Differential Cryptanalysis of Round-Reduced Simon and Speck

Farzaneh Abed, Eik List(✉), Stefan Lucks, and Jakob Wenzel

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

Abstract. This paper presents differential attacks on Simon and Speck, two families of lightweight block ciphers that were presented by the U.S. National Security Agency in June 2013. We describe attacks on up to slightly more than half the number of rounds. While our analysis is only of academic interest, it demonstrates the drawback of the intensive optimizations in Simon and Speck.

Keywords: Differential cryptanalysis · Block cipher · Lightweight · Simon · Speck

1 Introduction

Due to the continuously growing impact of RFID tags, smartcards, and FPGAs, cryptographic algorithms which are suitable for resource-constrained devices become more and more important. Lightweight ciphers are optimized to operate in such environments which are limited with respect to their memory, battery supply, and computing power. For these applications, hard- and software efficiency are crucial, and designing cryptographic primitives which preserve security under these constraints is a major challenge.

During the last decade, many lightweight ciphers have been developed including but not limited to HIGHT [11], KATAN [8], KLEIN [9], L-Block [16], LED [10], mCrypton [12], PRESENT [6], and PRINCE [7]. In June 2012, Beaulieu et al. from the U.S. National Security Agency (NSA) contributed to this ongoing research process with the announcement of two novel families of lightweight cipher families, called Simon and Speck [3]. Both constructions support an uncommonly large range of block sizes from 32 to 128 and key sizes from 64 to 256 bits in order to suit a variety of implementations. Simon was thereby optimized for hardware (like KATAN, LED, or PRESENT), and Speck for software implementations (such as KLEIN); though, due to immense optimizations in their round functions, both cipher families perform well in hard- *and* software.

Related Work. Due to their simple structure, Simon and Speck were already target of various cryptanalytical efforts. Alkhzaimi and Lauridsen [2] presented – parallel to our work – differential attacks on up to 16, 18, 24, 29, and 40 rounds for Simon with 32-, 48-, 64-, 96-, and 128-bit state size, respectively. In addition, the authors showed impossible-differential attacks on up to 14, 15, 16, 19, and 22

C. Cid and C. Rechberger (Eds.): FSE 2014, LNCS 8540, pp. 525–545, 2015.
DOI: 10.1007/978-3-662-46706-0_27

rounds and discussed observations regarding rotational cryptanalysis and weak keys. Alizadeh et al. [1] recently presented the best linear attacks on SIMON, with attacks on 12, 15, 19, 28, and 35 rounds.

Biryukov and Velichkov [5] followed another promising approach, where they showed differential characteristics and trails on up to 14, 15, and 21 rounds of SIMON and 9, 10, and 13 rounds of SPECK with 32-, 48-, and 64-bit state size, respectively. The authors adapted Matsui's algorithm (which can find optimal differential characteristics for S-box-based ciphers) for ARX constructions by a concept they called *highways and country roads*. They pointed out that the computation of a complete differential distribution table (DDT) is infeasible for ARX-based primitives. To overcome this challenge, the authors constructed two partial DDTs: a first one with the characteristics of highest probability (highways), and a second one with trails of slightly lower probabilities (country roads) in order to connect and/or improve their previous characteristics.

Contribution and Outline. This paper describes our differential attacks on SIMON and SPECK, which are summarized in Table 1. In what follows, Sect. 2 first reviews the necessary details of the encryption functions of SIMON and SPECK. Section 3 recaps properties of the differential propagation through their respective round functions. Section 4 follows up with a description of how we constructed differential characteristics through parts of both ciphers, and how to extend these characteristics over further rounds. We later use these characteristics for basic differential key-recovery attacks, which we explain first for SIMON in Sect. 5. Then, Sect. 6 describes our differential attacks on SPECK. Section 7 shows rectangle attacks on SPECK. We conclude this work in Sect. 8.

Notions. We follow the notions of [3], where n denotes the word size in bits, $2n$ the state size in bits, and the tuple (L^r, R^r) (the left and right parts of) a state after the encryption of Round r. Further, k represents the length of the secret key. Furthermore, $\oplus$ denotes the bit-wise XOR, $+$ the addition modulo 2^n, $\wedge$ bit-wise AND, $\vee$ bit-wise OR, and $\overline{x}$ the bit-wise inverse of x. We denote by x_i the i-th least significant bit of value x, and enumerate the bits by $x = x_{n-1}x_{n-2}\ldots x_1x_0$. Alternatively, we write values in typewriter font, i.e., $\mathtt{x}$ for hex, and $\mathtt{x}_2$ for binary values, e.g., $\mathtt{1F} = 31$ and $\mathtt{110}_2 = 6$. Concerning differences, we denote by Δ_i a difference with all bits are zero, except for the i-th (least significant) bit, and by $\Delta_{i,[j,k,\ldots]}$ a difference where the i-th bit is active and the values of the bits in square brackets are unknown. Further, we denote a differential characteristic or trail from an input difference α to an output difference β by $\alpha \rightarrow \beta$.

2 Brief Description of SIMON and SPECK

SIMON and SPECK are two simple Feistel constructions that apply a combination of rotation, XOR, and either addition (SPECK) or the logical AND (SIMON) iteratively over many rounds. The encryption process of SIMON is given in Algorithm 1, that for SPECK in Algorithm 2. Both cipher families are defined for state sizes $2n$ and key sizes k: 32/64, 48/72, 48/96, 64/96, 64/128, 96/96, 96/144, 128/128, 128/192, and 128/256.

Table 1. Summary of our results on Simon and Speck. (*) = the time complexities assume that we have two independent filtering steps (cf. Remark 1). CP = chosen plaintexts, † = attack uses chosen ciphertexts.

