Cryptanalysis of the SIMON Family of Block Ciphers

Hoda A. Alkhzaimi and Martin M. Lauridsen

DTU Compute
Section for Cryptology**
Department of Applied Mathematics and Computer Science
Matematiktorvet, building 324
{hoalk, mmeh}@dtu.dk

Abstract. Recently, the U.S National Security Agency has published the specifications of two families of lightweight block ciphers, SIMON and SPECK, on ePrint [2]. The ciphers are developed with optimization towards both hardware and software in mind. While the specification paper discusses design requirements and performance of the presented lightweight ciphers thoroughly, no security assessment is given. This paper is a move towards filling that cryptanalysis gap for the SIMON family of ciphers. We present a series of observations on the presented construction that, in some cases, yield attacks, while in other cases may provide basis of further analysis by the cryptographic community. Specifically, we obtain attacks using classical- as well as truncated differentials. In the former case, we show how the smallest version of SIMON, Simon32/64, exhibits a strong differential effect.

Keywords: lightweight, block cipher, Feistel, SIMON, differential cryptanalysis, impossible differentials, rotational cryptanalysis, weak keys

1 Introduction

Lightweight cryptography is a rapidly evolving and active area of research. It is driven by the need to provide security or cryptographic measures to different applications in pervasive, ubiquitous computing environments, which are widely used on resource-constrained devices. Examples of such, include mobile phones, smart cards, RFID tags and sensor networks.

These lightweight cryptographic primitives are designed to be efficient, yet secure, when limited hardware resources are available. Consequently, the main motive for current efforts of constructing lightweight cryptographic primitives is to maintain a reasonable trade-off between security, efficient hardware performance and low overall cost, measured by a number of metrics. These metrics include, but are not limited to: area (in terms of *gate equivalences*), throughput, power/energy consumption and production cost. A number of lightweight hash functions, block ciphers and stream ciphers were developed by the research community for the purpose of obtaining a better trade-off without totally losing security. For example, lightweight block ciphers designs include, but are not limited to, HIGHT [15], ICEBERG [23], KATAN [7], KLEIN [13], LED [14], mCrypton [20], Piccolo [22], PRESENT [5], PRINCE [6], TWINE [24] and EPCBC [25].

In July 2013, the NSA publicly joined these efforts by introducing the specifications of their own highly optimized lightweight block cipher families SIMON and SPECK. In comparison to other ciphers which are currently available in the field, these families are meant to have a better performance for both hardware and software platforms with respect to area needed for a given throughput, code size and memory usage.

** The presented work is a result of a PhD summer school titled “Theoretical and Practical Topics in Resource-Efficient Cryptography” held in June 2013 at the Technical University of Denmark

The specification document of the cipher highlights several points. First, despite the fact that the SIMON family is optimized for hardware platforms and SPECK is optimized for software platforms, both families can perform well in hardware *and* software. This means that both families can be used across the full spectrum of lightweight applications. Second, both families are meant to fill the need for secure, flexible and analyzable lightweight block ciphers.

For the first point, the specification document [2] provides a lengthy description of the different performance results in hardware, for ASIC implementations, and in software, for 8-bit microcontrollers, for both SIMON and SPECK. Then, a comparison is made with lightweight implementations of KATAN [7], KLEIN [13], mCrypton [20], Piccolo [22], PRESENT [5], TWINE [24], EPCBC [25] and AES [10]. Another technical report by the NSA, prior to the specification release, presents numbers on performance [1].

For the second point, however, there are no specific cryptanalytic results nor analysis provided in the specification document, to support design rationale for either cipher family. This is with the exception of the mentioning that i) no related-key attacks are possible, and that ii) eliminating sliding properties in the key scheduling of SIMON has been considered.

Contribution This paper is a move toward providing an initial cryptanalytic research and results for the SIMON family of ciphers. A series of observations on the presented SIMON construction are made, some utilized into classical differential and truncated impossible differential attacks, while others may provide grounds for more analysis by the cryptographic community. The attacks that are presented follow simple chosen plaintext settings i.e. do not require chosen-ciphertext oracles or any known- chosen- or related-keys.

The main contributions of this paper can be summarized in three points as follows and described in Table 1:

- Differential attacks for reduced-round versions of all SIMON variants,
- Impossible differential attacks for reduced-round versions of most SIMON variants, and
- Observations regarding rotational cryptanalysis and weak key classes.

Organization This paper is organized into five main sections. Section 2 gives a compact description of the SIMON family of block ciphers, including key scheduling and cipher parameters. Section 3 discusses attacks using differential cryptanalysis, especially for Simon32/64. In Section 4 we present attacks on most variants of SIMON using impossible differentials arising from truncated differential analysis. In Section 5 we present further observations that have not led directly to attacks, but pose open and interesting research problems for further investigation. Finally, we conclude and propose additional further ideas for analysis of SIMON in Section 6.

