairs which yielded the correct $\varDelta^9$ for the potentially correct key(s) as correct pairs.

Brute-force phase:

7. Partially decrypt all correct pairs round by round the correct subkey bits K^9_{0-3}, K^8, K^7, and K^6.

The probability that a pair follows our differential characteric is about 2^{-24}. Hence, the probability that no more than three correct pairs occur when using Speck can be approximated by

$$Pr[\textbf{false random}] := Pr^{Poisson}[n = 2^{28}, p = 2^{-24}, x \leq 3] \approx 9.31 \cdot 10^{-5}.$$

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 three false-positive pairs occur is

$$1 - Pr^{Poisson}[n = 2^{28}, p = 2^{-32}, x \leq 3] \approx 6.05 \cdot 10^{-7}.$$

Since $\mathcal{A}$ guesses 12 key bits, the probability that any key candidate produces more than three false-positive pairs is about

$$Pr[\textbf{false real}] := 1 - Pr^{Poisson}[n = 2^{12}, p = 6.05 \cdot 10^{-7}, x \leq 0] \approx 2.47 \cdot 10^{-3}.$$

Hence, the error probability of $\mathcal{A}$ is very close to 0, if it interprets a key candidate as the secret key when at least four pairs satisfy Δ^9.

Attack Complexity. Our attack requires 2^{29} chosen plaintexts. The computational effort for the collection phase, C_{collect}, is equivalent to 2^{29} full encryptions performed by an encryption oracle. The filtering effort, C_{filter}, is twofold. First, we partially decrypt all ciphertext pairs over the final round. There, we have a 20-bit filter from the four least-significant bits of ΔL^9 and the full ΔR^9. Assuming that all differences occur uniformly at random, we can say that we expect to have $2^{28-20} = 2^8$ remaining pairs afterwards. Thereupon, for 2^{12} values of K^9_{4-15}, we derive the remaining 2^8 pairs and derive ΔL^9. In the brute-force phase, the adversary then partially decrypts the remaining pairs round by round to identify the correct round keys. The computational complexity is given by

$$\underbrace{2^{29}}_{C_{\text{collect}}} + \underbrace{2^{29} \cdot \frac{1}{10} + 2^8 \cdot 2^{12} \cdot \frac{1}{10}}_{C_{\text{filter}}} + \underbrace{\left(2^4 + 2^{16} + 2^{16} + 2^{16}\right) \cdot 2^8 \cdot \frac{1}{10}}_{C_{\text{bruteforce}}} \approx 2^{29.16}$$

encryptions. 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 former option implies a lower memory complexity of 2^{12} bytes for the first filtering phase and 2^{16} bytes for the counters of the round keys in the brute-force phase.

For the further versions of Speck, we can apply a similar procedure. The parameters for these attacks are summarized in Table 2.

State size	**Key size**	**Rounds**	**Pr[diff.]**	**Pairs**	**Filter**	**Thresh. pairs**
32	64	10	$2^{-24.00}$	2^{28}	20	> 3
48	all	12	$2^{-40.55}$	2^{44}	25	> 3
64	all	15	$2^{-55.90}$	2^{60}	35	> 3
96	all	15	$2^{-84.00}$	2^{88}	54	> 3
128	all	17	$2^{-117.28}$	2^{121}	67	> 3

Table 2. Parameters of our differential attacks. Filter represents the number of bits to filter after inverting the final round with a all-zero round key.