Cipher	Attacked Rounds	Time (*)	Data (CP)	Memory (Bytes)	Success Rate
Differential					
Simon32/64	18/32 (0.56)	$2^{46.0}$	$2^{31.2}$	$2^{15.0}$	0.63
Simon48/72 †	19/36 (0.52)	$2^{52.0}$	$2^{46.0}$	$2^{20.0}$	0.98
Simon48/96 †	19/36 (0.52)	$2^{76.0}$	$2^{46.0}$	$2^{20.0}$	0.98
Simon64/96	26/42 (0.61)	$2^{63.9}$	$2^{63.0}$	$2^{31.0}$	0.86
Simon64/128	26/44 (0.59)	$2^{94.0}$	$2^{63.0}$	$2^{31.0}$	0.86
Simon96/96	35/52 (0.67)	$2^{93.3}$	$2^{93.2}$	$2^{37.8}$	0.63
Simon96/144	35/54 (0.64)	$2^{101.1}$	$2^{93.2}$	$2^{37.8}$	0.63
Simon128/128	46/68 (0.67)	$2^{125.7}$	$2^{125.6}$	$2^{40.6}$	0.63
Simon128/192	46/69 (0.66)	$2^{142.0}$	$2^{125.6}$	$2^{40.6}$	0.63
Simon128/256	46/72 (0.63)	$2^{206.0}$	$2^{125.6}$	$2^{40.6}$	0.63
Differential					
Speck32/64	10/22 (0.45)	$2^{29.2}$	2^{29}	2^{16}	0.99
Speck48/72	12/22 (0.54)	$2^{45.3}$	2^{45}	2^{24}	0.99
Speck48/96	12/23 (0.52)	$2^{45.3}$	2^{45}	2^{24}	0.99
Speck64/96	15/26 (0.57)	$2^{61.1}$	2^{61}	2^{32}	0.99
Speck64/128	15/27 (0.55)	$2^{61.1}$	2^{61}	2^{32}	0.99
Speck96/96	15/28 (0.51)	$2^{89.1}$	2^{89}	2^{48}	0.99
Speck96/144	15/29 (0.51)	$2^{89.1}$	2^{89}	2^{48}	0.99
Speck128/128	16/32 (0.50)	$2^{111.1}$	2^{116}	2^{64}	0.99
Speck128/192	16/33 (0.48)	$2^{111.1}$	2^{116}	2^{64}	0.99
Speck128/256	16/34 (0.47)	$2^{111.1}$	2^{116}	2^{64}	0.99
Rectangle					
Speck32/64	11/22 (0.50)	$2^{46.7}$	$2^{30.1}$	$2^{37.1}$	≈ 1
Speck48/72	12/22 (0.54)	$2^{58.8}$	$2^{43.2}$	$2^{45.8}$	≈ 1
Speck48/96	12/23 (0.52)	$2^{58.8}$	$2^{43.2}$	$2^{45.8}$	≈ 1
Speck64/96	14/26 (0.53)	$2^{89.4}$	$2^{63.6}$	$2^{65.6}$	≈ 1
Speck64/128	14/27 (0.51)	$2^{89.4}$	$2^{63.6}$	$2^{65.6}$	≈ 1
Speck96/144	16/29 (0.55)	$2^{135.9}$	$2^{90.9}$	$2^{94.5}$	≈ 1
Speck128/192	18/33 (0.54)	$2^{182.7}$	$2^{125.9}$	$2^{121.9}$	≈ 1
Speck128/256	18/34 (0.52)	$2^{182.7}$	$2^{125.9}$	$2^{121.9}$	≈ 1

For Simon, $f : \{0,1\}^n \to \{0,1\}^n$ is defined as $f(x) := (x \lll 1) \wedge (x \lll 8)$. The rotation constants in Speck are $\alpha = 8$ and $\beta = 3$ for the most versions of Speck; only Speck32/64 uses $\alpha = 7$ and $\beta = 2$.

Algorithm 1. Encryption with SIMON.

Input: $(L^0, R^0) \in \{0,1\}^{2n}$
Output: $(L^r, R^r) \in \{0,1\}^{2n}$
1: **for** $i = 1, \ldots, r$ **do**
2: $\quad L^i \leftarrow R^{i-1} \oplus f(L^{i-1})$
3: $\quad L^i \leftarrow L^i \oplus K^{i-1} \oplus (L^{i-1} \lll 2)$
4: $\quad R^i \leftarrow L^{i-1}$
5: **end for**
6: **return** (L^r, R^r)

Algorithm 2. Encryption with SPECK.

Input: $(L^0, R^0) \in \{0,1\}^{2n}$
Output: $(L^r, R^r) \in \{0,1\}^{2n}$
1: **for** $i = 1, \ldots, r$ **do**
2: $\quad L^i \leftarrow (L^{i-1} \ggg \alpha) + R^{i-1} \bmod 2^n$
3: $\quad L^i \leftarrow L^i \oplus K^{i-1}$
4: $\quad R^i \leftarrow (R^{i-1} \lll \beta) \oplus L^i$
5: **end for**
6: **return** (L^r, R^r)

3 Differential Properties of SIMON and SPECK

Differential Properties for the Round Function of SPECK. For SPECK, one requires only the well-known XOR-differential probabilty of the modular addition (xdp^+), which was studied in detail by Lipmaa and Moriai [13,14].

Definition 1 (XOR-Differential Probabilty of Addition [14]). *Let α, β, γ be fixed n-bit XOR differences, and $f(x, y) = x + y \bmod n$. Then, xdp^+ is defined as the probability over all $x \in \{0,1\}^n$, such that*

$$xdp^+(\alpha, \beta \rightarrow \gamma) = 2^{-2n} \left|\{(x, y) : f(x, y) \oplus f(x \oplus \alpha, y \oplus \beta) = \gamma\}\right|.$$

Differential Properties for the Round Function of SIMON. For SIMON, one has to consider the differential probability for the round function $f(x)$. At the end of this section, we provide an algorithm that yields the set and number of all possible output differences for a fixed input difference. In the following, we explain first the differential probability (DP) of logical AND; next, we derive the DP of AND in combination with rotation, and then consider the DP of AND with rotationally dependent inputs. We follow the notation by [5].

Property 1 (Absorption of Logical AND)**.** *Let $x, x', y, y' \in \{0,1\}$ and $f(x, y) = x \wedge y$. Let $\alpha = x \oplus x', \beta = y \oplus y', \gamma = f(x, y) \oplus f(x', y')$. Then, it applies that*

$$\Pr[\alpha, \beta \rightarrow \gamma = 0] = \begin{cases} 1 & \text{if } \alpha = \beta = 0, \\ 1/2 & \text{otherwise.} \end{cases}$$

Property 1 states that the differential output of the logical AND is biased: if α and β are 0, then γ must be 0. If α and/or β is 1, there is still a probability of $1/2$ that the AND operation will cancel the active bit in the output difference.

Definition 2 (XOR-Differential Probability of AND). *Let α, β, γ be fixed n-bit XOR differences, and let $f(x, y) = x \wedge y$. The XOR-differential probability of the logical AND ($xdp^\wedge$) is the probability over all $x, y \in \{0,1\}^n$, such that*

$$xdp^\wedge(\alpha, \beta \rightarrow \gamma) = 2^{-2n} \left|\{(x, y) : f(x, y) \oplus f(x \oplus \alpha, y \oplus \beta) = \gamma\}\right|.$$

Property 2 (XOR-Differential Probability of AND). *Let α, β, γ be fixed n-bit XOR differences and $hw(\cdot)$ the hamming-weight function. Then,*

$$xdp^{\wedge}(\alpha, \beta \rightarrow \gamma) = \begin{cases} 0 & \text{if } \gamma \wedge \overline{\alpha \vee \beta} \neq 0^n, \\ 2^{-hw(\alpha \vee \beta)} & \text{otherwise.} \end{cases}$$

Property 2 transfers Property 1 from bits to n-bit differences. Only those bits that are active in α and/or β can be active in γ – each with probability 1/2. This is reflected by the term $(\alpha \vee \beta)$. If γ contains active bits at other positions, then, $\gamma \wedge \overline{\alpha \vee \beta} \neq 0^n$ and $\Pr[\alpha, \beta \rightarrow \gamma] = 0$. Otherwise, all other possible differences γ are equally possible. Thus, the term $\alpha \vee \beta$ can be interpreted as the definition of a *set* of possible output differences, i.e., one can efficiently iterate over all possible combinations of values for its active bits and will obtain all possible output differences γ.