2 General Description of SIMON

SIMON is a family of lightweight block ciphers designed by the NSA with the aim of providing a cipher of an optimal hardware performance [2]. The design of SIMON is a classical Feistel scheme, operating on two n-bit halves in each round, thus the general round block size is $2n$ bits. In the remainder of this paper, we use n to refer to half the block size of the cipher, i.e. the size of the left and right branches, respectively. Each round of SIMON applies a non-linear, non-bijective hence non-invertible function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ to the left half of the state. The output of F is added using XOR to the right half along with a round key, and the two halves are swapped. The function F is defined as

$$F(x) = ((x \lll 8) \odot (x \lll 1)) \oplus (x \lll 2)$$

Cryptanalysis	Cipher	Rounds		Data	Memory	Time
		Total	Attacked			
Differential	Simon32/64	32	16	$2^{29.481}$	2^{16}	$2^{26.481}$
	Simon48/72	36	18	$2^{46.423}$	2^{24}	$2^{43.253}$
	Simon48/96	36	18	$2^{46.423}$	2^{24}	$2^{43.253}$
	Simon64/96	42	24	$2^{62.012}$	2^{32}	$2^{58.427}$
	Simon64/128	44	24	$2^{62.012}$	2^{32}	$2^{58.427}$
	Simon96/92	52	29	$2^{87.532}$	2^{48}	$2^{83.674}$
	Simon96/144	54	29	$2^{87.532}$	2^{48}	$2^{83.674}$
	Simon128/128	68	40	$2^{124.796}$	2^{64}	$2^{120.474}$
	Simon128/192	69	40	$2^{124.796}$	2^{64}	$2^{120.474}$
	Simon128/256	72	40	$2^{124.796}$	2^{64}	$2^{120.474}$
Impossible Differential	Simon32/64	32	14	$2^{33.291}$	$2^{29.203}$	$2^{44.183}$
	Simon48/72	36	15	$2^{50.262}$	$2^{45.618}$	$2^{69.079}$
	Simon48/96	36	15	$2^{50.262}$	$2^{45.618}$	$2^{69.079}$
	Simon64/96	42	16	$2^{65.248}$	$2^{60.203}$	$2^{91.986}$
	Simon64/128	44	16	$2^{65.248}$	$2^{60.203}$	$2^{91.986}$
	Simon96/92	52	19	$2^{97.233}$	$2^{91.618}$	$2^{139.738\dagger}$
	Simon96/144	54	19	$2^{97.233}$	$2^{91.618}$	$2^{139.738}$
	Simon128/128	68	22	$2^{129.226}$	$2^{123.203}$	$2^{187.527\dagger}$
	Simon128/192	69	22	$2^{129.226}$	$2^{123.203}$	$2^{187.527}$
	Simon128/256	72	22	$2^{129.226}$	$2^{123.203}$	$2^{187.527}$

Table 1: Summery of our cryptanalytic results on SIMON. Note, that entries with a † in the complexity column indicate results which are worse than brute-force search. The parameters for impossible differentials are such, that the expected fraction of remaining keys after the attack is 1%.

where $x \lll j$ denotes left rotation of x by j positions and $\odot$ is binary AND. A single round of SIMON is depicted in Figure 1.

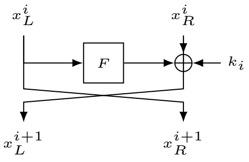


Fig. 1: The SIMON round function

Variants of SIMON exist for different parameters of key size, block size and number of rounds. The name of each SIMON variant with its parameters are presented in Table 2.

2.1 Key Schedule

The key schedule of SIMON is described as a function that will operate on two, three or four n-bit word registers, depending on the size of the master key. It performs two rotations to the right by $x \ggg 3$ and $x \ggg 1$ and XOR the results together with a fixed constant c and five constant