4 Rectangle Attacks on Speck

4.1 Boomerang and Rectangle Attacks

Boomerangs [16] are differential-based attacks that allow an adversary to concatenate two "short" differential characteristics, which is beneficial for primitives where "long" characteristics may have a very low probability. Boomerang attacks have been first introduced by Wagner in 1999 [16], and were later transformed into a chosen-plaintext attack by Kelsey, Kohno, and Schneier [13], 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 [9] 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 $\frac{2^{(n+2)/2}}{pq}$ plaintext pairs and let the oracle to encrypt them. Since any two pairs can be used to form a quartet, this gives the adversary $\frac{2^{n+1}}{p^2q^2}$ possible quartets. The difference γ holds with probability 2^{-n} after E_0. Thus, one can expect a few correct quartets for which holds $C \oplus D = C' \oplus D' = \delta$.
A first improvement to the boomerang was already considered by Wagner. Instead of requiring a single difference γ at the end of E_0, the attack can be mounted with all possible values γ' for which apply that $\gamma' \to \delta$. As a second improvement, Biham et al. proposed to consider all differences β' for which applies $\alpha \to \beta'$ as long as $\beta \neq \gamma$. Hence, the probability of a correct quartet increases to $(\hat{p}\hat{q})^2$, with

$$\hat{p} = \sqrt{\sum_{\beta'} Pr^2[\alpha \to \beta']} \quad \text{and} \quad \hat{q} = \sqrt{\sum_{\gamma'} Pr^2[\gamma' \to \delta]}.$$

4.2 Rectangle Attack on Speck32/64

In the remainder of this section, we explain our rectangle attack on 11-round Speck32/64. Since our attacks on the further versions of Speck work similar, we only specify the used trails and their complexities in Table 4.
For $\alpha \to \beta'$ and $\gamma' \to \delta$, we use the following trails:

$$\alpha = (\Delta_{11,13}, \Delta_4) \xrightarrow[E_1]{\hat{p} \geq 2^{-8.01}} \beta' \text{ and } \gamma' \xrightarrow[E_2]{\hat{q} \geq 2^{-4.56}} (\Delta_{15}, \Delta_{1,3,10,15}) = \delta.$$

E_1 represents the rounds 2-6, and E_2 the rounds 7-10. Again, we can split the attacking procedure into a *collection*, a *key-guessing*, and a *brute-force phase*:

Collection phase:

1. Initialize two empty hash tables $\mathcal{C}$, $\mathcal{D}$, and a list $\mathcal{Q}$.
2. Choose $\frac{2^{(n+2)/2}}{\hat{p}\hat{q}} = \frac{2^{34/2}}{2^{-8.01}2^{-4.56}} = 2^{29.57}$ plaintext pairs (P, P') s.t. their difference after the first round is α.
3. Ask an encryption oracle for their corresponding ciphertexts (C, C') and decrypt their right word over the inverse final round to the state after Round 10, $(R^{10}, {R'}^{10})$. Store them in $\mathcal{C}$. XOR the right part of δ, to $(R^{10} \oplus \Delta_{1,3,10,15}, {R'}^{10} \oplus \Delta_{1,3,10,15})$, and store these in $\mathcal{D}$.

4. Prior, lookup if there already is an entry in $\mathcal{D}$ under the index $(R^{10} \oplus \varDelta_{1,3,10,15}, {R'}^{10} \oplus \varDelta_{1,3,10,15})$. If yes, label the existing ciphertext pair in $\mathcal{D}$ as (D, D') and store the quartet (C, C', D, D') in $\mathcal{Q}$. Since this event requires a match in 16 bits of the first, and 16 bits of the second pair, we can approximate the average number of expected quartets with $2^{2 \cdot 29.57 - 1 - 32} \approx 2^{26.14}$.

Filtering phase:

5. Initialize a table $\mathcal{K}$ of 2^{16} counters for all subkey bits in K^{10}.
6. For all possible values of the subkeys K^{10}:
   6.1 Decrypt all quartets over the final round and check whether their difference $\varDelta L^{10}$ is equal to $\varDelta_{15}$. If yes, then increment the counter for the current key candidate.
7. Output the key candidate with the maximal count in $\mathcal{K}$.