Definition 3 (XOR-Differential Probabilty of AND with Rotations). *Let α, β, γ be fixed n-bit XOR differences, $r \in [0, n-1]$ be a fixed rotation amount, and $f(x, y) = x \wedge (y \lll r)$. Then, $xdp^{\wedge, \lll}$ is defined as the probability over all $x, y \in \{0,1\}^n$, such that*

$$xdp^{\wedge, \lll}(\alpha, \beta \rightarrow \gamma) = 2^{-2n} \left| \{(x, y) : f(x, y) \oplus f(x \oplus \alpha, y \oplus \beta) = \gamma\} \right|.$$

Since rotation and bit-wise logical AND are linear, we can derive

$$\mathrm{xdp}^{\wedge, \lll}(\alpha, \beta \rightarrow \gamma) = \begin{cases} 0 & \text{if } \gamma \wedge \overline{\alpha \vee (\beta \lll r)} \neq 0^n, \\ 2^{-hw(\alpha \vee (\beta \lll r))} & \text{otherwise.} \end{cases}$$

We can easily transform $f(x) = (x \lll s) \wedge (x \lll t)$, with $s, t \in [0, n-1], s \neq t$ into $f(x) = x \wedge (x \lll r)$ with $r = s - t \bmod n$. In the following, we also take rotationally dependent inputs into account.

Definition 4 (XOR-Differential Probabilty of AND with Rotationally Dependent Inputs). *Let α, β be fixed n-bit XOR differences, $r \in [0, n-1]$ be a fixed integer, and $f(x) = x \wedge (x \lll r)$. Then, $xdp^{x \wedge (x \lll r)}$ is defined as the probability over all $x \in \{0,1\}^n$, such that*

$$xdp^{x \wedge (x \lll r)}(\alpha \rightarrow \beta) = 2^{-n} \left| \{x : f(x) \oplus f(x \oplus \alpha) = \beta\} \right|.$$

Property 3 (Differential Propagation of $xdp^{x \wedge (x \lll r)}$). *Let α be fixed n-bit XOR difference and $r \in [0, n-1]$ be a fixed integer. Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be defined by $f(x) = x \wedge (x \lll r)$. Then, the set of possible output differences β for $xdp^{x \wedge (x \lll r)}$, can be efficiently computed in $O(n)$ as shown in Algorithm 3.*

Example: $n = 16, r = 7, \alpha = \texttt{0500}$.Let x, x' be two 16-bit values which serve as input to $f(x)$, with $x \oplus x' = \alpha$ and $\beta = f(x) \oplus f(x')$. We see that

$$\begin{aligned} \alpha = \alpha_{15} \ldots \alpha_0 &= \texttt{00000101 0000000}\ 0_2 \text{ (top)} \\ \alpha \lll r = \alpha_8 \ldots \alpha_0 \alpha_{15} \ldots \alpha_9 &= \texttt{10000000 0000001}\ 0_2 \text{ (bottom)} \\ \beta = \beta_{15} \ldots \beta_0 &= \texttt{1000010}*^1\texttt{000000}*^1 0_2 \end{aligned}$$

Algorithm 3 returns $\beta = \texttt{1000010}*^1\texttt{000000}*^1 0_2$, and $count = 3$. The star symbol $*$ denotes dependent bits and the index $*^i$, indicates pairs of bits that are related.

Algorithm 3. Given a n-bit input difference α, computes the set of possible output differences β for $f(x) = x \wedge (x \lll r)$.

Input: $\alpha \in \{0,1\}^n$ {Input difference}
Output: $\beta \in \{0,1\}^n$ {Set of all possible output differences},
  count {# of possible output differences}
  $\beta \leftarrow 0^n, count \leftarrow 0$
  for $i \leftarrow 0, \ldots, n-1$ **do**
    if $(\alpha_i \vee \alpha_{(i+r \bmod n)}) \wedge \beta_i = 0$ **then** {Bit β_i can be active}
      $\beta_i \leftarrow 1$
      $count \leftarrow count + 1$
    end if
    if $\overline{\alpha_i} \wedge \alpha_{(i-r \bmod n)} \wedge \alpha_{i+r \bmod n}$ **then** $\{\beta_i = \beta_{(i+r \bmod n)}\}$
      $\beta_i \leftarrow *^i$
      $\beta_{i+r \bmod n} \leftarrow *^i$
    end if
  end for
  return $(\beta, count)$

– β_1 depends on α_1 (top) and α_{10} (bottom), with $\alpha_1 = 0$ and $\alpha_{10} = 1$;
– β_8 depends on α_8 (top) and α_1 (bottom), with $\alpha_1 = 0$ and $\alpha_8 = 1$;

From $\alpha_1 = 0$ follows that $x_2 = x'_2$. When $x_2 = x'_2 = 0$ then $\beta_1 = \beta_8 = 0$; otherwise, when $x_2 = x'_2 = 1$, it must hold that $\beta_1 = \beta_8 = 1$. We call β_1 and β_8 *dependent bits*. Since β contains four active bits and one pair of them depends on each other, there are $2^{4-1} = 2^3$ possible output differences defined by β, namely:

$$\{0000, 0102, 0400, 0502, 8000, 8102, 8400, 8502.\},$$

Each difference can be formed by $2^{16-3} = 2^{13}$ possible pairs (x, x').

4 Search for Differential Characteristics and Differentials

During our analysis, we applied a two-step approach to find our differentials. Firstly, we employed Matsui's algorithm [15] to find some characteristics for the 32-, 48-, and 64-bit versions of Simon:

	Simon32/64	Simon48/k	Simon64/k
α	$(\Delta_5, 0)$	$(\Delta_{8,16}, \Delta_{6,14,18})$	$(\Delta_6, 0)$
β	$(\Delta_{14}, 0)$	$(\Delta_{6,14,18,22}, \Delta_{20})$	$(\Delta_{6,10,14}, \Delta_{12})$
Rounds	12	15	20
$\Pr[\alpha \rightarrow \beta]$	2^{-36}	2^{-52}	2^{-70}

Secondly, we applied a branch-and-bound search, similar to the approach of [2]. There, we started from the input difference α and propagated it round-wise in forward and backward direction. For each round, we collected all possible

output characteristics $\alpha \rightarrow \beta$ and their probability p as a tuple (β, p) in a set and used them as a starting point for the next round in a depth-first manner. Therefore, we used Algorithm 3 for Simon and a variant of the Algorithm by Lipmaa and Moriai [14] for Speck.

Since following each path is infeasible, we pruned the search tree by considering only characteristics $\alpha \rightarrow \beta$ with a probability above a chosen threshold. Therefore, we used the characteristic found with Matsui's algorithm as a reference, i.e., say Matsui's characteristic had probability $p = 2^{-q}$ after some round r, we only considered those characteristics β as input to round $r+1$ that had a probability $p \gg 2^{-q-thresh}$. We further pruned the search tree by only storing a (chosen) maximal number of characteristics.

Every time two differential characteristics lead to the same output difference β after a round, we merged them to one differential trail and added their probabilities. We emphasize that our characteristics have been found experimentally and do not necessarily represent the best possible ones. Further, note that we rely on the assumption that all possible round keys are equally probable and uniformly distributed for every round.