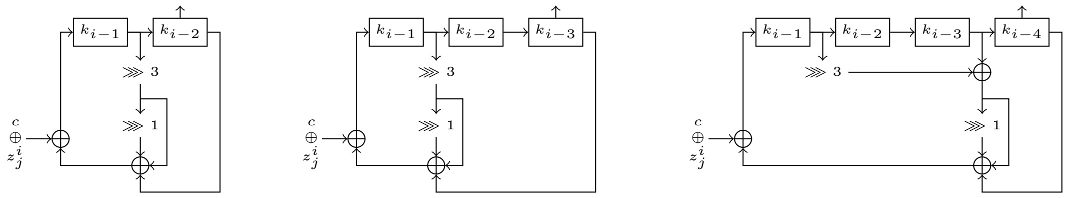
Cipher	Block size $2n$	Key words m	Key size mn	Rounds T	Index to z j
Simon32/64	32	4	64	32	0
Simon48/72	48	3	72	36	0
Simon48/96	48	4	96	36	1
Simon64/96	64	3	96	42	2
Simon64/128	64	4	128	44	3
Simon96/92	96	2	92	52	2
Simon96/144	96	3	144	54	3
Simon128/128	128	2	128	68	2
Simon128/192	128	3	192	69	3
Simon128/256	128	4	256	72	4

Table 2: Members of the SIMON family with their parameters

sequences z_j^i which are version-dependent. These constant sequences are obtained by using three 5×5 matrices over $\mathbb{F}_2$, and a linear feedback shift register where the first two are of period 31 and the last three are of period 62. The specification rationalizes the use of these constants as a mean of eliminating sliding properties and circular shift symmetries between the different rounds keys. Furthermore, they are used to provide cryptographic separation between different variants of SIMON that have the same block size, but with different key sizes.

Figure 2 describes the general function of SIMON key scheduling. The m master key words, each of n bits where $m \in \{2, 3, 4\}$, are used at the first iterations of key scheduling, and hence the first mn round key bits equal the master key.

Depending on m, the key schedule varies slightly, c.f. Figure 2. The value c is a constant equal to $(2^n - 1) \oplus 3$, i.e. a string of $n - 2$ ones and two zeroes on the least significant two bits. The value z_j^i is the ith bit (from most significant to least significant, where i is computed modulo n) of z_j, where z_j is from Table 3 and j is a parameter of the cipher, c.f. Table 2.



(a) $m = 2$ key words

(b) $m = 3$ key words

(c) $m = 4$ key words

Fig. 2: The SIMON key schedule for cases $m \in \{2, 3, 4\}$. The computation on round key k_i depends on k_{i-1} and k_{i-m}, and also k_{i-m+1} in the case of $m = 4$.

3 Differential Attack

Differential cryptanalysis is mainly a chosen plaintext attack that is considered one of the most utilized tools in achieving favourable attack results on different cryptographic primitives. It has been initially identified by the designers of Data Encryption Standard (DES) in [8] and was later

j	z_j
0	11111010001001010110000111001101111101000100101011000011100110
1	10001110111110010011000010110101000111011111001001100001011010
2	10101111011100000011010010011000101000010001111110010110110011
3	11011011101011000110010111100000010010001010011100110100001111
4	11010001111001101011011000100000010111000011001010010011101111

Table 3: The z_j vectors used in the SIMON key schedule

invented and published by Biham and Shamir in [4]. The key goal is to trace the input/output difference propagation through the cipher structure, for a specific number of rounds, and detect the non-random behaviour exhibited in the final output, with a certain success probability. The differential property can be utilized to recover the (parts of) a sub-key, typically the first or the last, in a reduced r-round version of the cipher. Several chosen plaintext pairs are used, in a combination with trying all candidates for the sub-key under attack, and the expected net result is that the correct sub-key is suggested more frequently than the wrong ones, allowing the attacker to tell which is correct.

First, we discuss iterated differentials, i.e. differentials using the same input/output difference. For the SIMON family of block ciphers, we are interested in one of two properties of F for constructing the iterated differentials. Firstly, we consider pairs of n-bit differences (a, b), for which the combined probability $\Pr(a \to b) \cdot \Pr(b \to a)$ is maximized. Here, $\Pr(a \to b)$ denotes the probability that a difference a goes to a difference b over the function F, taken over all inputs. We refer to this as a *type-1* iterated characteristic. Secondly, we may consider looking for a characteristic using a single difference a, for which $\Pr(a \to a)$ is maximized. We refer to this as a *type-2* iterated characteristic.

For type-1 characteristics, we can construct a 6-round iterative characteristic, while for type-2 we get a similar 3-round characteristic. Both are shown in Figure 3.