Brute-force phase:

8. Partially decrypt round by round of the remaining pairs to identify the further round keys K^9, K^8, and K^7.

Attack Complexity. The attack requires $2^{30.07}$ chosen plaintexts. The adversary has to store the corresponding ciphertexts, the remaining $2^{26.14}$ quartets and a list of 2^{16} counters for all round-key candidates. So, we can approximate the required memory by $(2^{30.07} + 4 \cdot 2^{26.14}) \cdot 32/8 + 2^{16} \approx 2^{32.4}$ bytes. The computational effort for the collection phase, C_{collect} consists of $2^{30.07}$ full encryptions performed by the oracle, and $2^{30.07}$ half-round decryptions. Additionally, the adversary needs $2^{30.07}$ memory accesses to look up potential quartets and $4 \cdot 2^{26.14}$ memory accesses in average to store the remaining quartets. To use consistent units, we overestimate a memory access by a half-round computation. In the filtering phase, it has to perform $2^{16} \cdot 4 \cdot 2^{26.14} = 2^{44.14}$ half-round decryptions to obtain the difference in the left word after Round 10. Summing up, we have

$$\underbrace{2^{30.07} + (2^{30.07} + 2^{30.07} + 4 \cdot 2^{26.14}) \cdot \frac{1}{22}}_{C_{\text{collect}}} + \underbrace{2^{44.14} \cdot \frac{1}{22}}_{C_{\text{filter}}} + \underbrace{2^{16} + 2^{16} + 2^{16}}_{C_{\text{bruteforce}}} \approx 2^{40.68}$$

encryptions. We can apply a similar procedure to mount attacks on the further versions of Speck. The parameters of our attacks with error probabilities of the adversary are summarized in Table 3. The used α- and δ-differences for our rectangle attacks on the individual versions of Speck are summarized in Table 4.

5 Conclusion

In this work, we analyzed the security of the lightweight block cipher family Speck by applying differential and rectangle 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

State	Key	Rounds				$\hat{\mathbf{p}}$	$\hat{\mathbf{q}}$	Data	Memory	Time
size	size	Full	Att.	$\mathbf{E_1}$	$\mathbf{E_2}$			(CP)	(bytes)	
32	64	22	11	5	4	$2^{-8.01}$	$2^{-4.56}$	$2^{30.1}$	$2^{32.4}$	$2^{40.7}$
48	72	22	12	5	5	$2^{-9.06}$	$2^{-9.11}$	$2^{43.2}$	$2^{45.8}$	$2^{58.8}$
48	96	23	12	5	5	$2^{-9.06}$	$2^{-9.11}$	$2^{43.2}$	$2^{45.8}$	$2^{58.8}$
64	96	26	14	6	6	$2^{-15.02}$	$2^{-14.58}$	$2^{63.6}$	$2^{65.6}$	$2^{89.4}$
64	128	27	14	6	6	$2^{-15.02}$	$2^{-14.58}$	$2^{63.6}$	$2^{65.6}$	$2^{89.4}$
96	144	29	16	7	7	$2^{-22.46}$	$2^{-19.39}$	$2^{90.9}$	$2^{94.5}$	$2^{135.9}$
128	192	33	18	8	8	$2^{-28.47}$	$2^{-28.39}$	$2^{121.9}$	$2^{125.9}$	$2^{182.7}$
128	256	34	18	8	8	$2^{-28.47}$	$2^{-28.39}$	$2^{121.9}$	$2^{125.9}$	$2^{182.7}$

Table 3. Parameters of our rectangle attacks on Speck$2n/k$. CP = chosen plaintexts.

Cipher	α	δ
Speck32/64	$(\Delta_{11,13}, \Delta_4)$	$(\Delta_{15}, \Delta_{1,3,10,15})$
Speck48/k	$(\Delta_{12,15}, \Delta_4)$	$(\Delta_{2,7,23}, \Delta_{5,7,18,23})$
Speck64/k	$(\Delta_{9,17,20}, \Delta_{1,9})$	$(\Delta_{1,14,30}, \Delta_{4,14,25,30})$
Speck96/k	$(\Delta_{9,17,23}, \Delta_{1,9,12})$	$(\Delta_{5,23,31,39,47}, \Delta_{2,5,8,23,31,34,39,45,47})$
Speck128/k	$(\Delta_{6,22,25,28,31}, \Delta_{9,14,20,62})$	$(\Delta_{2,5,8,31,50,63}, \Delta_{0,11,31,42,53,58,63})$

Table 4. Differential characteristics for our rectangle attacks.

find conventional differentials for all versions of the cipher which helped us to mount differential and boomerang attacks on these versions with up to half of the total number of rounds.