Extending Differential Characteristics to Attacks. A given differential can be extended by a few more rounds in a key-recovery attack for any version of Simon$2n/k$. Assume, we are given an r-round differential $(\alpha, \beta) \rightarrow (\gamma, \delta)$. Because Simon injects the subkey at the end of the rounds, the adversary itself can compute the output of $f(x)$ in the first round, choose $(\beta, \alpha \oplus f(\beta))$ as input difference and obtains an $(r+1)$-round differential with equal probability. A similar strategy can be applied at the output side. Given an output difference (γ, δ) after $(r+1)$ rounds, the difference after $(r+2)$ rounds is $(\delta \oplus f(\gamma), \gamma)$. Since the subkey in the last round of a characteristic does not affect the output difference $\delta \oplus f(\gamma)$, the adversary can compute $f(\gamma)$ itself and obtains an $(r+2)$-round differential with equal probability.

For the versions 48/72-, 64/96-, 96/144-, and 128/192-bit versions, one can append a further round by simply guessing its full subkey. The total computational effort for collecting plaintext-ciphertext pairs and testing all subkey candidates for the appended round remains significantly smaller than that for exhaustively searching the full key space. Moreover, for the 32/64-, 48/96-, 64/128-, and 128/256-bit versions, one can append another round by guessing its subkey.

5 Key-Recovery Attacks on Simon

In this section we describe a key-recovery attack on round-reduced Simon32/64. Since attacks on the further variants follow a similar procedure, we specify only their complexities at the end of this section. For Simon32/64 we use the 13-round differential characteristic with $p \approx 2^{-30.2}$ (see Table 4 in Appendix A) over the rounds $2-14$:

$$\Delta^1 = (0, \Delta_6) \rightarrow (\Delta_{14}, 0) = \Delta^{14}.$$

Note that we can choose the left part of the plaintext pairs P, P', s.t. we obtain the desired difference Δ^1 after the first round. We can append four additional rounds to the end of the cipher, where we will guess in total 18 key bits. From the obtained ciphertexts, we still know many bits from the truncated trail:

$$
\begin{aligned}
(\Delta L^{15}, \Delta R^{15}) &= (\Delta_{0,[6,15]}, \Delta_{14}),\\
(\Delta L^{16}, \Delta R^{16}) &= (\Delta_{2,[0,1,7,8,14]}, \Delta_{0,[6,15]}),\\
(\Delta L^{17}, \Delta R^{17}) &= (\Delta_{4,[0,1,2,3,6,8,9,10,15]}, \Delta_{2,[0,1,7,8,14]}),\\
(\Delta L^{18}, \Delta R^{18}) &= (\Delta_{6,[14]}, \Delta_{4,[0,1,2,3,6,8,9,10,15]}).
\end{aligned}
$$

Attack Procedure. The full attacking procedure can be split into a *collection*, a *pair-filtering*, a *key-guessing*, and a *brute-force phase*:

Collection Phase

1. Initialize an empty set $\mathcal{C} = \emptyset$.
2. Choose $2^{30.2}$ plaintext pairs (P_i, P_i'), s.t. their difference after the first round yields Δ^1.
3. Collect their corresponding ciphertext pairs (C_i, C_i') from an encryption oracle, where $C_i = E_K(P_i)$ and $C_i' = E_K(P_i')$.

Pair-Filtering Phase

4. For all ciphertext pairs, invert the final round to derive Δ^{17} and store all pairs (C_i, C_i') with the correct difference at the known bits Δ^{17} in $\mathcal{C}$. We know seven bits of ΔL^{17} and 11 bits of ΔR^{17}. Assuming the differences Δ^{17} are uniformly distributed, we can expect $2^{30.2-18} = 2^{12.2}$ pairs in average.

Key-Guessing Phase

5. Create a list of counters for all 2^{18} possible values of the round-key bits $K^{17}_{0,1,5,7-11,14,15}$, $K^{16}_{6-9,13,15}$, and $K^{15}_{9,7}$, and perform the following steps for each candidate:
   - For all pairs $(C_i, C_i') \in \mathcal{C}$:
     - Partially decrypt (C_i, C_i') to the state after the encryption of Round 14. If their difference matches Δ^{14}, increment the counter for the current key candidate.
6. Output the key candidate(s) which is/are associated to the highest counter values.

Brute-Force Phase

7. For all bits of K^{17}, K^{16}, K^{15}, and K^{14} that are not guessed in the previous steps, perform further encryptions to identify their correct values.

Attack Complexity. The attack requires $2^{31.2}$ chosen plaintexts. Regarding the memory complexity, we store $2 \cdot 2^{12.2}$ texts of 32 bits each, or $2^{15.2}$ bytes for the attack. The computational effort for the collection phase, C_{collect}, is equivalent to $2^{30.2}$ full encryptions performed by the oracle. The filtering effort, C_{filter}, is given by $2^{30.2}$ one-round decryptions to check 18 bits of Δ^{17}. The effort for the key-guessing phase, $C_{\text{key-guessing}}$, consists of decrypting the remaining pairs for each of the 2^{18} key candidates over three further rounds.

Assuming that both filtering steps of the pair-filtering and the key-guessing phase are independent from each other, we can identify the correct value of the 18 guessed key bits. A trivial brute-force search can find the correct value of the 46 remaining bits of the considered subkeys $K^{14}, K^{15}, K^{16}, K^{17}$ with about 2^{46} encryptions. The total computational complexity can be estimated by

$$\underbrace{2 \cdot 2^{30.2}}_{C_{\text{collect}}} + \underbrace{2 \cdot 2^{30.2} \cdot \frac{1}{18}}_{C_{\text{filter}}} + \underbrace{2 \cdot 2^{18} \cdot 2^{18} \cdot \frac{3}{18}}_{C_{\text{key-guessing}}} + \underbrace{2^{46}}_{C_{\text{bruteforce}}} \approx 2^{46} \text{ encryptions.}$$

Remark 1. Note that in the case that our assumption would not hold, we still have a differential that is satisfied with probability $p = 2^{-30.2}$, and a 32-bit filter at Δ^{14}. Hence, we can expect to be able to reduce the candidates of the 18 key bits we guess in the final four rounds to $2^{30.2} \cdot 2^{18} \cdot 2^{-32} = 2^{16.2}$, increasing the complexity of the brute-force step to $2^{46} \cdot 2^{16.2} = 2^{62.2}$ encryptions, which is still significantly faster than exhaustive search. In general case, the computational effort for our attacks would then be dominated by the costs for a simple exhaustive search on the remaining key space. Hence, the time complexities would then become approximately $2^k/(p \cdot 2^{2n})$ ($k = 64$, $n = 16$, $p \approx 2^{-30.2}$ for Simon32/64).

Success Rate. Since the probability of a pair to follow our differential is about $2^{-30.2}$, the probability that at least one correct pair occurs for the correct key can be approximated by

$$1 - \Pr[n = 2^{30.2}, p = 2^{-30.2}, x \leq 0] = 1 - 1/e \approx 0.632.$$

Similar Attacks on Further Versions. We can apply the same procedure to further versions of Simon. To cover one additional round, we use chosen ciphertexts in the attack on Simon48/k. Table 2 summarizes the probabilities, required number of pairs, known state bits to filter (1st filter), guessed key bits (key bits), and success rates (where false random shows the probability that no correct pair occurs during execution of the respective attack, and false real denotes the probability of a false-positive pair to occur) for each attack. The differential characteristics for the further version are illustrated in Tables 4, 5 and 6 in Appendix A.