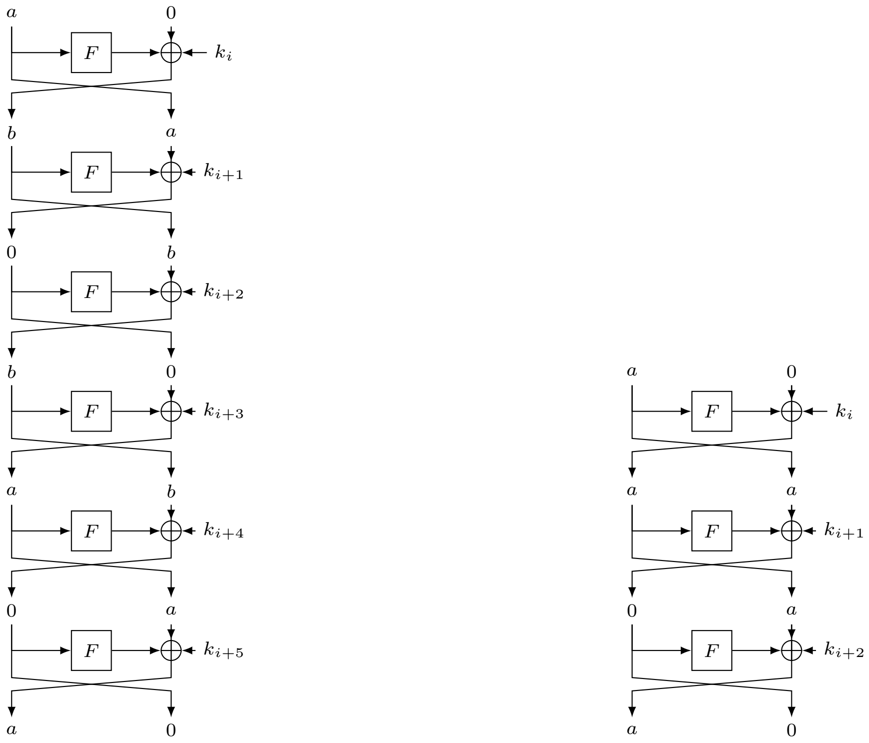
Difference Distribution Table For block ciphers using a Substitution Permutation Network (SPN) design structure, a common method for obtaining a non-linearity is to use parallel applications of small b-bit S-Boxes. In this case, the output difference on b consecutive bits depends solely on the input difference on the corresponding b bits. As such, a difference distribution table for the whole non-linear component can be derived directly from the corresponding table for the S-Box. For the function F used in SIMON there is no S-Box, and in general a single bit of the output difference Δy depends on 2 bits of the input x and 3 bits of the input difference Δx, by the relation

$$\Delta y_i = x_{i-1} \cdot \Delta x_{i-8} \oplus \Delta x_{i-1} \cdot x_{i-8} \oplus \Delta x_{i-1} \cdot \Delta x_{i-8} \oplus \Delta x_{i-2},$$

where all indices are computed modulo n. As such, constructing the difference distribution table requires $O(2^{2n})$ memory and has the same complexity. Thus, for $n = 16$, this requires 8 GB of memory using an unsigned 16-bit data type for the entries.

For $n = 16$, we construct the table exhaustively and determine the best pairs (a, b) as above for the type-1 characteristic. The best pairs (a, b) yield a probability

$$\begin{aligned} \Pr(a \to b) \cdot \Pr(b \to a) &= \frac{256}{2^{16}} \cdot \frac{2048}{2^{16}} \\ &= 2^{-13}. \end{aligned}$$



(a) A 6-round iterated characteristic using two input/output relations

(b) A 3-round iterated characteristic using a single input/output relation

Fig. 3: Type-1 and type-2 iterated differential characteristics for SIMON

If we square this probability, we find that 2^{-26} is the probability of the 6-round type-1 characteristic shown in Figure 3, using those (a, b) pairs. The pairs are listed in Table 4.

As the type-1 characteristic uses only the difference a in the input/output, we may instead think of it as a 6-round differential, where the difference b can take on any possible value. As such, we can search for the best difference a, s.t.

$$\sum_{b \in \mathbb{F}_2^n} \Pr(a \to b) \cdot \Pr(b \to a)$$

is maximized. Doing so, we find that for $n = 16$ there are four best such differences, $a \in \{\texttt{1111}, \texttt{2222}, \texttt{4444}, \texttt{8888}\}$.

These represent 3-round differentials of probability $2^{-11.19}$, where we do not care about the intermediate differences, i.e. the type-2 characteristics considered as differentials. When putting two such differentials together, we get a 6-round differential of probability at least $2^{-2 \cdot 11.19} = 2^{-22.38}$, which is similar to the type-1 characteristic considered as a differential, except that after 3 rounds we know the difference is $(a \parallel 0)$.