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 the cipher 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.

6 Acknowledgment

We would like to thank Ivica Nikolić for helpful comments.

References

1. Hoda A. Alkhzaimi and Martin M. Lauridsen. Cryptanalysis of the SIMON Family of Block Ciphers. Cryptology ePrint Archive, Report 2013/543, 2013. `http://`

eprint.iacr.org/.

2. 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.
3. 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/.
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 Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.
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. 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.
10. 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.
11. 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.
12. 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.
13. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In *Fast Software Encryption*, pages 75–93, 2000.
14. 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.

15. 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/`.
16. David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

A Differential Characteristics

Rd.	ΔL^i	ΔR^i	$\log_2(\mathbf{Pr})$
0	$\Delta_{5,6,9,11}$	$\Delta_{0,2,9,14}$	
1	$\Delta_{0,4,9}$	$\Delta_{2,9,11}$	-5
2	$\Delta_{11,13}$	Δ_{4}	-9
3	Δ_{6}	0	-11
4	Δ_{15}	Δ_{15}	-11
5	$\Delta_{8,15}$	$\Delta_{1,8,15}$	-12
6	Δ_{15}	$\Delta_{1,3,10,15}$	-15
7	$\Delta_{1,3,8,10,15}$	$\Delta_{5,8,10,12,15}$	-18
8	$\Delta_{1,3,5,15}$	$\Delta_{3,5,7,10,12,14,15}$	-24

Table 5. Our used differential characteristic for Speck32/64.

Rd.	ΔL^i	ΔR^i	$\log_2(\mathbf{Pr})$
0	$\Delta_{0,8,9,11,19,22}$	$\Delta_{0,3,14,16,19}$	
1	$\Delta_{1,11,12,19}$	$\Delta_{1,3,6,11,17,22}$	-7
2	$\Delta_{1,4,6,22}$	$\Delta_{9,14,20,22}$	-14.55
3	$\Delta_{9,17,23}$	$\Delta_{1,9,12}$	-19.55
4	$\Delta_{12,15}$	Δ_{4}	-23.55
5	Δ_{7}	0	-25.55
6	Δ_{23}	Δ_{23}	-25.55
7	$\Delta_{15,23}$	$\Delta_{2,15,23}$	-26.55
8	$\Delta_{2,7,23}$	$\Delta_{5,7,18,23}$	-29.55
9	$\Delta_{5,7,15}$	$\Delta_{2,5,7,8,10,15,21}$	-33.55
10	$\Delta_{2,5,8,10,15}$	$\Delta_{0,2,11,13,15,18,23}$	-40.55

Table 6. Our used differential characteristic for Speck48/k.

Rd.	ΔL^i	ΔR^i	$\log_2(\mathbf{Pr})$
0	$\Delta_{6,17,22,28}$	$\Delta_{14,17,30}$	
1	$\Delta_{9,17,20}$	$\Delta_{1,9}$	-5
2	Δ_{12}	Δ_{4}	-8
3	0	Δ_{7}	-9
4	Δ_{30}	Δ_{30}	-10
5	$\Delta_{22,30}$	$\Delta_{1,22,30}$	-12
6	$\Delta_{1,14,30}$	$\Delta_{4,14,25,30}$	-16
7	$\Delta_{4,6,7,14,22,30}$	$\Delta_{1,4,6,14,17,22,28,30}$	-22.93
8	$\Delta_{1,4,7,17,31}$	$\Delta_{9,20,25}$	-31.82
9	$\Delta_{20,23,28,31}$	$\Delta_{12,20,31}$	-36.9
10	$\Delta_{15,23,31}$	$\Delta_{2,31}$	-40.9
11	$\Delta_{2,7,15,23,31}$	$\Delta_{5,7,15,23,31}$	-44.9
12	$\Delta_{5,26}$	$\Delta_{2,5,8,10,18}$	-49.9
13	$\Delta_{2,5,8,10,29}$	$\Delta_{2,10,11,13,21,29}$	-55.9

Table 7. Our used differential characteristic for Speck64/k.