Table 2. Parameters of our differential attacks on Simon. "1st Filter" denotes the number of bits that can be used to filter out pairs after inverting the final round; key bits = # guessed key bits; p = Probability of the used differential.

Cipher	Rounds	Pairs	1st Filter	Key Bits	Stored pairs	p	Succ. rate
Simon32/k	18	$2^{30.2}$	18	18	$2^{12.2}$	$2^{-30.2}$	0.632
Simon48/k	19	$2^{45.0}$	28	20	$2^{17.0}$	$2^{-43.0}$	0.981
Simon64/k	25	$2^{62.0}$	35	36	$2^{27.0}$	$2^{-61.0}$	0.863
Simon96/k	35	$2^{92.2}$	59	43	$2^{33.2}$	$2^{-92.2}$	0.632
Simon128/k	46	$2^{124.6}$	89	50	$2^{35.6}$	$2^{-124.6}$	0.632

6 Differential Attacks on SPECK

In this section we describe our differential analysis of SPECK. Since the small version of SPECK (SPECK32/64) allows a simple practical verification, in the following, we only discuss this version in detail. We apply the same strategy to the remaining family members of SPECK and present only their complexities at the end of this section.

6.1 Key-Recovery Attack on SPECK34/64

We use the characteristic for SPECK32/64 from Table 7 in Appendix A with $p \approx 2^{-24}$ over rounds $2-9$ to mount a 10-round attack.

$$\Delta^1 = (\Delta_{5,6,9,11}, \Delta_{0,2,9,14}) \rightarrow (\Delta_{1,3,5,15}, \Delta_{3,5,7,10,12,14,15}) = \Delta^9.$$

Attack Procedure. Again, we split the attacking procedure into a *collection*, a *key-guessing*, and a *brute-force phase*:

Collection Phase

1. Initialize an empty list $\mathcal{C} = \emptyset$.
2. Choose 2^{28} pairs (P_i, P_i') s.t. their difference after the first round is Δ^1.
3. 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 $\Delta L^9_{0-3}, \Delta R^9$ and store all pairs (C_i, C_i') with $\Delta L^9_{0-3} = \Delta_3$ and $\Delta R^9 = \Delta_{3,5,7,10,12,14,15}$ in the list $\mathcal{C}$.

Key-Guessing Phase

4. Create a list of 2^{12} counters.
5. 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 ΔL^9. If $\Delta L^9 = \Delta_{1,3,5,15}$, then increment the counter for the current key candidate.
6. Output those keys as potentially correct for which their counter has a value of at least four.
7. Mark all pairs which yielded the correct Δ^9 for the potentially correct key(s) as correct pairs.

Brute-Force Phase

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

Success Rate. The probability that a pair follows our differential characteristic is about 2^{-24}. Hence, the probability that no more than three correct pairs occur when using SPECK (i.e., the correct subkey will not be found) is about

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

and hence, the success probability of the attack approx. $1 - 9.31 \cdot 10^{-5} > 0.99$.

Table 3. Parameters of our differential attacks on Speck. "1st Filter" denotes the number of bits that can be used to filter out pairs after inverting the final round; key bits = # guessed key bits; p = Probability of the used differential.

Cipher	Rounds	Pairs	1st Filter	Key Bits	Stored pairs	p	Succ. rate
Speck32/k	10	2^{28}	20	12	2^{8}	$2^{-24.0}$	0.99
Speck48/k	12	2^{44}	25	23	2^{19}	$2^{-40.6}$	0.99
Speck64/k	15	2^{61}	35	29	2^{26}	$2^{-58.9}$	0.99
Speck96/k	15	2^{88}	54	42	2^{34}	$2^{-84.0}$	0.99
Speck128/k	16	2^{115}	67	61	2^{48}	$2^{-111.1}$	0.99

Attack Complexity. Our attack on Speck32/64 requires 2^{29} chosen plaintexts. The computational effort for C_{collect} covers 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 all differences occur uniformly at random, 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 partially decrypts the remaining pairs round by round to identify the correct round keys. Therefore, the full 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, we store a list of counters for all key candidates, which requires 2^{12} bytes for the first filtering phase and 2^{16} bytes for the counters of the round keys in the brute-force phase.

We can apply a similar procedure for the remaining versions of Speck and obtain the results of Table 3. In all cases, the computational effort is dominated by the brute-force step. The differentials for the individual versions of Speck can be found in the Tables 7, 8, 9 and 10 in Appendix A.

7 Rectangle Attacks on Speck

Boomerangs and Rectangles. Boomerangs and rectangles allow to use two short differential characteristics with high probabilities instead of a single long differential. Therefore, one first splits a given cipher E into parts $E = E^2 \circ E^1$, and searches for two differentials $\alpha \xrightarrow[E^1]{p} \beta$ and $\gamma \xrightarrow[E^2]{q} \delta$. Next, one collects

quartets of plaintexts (P, P', Q, Q') with $P \oplus P' = Q \oplus Q' = \alpha$. In the following we denote by (R, R', S, S') their encryptions after E^1 and by (C, C', D, D') their encryptions after E^2.

Each quartet has a probability of p^2 that (R, R', S, S') fulfils $R \oplus R' = S \oplus S' = \beta$. We are interested in the case when $R \oplus S = \gamma$ since then, it automatically applies that $R' \oplus S' = \gamma$. With probability q^2, a ciphertext quartet (C, C', D, D') fulfils $C \oplus D = C' \oplus D' = \delta$. In this case, we call it a *right quartet*. If an adversary collects m pairs with difference α, then, the expected number of right quartets according to [4] is:

$$m^2 \cdot 2^{-n} \cdot (pq)^2.$$

Hence, it must apply that $pq < 2^{-n/2}$ in order to mount an attack on E.

As an improvement Biham et al. proposed in [4] to use quartets with any possible difference β' and γ' in the middle, as long as both pairs in a quartet share the same difference β' and γ' after E^1. Thus, the probabilities of p and q increase to

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

In the remainder of this section, we describe in details a rectangle attack on 11 rounds of Speck32/64. Since our attacks on the further versions of Speck work similar, we only specify the used trails and their complexities in Tables 11 and 12 in Appendix B.

7.1 Rectangle Attack on Speck34/64

For the attack on Speck32/64 we use the following trails $\alpha \to \beta'$ and $\gamma' \to \delta$:

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

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 for the encryption of (P, P') and receive the corresponding ciphertext pair (C, C'). Then, partially decrypt C, C' over the final round to the state after Round 10, $(R^{10}, {R'}^{10})$ and store the result 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 them in $\mathcal{D}$.

4. Prior, lookup if there is already an entry in $\mathcal{D}$ under the index

$$(R^{10} \oplus \Delta_{1,3,10,15}, R'^{10} \oplus \Delta_{1,3,10,15}).$$

If there is, label the existing ciphertext pair in $\mathcal{D}$ as (D, D') and store the quartet (C, C', D, D') in $\mathcal{Q}$. We can build $(2^{29.57})^2/2 = 2^{58.14}$ quartets from our pairs. Since this event requires a match in 16 bits of the first, and 16 bits of the second pair, we can expect to have at average a number of $2^{58.14-32} \approx 2^{26.14}$ false positive quartets for which this condition holds. Since the probability of a right quartet is $2^{2\cdot-8.01+2\cdot-4.56} = 2^{-25.14}$, we can expect $2^{58.14-25.14} = 2^{33}$ right quartets in addition. We approximate $2^{33} + 2^{26.14} \approx 2^{33}$ hereafter.