For $n > 16$, the memory and complexity required renders constructing the difference distribution table infeasible. However, a method by Dinur et al. which was presented at the Eurocrypt 2013 rump session [12] computes the diagonal of the difference distribution table using $O(2^n)$ memory and complexity. Thus, we can use this method to obtain results for $n = 24$ as well. The diagonal entries of the difference distribution table represent the iterative characteristics $a \to a$.

a	b	$\log_2(\Pr(a \to b))$	$\log_2(\Pr(b \to a))$
0045	051e	-5	-8
008a	0a3c	-5	-8
0114	1478	-5	-8
0228	28f0	-5	-8
028f	8022	-8	-5
0450	51e0	-5	-8
08a0	a3c0	-5	-8
1140	4781	-5	-8
1401	7814	-5	-8
1e05	4500	-8	-5
2280	8f02	-5	-8
2802	f028	-5	-8
3c0a	8a00	-8	-5
4011	8147	-5	-8
5004	e051	-5	-8
a008	c0a3	-5	-8

Table 4: Best possible (a, b) pairs for type-1 differential characteristics obtained for Simon32/64

The algorithm uses a hash table M which maps values $x \oplus F(x)$ to a list holding the x values giving this difference. M is constructed by iterating over all $x \in \mathbb{F}_2^n$. After this, any pair of distinct x, x' in the list associated with the same key in M, are values s.t. $x \oplus F(x) = x' \oplus F(x')$, or in other words, $\Delta = x \oplus x'$ is the diagonal entry under consideration. However, to compute the actual differential probability, we must again iterate over all $x \in \mathbb{F}_2^n$ and check how many times $F(x) \oplus F(x \oplus \Delta) = \Delta$.

For $n = 16$ and $n = 24$, we obtain a list of best diagonal differential probabilities, presented in Table 5.

n	p	Differences
16	2^{-8}	5555, aaaa, ac0e, 1d58, ab03, 581d, 3ab0, 6075, 5607, 0eac, b03a, 7560, c0ea, 03ab, eac0, 81d5, 0756, d581
24	2^{-12}	555555, aaaaaa, 0e22ac, 1c4558, 388ab0, 711560, c45581, e22ac0, 88ab03, 115607, 22ac0e, 45581c, ab0388, b0388a, 560711, 8ab038...

Table 5: Best diagonal entries of the difference distribution table for $n \in \{16, 24\}$

It is evident from Table 5 that already for $n = 16$, $\Pr(\Delta \to \Delta)$, for some difference Δ, is very low, and will not lead to any good differential characteristic using this method. The table *suggests* that the best probability for a diagonal entry is $2^{-n/2}$. Thus, the probability paid for such characteristic would be too low, even for two iterations of the type-2 characteristic, as the number of plaintext pairs needed for the attack would exceed the possible number of plaintext pairs, 2^{2n}.

3.1 Input/Output Differences over F

For SIMON, consider an n-bit input difference $\alpha = x \oplus x'$ to F of Hamming weight one. As the $\oplus$ operation is invariant with respect to rotation, say w.l.o.g. that $\alpha = (\mathtt{0}\cdots\mathtt{01})$. Recall that $F(x)$ left rotates x by eight and one positions respectively, applies binary AND to those two, and to the result of that XORs the left rotation of x by two positions. Due to the rotation by two and the XOR, the output difference $F(x) \oplus F(x')$ will, for this particular α, have a '1' on position 2. Also, on positions 1 and 8. There *may* be a '1' in the output difference (in fact each case occurs, on both bits independently, with probability $\frac{1}{2}$). As the $\odot$ operation is non-linear with respect to differences, this depends on the actual inputs x and x'. We may describe the output difference in truncated form as $(\mathtt{0}\cdots\mathtt{0} * \mathtt{000001} * \mathtt{0})$. Here, an asterisk denotes an unknown bit.

This approach of determining a truncated mask captures all possible output differences can be generalized to arbitrary input differences, and each time we put an asterisk on a position we lose certainty about that particular bit of the output difference. Note, that this also provides a means of determining all possible output difference, given some input difference, which in general is very useful for differential analysis. We will use this observation in the following section, and when we consider impossible differentials in Section 4.

3.2 Branch-and-Bound Approach to Differentials

Given a way of determining the possible output differences, along with their probabilities, when using a fixed input difference α, one can think of a tree where each difference at reach round spawns several possible output differences.

Besides fixing an input difference α, we fix a number of rounds to r for which we search for differentials. Starting with α, we progress in a depth-first manner, searching through characteristics until we reach round r. At that point, we add the characteristic probability to the output difference β in a lookup table. At the same time, we keep running score of the best seen output difference, for the fixed α, in terms of differential probability.