Rd.	ΔL^i	ΔR^i	$\log_2(\mathbf{Pr})$
0	$\Delta_{1,5,7,19,29,37,41,43,45}$	$\Delta_{0,11,19,21,22,29,32,33,37,41,44,45}$	
1	$\Delta_{0,19,22,32,35,44,47}$	$\Delta_{3,14,19,24,25,36,40}$	-13
2	$\Delta_{3,11,19,25,27,39}$	$\Delta_{3,6,11,17,19,22,25,28,43}$	-23
3	$\Delta_{6,22,25,28,31}$	$\Delta_{9,14,20,46}$	-33
4	$\Delta_{9,17,23}$	$\Delta_{1,9,12}$	-39
5	$\Delta_{12,15}$	Δ_{4}	-43
6	Δ_{7}	0	-45
7	Δ_{47}	Δ_{47}	-45
8	$\Delta_{39,47}$	$\Delta_{2,39,47}$	-46
9	$\Delta_{2,31,47}$	$\Delta_{5,31,42,47}$	-49
10	$\Delta_{5,23,31,39,47}$	$\Delta_{2,5,8,23,31,34,39,45,47}$	-54
11	$\Delta_{2,5,8,15,34,47}$	$\Delta_{0,11,15,26,37,42,47}$	-63
12	$\Delta_{7,11,15,37,39,45,47}$	$\Delta_{2,3,7,11,14,15,18,29,37,39,40,47}$	-72
13	$\Delta_{2,11,14,15,18,31,40}$	$\Delta_{5,6,10,11,15,17,21,31,32,42,43}$	-84

Table 8. Our used differential characteristic for Speck96/k.

Rd.	ΔL^i	ΔR^i	$\log_{\mathbf{2}}(\mathbf{Pr})$
0	$\Delta_{6,10,13,26,35,42,45,54,57,58}$	$\Delta_{2,18,29,34,35,46,50,61,62}$	
1	$\Delta_{5,27,29,35,37,49,61}$	$\Delta_{0,1,21,27,29,32,35,38,53,61}$	-13
2	$\Delta_{0,19,22,32,35,44,47}$	$\Delta_{3,14,19,24,25,36,56}$	-25.66
3	$\Delta_{3,11,19,27,33}$	$\Delta_{3,6,7,11,19,22,59}$	-35.36
4	$\Delta_{6,22,25}$	$\Delta_{9,10,14,62}$	-44.04
5	$\Delta_{9,17}$	$\Delta_{1,9,12,13}$	-49.72
6	$\Delta_{12,15,16}$	Δ_{4}	-58.72
7	Δ_{7}	Δ	-61.72
8	Δ_{63}	Δ_{63}	-61.72
9	$\Delta_{55,63}$	$\Delta_{2,55,63}$	-62.72
10	$\Delta_{2,47,63}$	$\Delta_{5,47,58,63}$	-65.72
11	$\Delta_{5,39,47,55,63}$	$\Delta_{2,5,8,39,47,50,55,61,63}$	-70.72
12	$\Delta_{2,5,8,31,50,63}$	$\Delta_{0,11,31,42,53,58,63}$	-79.72
13	$\Delta_{11,23,31,53,55,61,63}$	$\Delta_{2,3,11,14,23,31,34,45,53,55,56,63}$	-88.72
14	$\Delta_{2,11,14,31,34,47,56,63}$	$\Delta_{5,6,11,17,26,31,37,47,48,58,59,63}$	-102.31
15	$\Delta_{3,5,11,17,23,31,37,39,47,55,59,63}$	$\Delta_{2,3,5,8,9,11,14,17,20,23,29,31,34 \atop 37,39,40,47,50,51,55,59,61,62,63}$	-117.28

Table 9. Our used differential characteristic for Speck128/k.