Filtering Phase

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

Brute-Force Phase

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

Attack Complexity. The attack requires $2^{30.07}$ chosen plaintexts. We have to store the corresponding ciphertexts, the remaining 2^{33} quartets, and a list of 2^{16} counters for all round-key candidates. So, we can approximate the required memory by $(2^{30.07}+4\cdot 2^{33})\cdot 32/8+2^{16} \approx 2^{37.1}$ 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, we need $2^{30.07}$ memory accesses to look up potential quartets and about $4 \cdot 2^{33}$ 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, we have to perform $2^{16} \cdot 4 \cdot 2^{33} = 2^{51}$ half-round decryptions to obtain the difference in the left word after Round 10. Summing up, we obtain a computational effort of

$$\underbrace{2^{30.07} + (2^{30.07} + 2^{30.07} + 4 \cdot 2^{26.14}) \cdot \frac{1}{22}}_{C_{\text{collect}}} + \underbrace{2^{51.14} \cdot \frac{1}{22}}_{C_{\text{filter}}} + \underbrace{2^{16} + 2^{16} + 2^{16}}_{C_{\text{bruteforce}}} \approx 2^{46.68}$$

encryptions.

8 Discussion and Conclusion

This work presented differential attacks on round-reduced versions of the SIMON and SPECK. Furthermore, we briefly considered rectangle attacks on SPECK. We also studied rectangle attacks on SIMON and impossible-differential attacks; however, we omitted those since they did not improve our results with conventional differentials.

Our analysis can be seen as a starting point for further research on SIMON and SPECK. For SIMON, it demonstrates that up to half the number of rounds are vulnerable against differential attacks due to its highly optimizied round function. Moreover, the cipher shows a strong differential effect, i.e., there are many possible characteristics for given input and output difference.

SPECK is much closer to previous ARX designs such as ThreeFish than SIMON. However, while ThreeFish has been published four years ago, still only 1/3 of the rounds have been attacked so far, whereas the current analysis of SPECK already threatened the security of up to half of the rounds little time after publication. Moreover, any new analysis method on addition-based ARX would be a threat to both NSA constructions as well. In conclusion, we can learn from SIMON that ARX designs should incorporate additions to provide reasonably fast diffusion.

Acknowledgments. We thank all reviewers of the FSE 2014 for their helpful comments and furthermore, we would like to thank Christian Forler, Ivica Nikolić, Douglas Shors, and Vesselin Velichkov for fruitful discussions.

References

1. Alizadeh, J., Bagheri, N., Gauravaram, P., Kumar, A., Sanadhya, S.K.: Linear Cryptanalysis of Round Reduced SIMON. Cryptology ePrint Archive, Report 2013/663 (2013). http://eprint.iacr.org/
2. Alkhzaimi, H.A., Lauridsen, M.M.: Cryptanalysis of the SIMON Family of Block Ciphers. Cryptology ePrint Archive, Report 2013/543 (2013). http://eprint.iacr.org/
3. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404 (2013). http://eprint.iacr.org/
4. Biham, E., Dunkelman, O., Keller, N.: The Rectangle Attack - Rectangling the Serpent. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 340–357. Springer, Heidelberg (2001)
5. Biryukov, A., Velichkov, V.: Automatic Search for Differential Trails in ARX Ciphers (Extended Version). Cryptology ePrint Archive, Report 2013/853 (2013). http://eprint.iacr.org/
6. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)

7. Borghoff, J., et al.: PRINCE: A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In: Sako, K., Wang, X. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
8. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN: A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
9. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: A New Family of Lightweight Block Ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012)
10. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED Block Cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
11. Hong, D., et al.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
12. Lim, C.H., Korkishko, T.: mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
13. Lipmaa, H.: On Differential Properties of Pseudo-Hadamard Transform and Related Mappings. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 48–61. Springer, Heidelberg (2002)
14. Lipmaa, H., Moriai, S.: Efficient Algorithms for Computing Differential Properties of Addition. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 336–350. Springer, Heidelberg (2002)
15. Matsui, M.: On Correlation Between the Order of S-boxes and the Strength of DES. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 366–375. Springer, Heidelberg (1995)
16. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)

A Differential Characteristics for Simon and Speck

We use the following notions for the tables in this section. Each table contains at least a differential characteristic for one version of Simon or Speck. We denote by $\sum$ the total probability of the full characteristic, and by $\sum_{acc}$ the accumulated probability of all found trails from start to end difference.

Table 4. Differential characteristics for SIMON32/64 and SIMON48/k.

Rd.	SIMON32/64			SIMON48/k		
	ΔL^i	ΔR^i	$\log_2(p)$	ΔL^i	ΔR^i	$\log_2(p)$
0	0	Δ_6		$\Delta_{8,16}$	$\Delta_{6,14,18}$	
1	Δ_6	0	0	$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4
2	Δ_8	Δ_6	-2	Δ_{12}	$\Delta_{6,10,14}$	-6
3	$\Delta_{6,10}$	Δ_8	-2	$\Delta_{6,10}$	Δ_{12}	-2
4	Δ_{12}	$\Delta_{6,10}$	-4	Δ_8	$\Delta_{6,10}$	-4
5	$\Delta_{6,10,14}$	Δ_{12}	-2	Δ_6	Δ_8	-2
6	$\Delta_{0,8}$	$\Delta_{6,10,14}$	-6	0	Δ_6	-2
7	$\Delta_{2,6,14}$	$\Delta_{0,8}$	-4	Δ_6	0	0
8	Δ_4	$\Delta_{2,6,14}$	-6	Δ_8	Δ_6	-2
9	$\Delta_{2,14}$	Δ_4	-2	$\Delta_{6,10}$	Δ_8	-2
10	Δ_0	$\Delta_{2,14}$	-4	Δ_{12}	$\Delta_{6,10}$	-4
11	Δ_{14}	Δ_0	-2	$\Delta_{6,10,14}$	Δ_{12}	-2
12	0	Δ_{14}	-2	$\Delta_{8,15,16}$	$\Delta_{6,10,14}$	-6
13	Δ_{14}	0	0	$\Delta_{6,14,18}$	$\Delta_{8,15,16}$	-6
14				Δ_{20}	$\Delta_{6,14,18}$	-6
15				$\Delta_{6,14,18,22}$	Δ_{20}	-2
Σ			-36			-50
Σ_{acc}			-30.22			-43.01

Table 5. Differential characteristics for SIMON64/k.