Using this approach gives us the best results on differential probabilities. Naturally, one can not hope to exhaustively try all input differences and still look through much of the tree. To that end, we maintain an array containing the best characteristic probability seen, for each level of tree, corresponding to each number of rounds $1, \ldots, r$. We bound the search at round i by allowing it only to go to round $i + 1$ if the computed characteristic probability for level $i + 1$ is within some fraction away from the best observed probability, which is stored in the array. Otherwise, we cut off that part of the tree and backtrack to the previous round. The constant fraction used in the bounding, giving the best results, is determined experimentally for each variant of SIMON. Note, that this method of cutting off sub-trees helps keep the Hamming weight of the differences low. Furthermore, we considered only input differences of low Hamming weight, as these intuitively have less possible output differences in the beginning, which are also of low Hamming weight.

As such, we can not claim to have found the best differentials for any of the variants, but our results certainly do provide lower bounds. A summary of the attack parameters and complexities can be found in Table 1.

3.3 Differential Effect

Using the branch-and-bound method described in Section 3.2, we are able, due to the small block size of Simon32/64, for a given number of rounds of the cipher to determine lower bounds on the *Expected Differential Probability* (EDP), which is defined in the following way, c.f. [11, 9]

$$\mathrm{EDP}(\alpha, \beta) = 2^{-n} \sum_{k \in \mathbb{F}_2^n} \mathrm{DP}_k(\alpha, \beta), \tag{1}$$

where $\mathrm{DP}_k(\alpha, \beta)$ is the differential probability for input difference α and output difference β using key k.

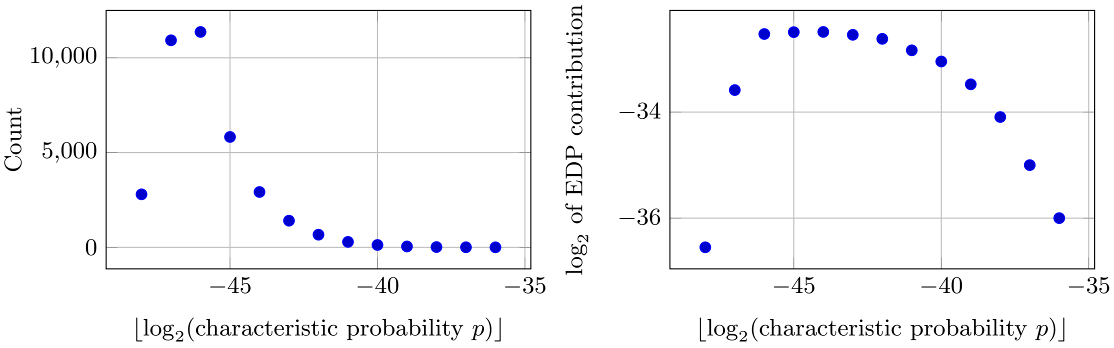
The 12-round differential leading to our 16-round differential attack on Simon32/64, as described in Table 1, is

$$\alpha \to \beta = (\texttt{0001} \parallel \texttt{0000}) \to (\texttt{0100} \parallel \texttt{0000}),$$

for which we found that $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$. The reason that the bound is not tight is twofold:

1. Firstly, due to the pruning of branches during the search, we never consider a large portion of characteristics belonging to some differential
2. Secondly, the search was, in some cases, stopped before considering all characteristics, even when using the pruning as just described, due to time limits.

An interesting question we are able to answer using the presented search method, for this small version of SIMON, is how strong the *differential effect* is. That is, we can determine if the EDP is due to the contribution of a few (or even a single) characteristics of high probability, or rather is the result of clustering of many characteristics of lower probability.



(a) Number of characteristics with a given characteristic probability p

(b) Total contribution to the EDP by characteristics of probability in $]p; 2p]$

Fig. 4: Account of the number of characteristics of a certain probability p (left) and their accumulated probability (right). The first axis is determined as $\lfloor \log_2 p \rfloor$.

For the differential $\alpha \to \beta$ of Equation (1), we keep track of the number of characteristics of probability $]p; 2p]$ in this differential by mapping $\lfloor \log_2 p \rfloor$ to a counter. We note that the search, and hence the characteristic counting, is stopped at the same point as for differential search, i.e. when obtain the bound $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$.

The resulting distribution of the number of characteristics and their probabilities are shown in Figure 4a. Figure 4b shows a division of the characteristics of probability $]p; 2p]$ on the first axis, and their total contribution to the EDP as the plotted value.