Rd.	ΔL^i	ΔR^i	$\log_2(p)$	Rd.	ΔL^i	ΔR^i	$\log_2(p)$
0	Δ_8	$\Delta_{6,10}$		11	$\Delta_{6,14,17,18}$	$\Delta_{14,15,20}$	-6
1	Δ_6	Δ_8	-2	12	$\Delta_{8,16}$	$\Delta_{6,14,17,18}$	-8
2	0	Δ_6	-2	13	$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4
3	Δ_6	0	0	14	Δ_{12}	$\Delta_{6,10,14}$	-6
4	$\Delta_{7,8,14}$	Δ_6	-2	15	$\Delta_{6,10}$	Δ_{12}	-2
5	$\Delta_{6,10,16}$	$\Delta_{7,8,14}$	-6	16	Δ_8	$\Delta_{6,10}$	-4
6	Δ_{12}	$\Delta_{6,10,16}$	-6	17	Δ_6	Δ_8	-2
7	$\Delta_{6,10,14,16}$	Δ_{12}	-2	18	0	Δ_6	-2
8	$\Delta_{8,15,16,22}$	$\Delta_{6,10,14,16}$	-8	19	Δ_6	0	0
9	$\Delta_{6,14,18}$	$\Delta_{8,15,16,22}$	-8	20	Δ_8	Δ_6	-2
10	$\Delta_{14,15,20}$	$\Delta_{6,14,18}$	-6	21	$\Delta_{6,10}$	Δ_8	-2
Σ							-80
Σ_{acc}							-61.01

Table 6. Differential characteristics for Simon96/k and Simon128/k.

Rd.	Simon96/k			Simon128/k		
	ΔL^i	ΔR^i	$\log_2(p)$	ΔL^i	ΔR^i	$\log_2(p)$
0	Δ_{20}	$\Delta_{6,14,18,22}$		Δ_{12}	$\Delta_{6,10,14}$	
1	$\Delta_{6,14,18}$	Δ_{20}	-2	$\Delta_{6,10}$	Δ_{12}	-2
2	$\Delta_{8,16}$	$\Delta_{6,14,18}$	-6	Δ_{8}	$\Delta_{6,10}$	-4
3	$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4	Δ_{6}	Δ_{8}	-2
4	Δ_{12}	$\Delta_{6,10,14}$	-6	0	Δ_{6}	-2
5	$\Delta_{6,10}$	Δ_{12}	-2	Δ_{6}	0	0
6	Δ_{8}	$\Delta_{6,10}$	-4	Δ_{8}	Δ_{6}	-2
7	Δ_{6}	Δ_{8}	-2	$\Delta_{6,10}$	Δ_{8}	-2
8	0	Δ_{6}	-2	Δ_{12}	$\Delta_{6,10}$	-4
9	Δ_{6}	0	0	$\Delta_{6,10,14}$	Δ_{12}	-2
10	Δ_{8}	Δ_{6}	-2	$\Delta_{8,15,16}$	$\Delta_{6,10,14}$	-6
11	$\Delta_{6,10}$	Δ_{8}	-2	$\Delta_{6,14,18}$	$\Delta_{8,15,16}$	-6
12	Δ_{12}	$\Delta_{6,10}$	-4	$\Delta_{14,15,20}$	$\Delta_{6,14,18}$	-6
13	$\Delta_{6,10,14}$	Δ_{12}	-2	$\Delta_{6,14,17,18}$	$\Delta_{14,15,20}$	-6
14	$\Delta_{8,15,16}$	$\Delta_{6,10,14}$	-6	$\Delta_{8,16}$	$\Delta_{6,14,17,18}$	-8
15	$\Delta_{6,14,18}$	$\Delta_{8,15,16}$	-6	$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4

(*Continued*)

Table 6. (*Continued*)

Rd.	SIMON96/k			SIMON128/k		
	ΔL^i	ΔR^i	$\log_2(p)$	ΔL^i	ΔR^i	$\log_2(p)$
16	$\Delta_{14,15,20}$	$\Delta_{6,14,18}$	-6	Δ_{12}	$\Delta_{6,10,14}$	-6
17	$\Delta_{6,14,17,18}$	$\Delta_{14,15,20}$	-6	$\Delta_{6,10}$	Δ_{12}	-2
18	$\Delta_{8,16}$	$\Delta_{6,14,17,18}$	-8	Δ_{8}	$\Delta_{6,10}$	-4
19	$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4	Δ_{6}	Δ_{8}	-2
20	Δ_{12}	$\Delta_{6,10,14}$	-6	0	Δ_{6}	-2
21	$\Delta_{6,10}$	Δ_{12}	-2	Δ_{6}	0	0
22	Δ_{8}	$\Delta_{6,10}$	-4	Δ_{8}	Δ_{6}	-2
23	Δ_{6}	Δ_{8}	-2	$\Delta_{6,10}$	Δ_{8}	-2
24	0	Δ_{6}	-2	Δ_{12}	$\Delta_{6,10}$	-4
25	Δ_{6}	0	0	$\Delta_{6,10,14}$	Δ_{12}	-2
26	Δ_{8}	Δ_{6}	-2	$\Delta_{8,15,16}$	$\Delta_{6,10,14}$	-6
27	$\Delta_{6,10}$	Δ_{8}	-2	$\Delta_{6,14,18}$	$\Delta_{8,15,16}$	-6
28	Δ_{12}	$\Delta_{6,10}$	-4	$\Delta_{14,15,20}$	$\Delta_{6,14,18}$	-6
29	$\Delta_{6,10,14}$	Δ_{12}	-2	$\Delta_{6,14,17,18}$	$\Delta_{14,15,20}$	-6
30	$\Delta_{8,16}$	$\Delta_{6,10,14}$	-6	$\Delta_{8,16}$	$\Delta_{6,14,17,18}$	-8
31				$\Delta_{6,10,14}$	$\Delta_{8,16}$	-4
32				Δ_{12}	$\Delta_{6,10,14}$	-6
33				$\Delta_{6,10}$	Δ_{12}	-2
34				Δ_{8}	$\Delta_{6,10}$	-4
35				Δ_{6}	Δ_{8}	-2
36				0	Δ_{6}	-2
37				Δ_{6}	0	0
38				Δ_{8}	Δ_{6}	-2
39				$\Delta_{6,10}$	Δ_{8}	-2
40				Δ_{12}	$\Delta_{6,10}$	-4
41				$\Delta_{6,10,14}$	Δ_{12}	-2
Σ			-106			-144
Σ_{acc}			-92.2			-124.6

Table 7. Differential characteristics for Speck32/64 and Speck48/k.