Figure 4a shows a low frequency of characteristics of probability 2^{-43} to 2^{-36}. In fact, we find just one characteristic of $\lfloor \log_2 p \rfloor = -36$ and four characteristics of $\lfloor \log_2 p \rfloor = -37$. While these few characteristics do provide an accumulated probability of $\approx 2^{-36} + 4 \cdot 2^{-37} \approx 2^{-34.42}$, the majority of the $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$, is due to the vast number of characteristics of probability p s.t. $\lfloor \log_2 p \rfloor \in [-47; -39]$. Note that there is only one characteristic of probability 2^{-36}, which is a factor of $\approx 2^{6.5}$ from the bound on $\mathrm{EDP}(\alpha, \beta)$. This might give us an indication, that the theoretical bound on the EDP, chosen initially by the designers, is based on a provable bound on the characteristic probability, which is close to the 2^{-36} for 12 rounds, as seen above.

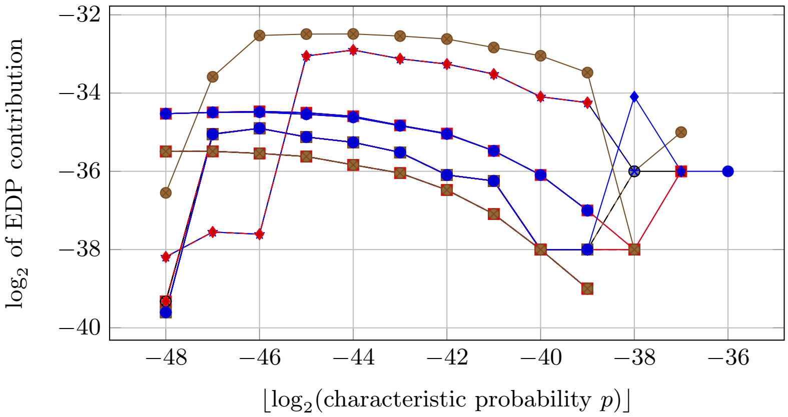


Fig. 5: Total contribution to the EDP by characteristics of probability in $]p; 2p]$, for every 12-round Simon32/64 differential found with $\mathrm{EDP}(\alpha, \beta) > 2^{-33}$. Each plot represents a single differential. Note that the plots for some differentials overlap, due to identical counts for the characteristic occurences.

In Figure 5, the same experiment is performed. However, characteristic probability frequencies for *all* differentials of $\mathrm{EDP}(\alpha, \beta) > 2^{-33}$ that we observe during our search, are collected. A total of 53 differentials were found, and in Figure 5, we clearly see the same large differential effect for all 53 cases.

Based on this observation, we conclude that, at least for Simon32/64, there is a prominent clustering of characteristics of lower probability, i.e. a strong differential effect. This might lead to a better understanding of the constraints imposed by the designers of SIMON, especially for smaller block sizes, when considering security bounds against certain attacks such as a differential attack.

3.4 Generic Extension by Two Rounds on Top

Consider an $(r-2)$-round differential property, where the desired input difference is of the form $(\alpha \parallel 0)$, i.e. an arbitrary non-zero difference on the left half of the input, and a zero difference on the right half.

As the difference is zero on the right half of the input, the corresponding input difference to F in the previous round is zero, and consequently the output difference of F is too. As such, we can extend the $(r-2)$-round property to an $(r-1)$-round property by using the input difference $(0 \parallel \alpha)$ instead.

Moreover, if we choose a plaintext $(x \parallel y)$, and set $x' = x \oplus \alpha$, then we will suffer an overhead of two applications of F. As a result, we determine the second plaintext $(x' \parallel y') = (x \oplus \alpha \parallel y \oplus F(x) \oplus F(x \oplus \alpha))$, such that the difference after one round becomes $(0 \parallel \alpha)$. Thus, after two rounds the difference is $(\alpha \parallel 0)$. This extends the $(r-2)$-round property to an r-round property without reducing the differential probability, but with the overhead of just two applications of F.

3.5 Key Recovery

When using a differential for key recovery, one would normally attack a reduced r-round version of the cipher using an $(r-1)$-round differential. However, as the round key addition is performed after the application of F in each round for SIMON, we will in fact do key recovery on an r-round version of SIMON by using an $(r-2)$-round differential. We refer to Figure 6 in our explanation of the key recovery.

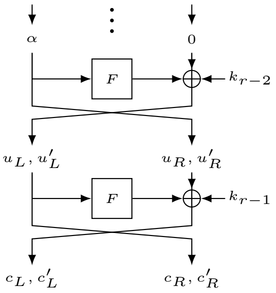


Fig. 6: Differential Key Recovery Attack on SIMON

The key recovery works as follows. We assume that the output difference of the $(r-2)$-round differential is $(\alpha \parallel 0)$. Furthermore, let an output ciphertext pair be $(c_L \parallel c_R)$ and $(c'_L \parallel c'_R)$, for which the corresponding input plaintext pair have a chosen difference dictated by the differential. We initialize a counter for each possible key guess v to zeroes.