Rd.	Speck32/64			Speck48/k		
	ΔL^i	ΔR^i	$\log_2(p)$	ΔL^i	ΔR^i	$\log_2(p)$
0	$\Delta_{5,6,9,11}$	$\Delta_{0,2,9,14}$		$\Delta_{0,8,9,11,19,22}$	$\Delta_{0,3,14,16,19}$	
1	$\Delta_{0,4,9}$	$\Delta_{2,9,11}$	-5	$\Delta_{1,11,12,19}$	$\Delta_{1,3,6,11,17,22}$	-8
2	$\Delta_{11,13}$	Δ_4	-4	$\Delta_{1,4,6,22}$	$\Delta_{9,14,20,22}$	-7
3	Δ_6	0	-2	$\Delta_{9,17,23}$	$\Delta_{1,9,12}$	-5
4	Δ_{15}	Δ_{15}	0	$\Delta_{12,15}$	Δ_4	-4
5	$\Delta_{8,15}$	$\Delta_{1,8,15}$	-1	Δ_7	0	-2
6	Δ_{15}	$\Delta_{1,3,10,15}$	-2	Δ_{23}	Δ_{23}	0
7	$\Delta_{1,3,8,10,15}$	$\Delta_{5,8,10,12,15}$	-4	$\Delta_{15,23}$	$\Delta_{2,15,23}$	-1
8	$\Delta_{1,3,5,15}$	$\Delta_{3,5,7,10,12,14,15}$	-6	$\Delta_{2,7,23}$	$\Delta_{5,7,18,23}$	-3
9	$\Delta_{3,5,7,8,15}$	$\Delta_{0,1,3,8,9,12,14,15}$	-7	$\Delta_{5,7,15}$	$\Delta_{2,5,7,8,10,15,21}$	-4
10				$\Delta_{2,5,8,10,15,23}$	$\Delta_{0,2,11,13,15,18,23}$	-7
Σ			-31			-41
Σ_{acc}			-30.99			-40.55

Table 8. Differential characteristic for Speck96/k.

Rd.	ΔL^i	ΔR^i	$\log_2(p)$
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	$\Delta_{3,11,19,25,27,39}$	$\Delta_{3,6,11,17,19,22,25,28,43}$	-10
3	$\Delta_{6,22,25,28,31}$	$\Delta_{9,14,20,46}$	-10
4	$\Delta_{9,17,23}$	$\Delta_{1,9,12}$	-6
5	$\Delta_{12,15}$	Δ_4	-4
6	Δ_7	0	-2
7	Δ_{47}	Δ_{47}	0
8	$\Delta_{39,47}$	$\Delta_{2,39,47}$	-1
9	$\Delta_{2,31,47}$	$\Delta_{5,31,42,47}$	-3
10	$\Delta_{5,23,31,39,47}$	$\Delta_{2,5,8,23,31,34,39,45,47}$	-5
11	$\Delta_{2,5,8,15,34,47}$	$\Delta_{0,11,15,26,37,42,47}$	-9
12	$\Delta_{7,11,15,37,39,45,47}$	$\Delta_{2,3,7,11,14,15,18,29,37,39,40,47}$	-9
13	$\Delta_{2,11,14,15,18,31,40}$	$\Delta_{5,6,10,11,15,17,21,31,32,42,43}$	-12
Σ			-84
Σ_{acc}			-83.98

Table 9. Differential characteristic for SPECK64/k.

Rd.	ΔL^i	ΔR^i	$\log_2(p)$	Rd.	ΔL^i	ΔR^i	$\log_2(p)$
0	$\Delta_{5,21,24,27,30}$	$\Delta_{8,13,19,29}$		7	$\Delta_{4,6,7,14,22,30}$	$\Delta_{1,4,6,14,17,22,28,30}$	-7
1	$\Delta_{8,16,22}$	$\Delta_{0,8,11}$	-6	8	$\Delta_{1,4,7,17,31}$	$\Delta_{9,20,25}$	-9
2	$\Delta_{11,14}$	Δ_{3}	-4	9	$\Delta_{20,23,28,31}$	$\Delta_{12,20,31}$	-5
3	Δ_{6}	0	-2	10	$\Delta_{15,23,31}$	$\Delta_{2,31}$	-4
4	Δ_{30}	Δ_{30}	-1	11	$\Delta_{2,7,15,23,31}$	$\Delta_{5,7,15,23,31}$	-4
5	$\Delta_{22,30}$	$\Delta_{1,22,30}$	-2	12	$\Delta_{5,26}$	$\Delta_{2,5,8,10,18}$	-5
6	$\Delta_{1,14,30}$	$\Delta_{4,14,25,30}$	-4	13	$\Delta_{2,5,8,10,29}$	$\Delta_{2,10,11,13,21,29}$	-6
Σ							-59
Σ_{acc}							-58.9

Table 10. Differential characteristic for SPECK128/k.

Rd.	ΔL^i	ΔR^i	$\log_2(p)$
0	$\Delta_{5,10,16,26,27,35,37,42,48,49,54,58,60}$	$\Delta_{2,5,18,34,37,46,49,50}$	
1	$\Delta_{5,8,19,27,29,37,40,41,49,52,61}$	$\Delta_{19,21,27,29,41,53,61}$	-16
2	$\Delta_{0,11,22,27,28,32,33,44}$	$\Delta_{11,24,27,30,33,56}$	-13
3	$\Delta_{3,11,14,19,25,27,30,33,36}$	$\Delta_{3,11,19,25,59}$	-12
4	$\Delta_{6,17,22,28}$	$\Delta_{14,17,62}$	-9
5	$\Delta_{9,17,20}$	$\Delta_{1,9}$	-5
6	Δ_{12}	Δ_{4}	-3
7	0	Δ_{7}	-1
8	Δ_{7}	$\Delta_{7,10}$	-1
9	$\Delta_{7,10,63}$	$\Delta_{7,13,63}$	-2
10	$\Delta_{2,7,13,55}$	$\Delta_{7,10,13,16,55}$	-4
11	$\Delta_{5,7,10,13,16,47,55,58,63}$	$\Delta_{5,7,19,47,55,63}$	-8
12	$\Delta_{2,7,8,19,39,50,61}$	$\Delta_{7,10,19,22,39,58,61}$	-10
13	$\Delta_{0,7,10,19,22,31,39,42,53,61,63}$	$\Delta_{7,13,19,25,31,39,53,63}$	-13
14	$\Delta_{2,7,11,13,14,19,23,25,34,39,45,55,56}$	$\Delta_{7,10,11,13,14,16,19,22,23,25,28,39,42,45,55}$	-15
Σ			-112
Σ_{acc}			-111.16

B Rectangle Attacks on Speck

Table 11. Parameters of our rectangle attacks on Speck$2n/k$. $\hat{p}$ denotes the accumulated probability of the characteristics over E_1, $\hat{q}$ the probability of characteristics over E_2. Note that we prepend one round before E_1 and append one round after E_2 in our attacks.

Cipher	Rounds			$\hat{p}$	$\hat{q}$
	Attacked	E_1	E_2		
Speck32/64	11/22	5	4	$2^{-8.01}$	$2^{-4.56}$
Speck48/72	12/22	5	5	$2^{-9.06}$	$2^{-9.11}$
Speck48/96	12/23	5	5	$2^{-9.06}$	$2^{-9.11}$
Speck64/96	14/26	6	6	$2^{-15.02}$	$2^{-14.58}$
Speck64/128	14/27	6	6	$2^{-15.02}$	$2^{-14.58}$
Speck96/144	16/29	7	7	$2^{-22.46}$	$2^{-19.39}$
Speck128/192	18/33	8	8	$2^{-28.47}$	$2^{-28.39}$
Speck128/256	18/34	8	8	$2^{-28.47}$	$2^{-28.39}$

Table 12. Differential characteristics for our rectangle attacks on the individual versions of Speck. α denotes the input differences, δ the output differences.

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	$(\Delta_{6,22,25,28,31}, \Delta_{9,14,20,62})$	$(\Delta_{2,5,8,31,50,63}, \Delta_{0,11,31,42,53,58,63})$