As we can compute F, we may determine

$$u_R \oplus u'_R = F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L,$$

and check if this difference matches the difference α dictated by the differential. If this is the case, then the plaintext/ciphertext pair is assumed to follow the $(r-2)$-round differential. By trying all possible values v for the last round key, we may partially decrypt to obtain the actual pairs $(u_L \parallel u_R), (u'_L \parallel u'_R)$. Again, as we can evaluate F, we can check if

$$F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L$$

equals zero. If this is the case, then the current guess for v was considered a candidate, and a counter for the key guess v is incremented.

The process above is repeated with about $\frac{c}{p}$ chosen plaintext pairs, for some small constant c, where p is the probability of the $(r-2)$-round differential. In the end, a ranking of key candidates by their counter values provides the attacker with the most probable key guesses for the attacked last round key.

3.6 Complexity

The general complexity of the differential key recovery attacks can be expressed in terms of the following:

- Data complexity which can be defined as the number of chosen plaintexts used in the attack

- Time complexity is determined as the work effort, spent in partially decrypting the last round(s), in terms of *encryption queries*, i.e. the equivalence of r rounds of encryption
- Memory used on average for given time complexity

The data complexity of the classical differential attack can be expressed as $\frac{c}{P}$, where P is the differential probability for $r-1$ rounds, and c is a small constant. For the presented attack, it will be $2^{29.481}$ chosen pairs for 16 rounds of Simon32/64, as shown in Table 1. As for time complexity, it is defined by the number of total number of encryption queries achieved for all filtered pairs, using all possible key values:

$$\frac{c}{P} \times \gamma \times 2^k \times \frac{2}{r},$$

where r is the number of rounds, k is the number of key bits to be guessed, which are equal to n for SIMON, and γ is the probability that a pair survives the filtering which is 2^{-n}. This will yield a time complexity of $\frac{2c}{rP}r$ encryption query equivalents for SIMON variants. As for the memory needed for the key recovery attack in the presented cases, it will be the number of key guesses which is 2^n words of memory.

4 Impossible Differential Attack

Impossible differential cryptanalysis was first mentioned in 1998 by Knudsen in his analysis of DEAL [18], and further extended to an attack on IDEA by Biham et al. at FSE 1999 [3]. The approach combines two certain properties (two differentials with probability 1), one in the forward direction and one in the backward direction, and uses a resulting conflict when both directions are joined. This miss-in-the-middle approach is used to obtain an impossibility result. This can be utilized in a chosen-plaintext attack by requesting encryptions of plaintext pairs with a fixed difference, guessing key material and checking for the impossibility property to discard wrong guesses. In our case, the forward and backward differentials are truncated.

Some impossible differentials rely on the round function F being a permutation, a prominent example being the general 5-round property on Feistel schemes presented in [18]. However, the F function of SIMON is not a bijection, and indeed the impossible differentials we present in the following do not rely on it being so.

In Section 3.1, we saw how one can determine the possible output differences of the F function of SIMON, using a fixed input difference, in the sense that we can determine the truncated output difference. We also saw, that all possible output differences are equiprobable. We are interested in investigating for how many rounds a particular input difference can go before we are uncertain about all output difference bits, i.e. before we have asterisks on all positions. Intuitively, using an input difference of Hamming weight one will be the best approach, as each active bit in the input difference gives rise to 1, 2 or 3 active bits in the output difference, ignoring the possibility of cancellations, which is less predictable. For $n \in \{16, 24, 32\}$, we exhaustively tried all possible input differences and saw that this was indeed the case. For $n = 16$ and $n = 32$, there was another pattern of Hamming weight two, namely $(\mathtt{0} \cdots \mathtt{00101})$ and any rotation of it, that covered equally many rounds in one direction. However, as there was no occurrence of both 0'es and 1's in the last truncated difference, the resulting impossible differential would cover less rounds than when using a Hamming weight one input difference.

Table 6 shows how the truncated differences progress over the rounds of SIMON for some block sizes. We refer to Appendix A for the rest of the cases. All progressions use the same input difference $(\mathtt{0} \cdots \mathtt{01} \parallel \mathtt{0} \cdots \mathtt{0})$. Other Hamming weight one input differences would yield a progression of truncated differences that are rotated correspondingly.

	32-bit block	
Rounds	Left	Right
0	0000000000000001	0000000000000000
1	0000000*000001*0	0000000000000001
2	00000**00001**0*	0000000*000001*0
3	000***0*01*****0	00000**00001**0*
4	0******1******0*	000***0*01*****0
5	****************	0******1******0*

(a) For $n = 16$

	48-bit block	
Rounds	Left	Right
0	000000000000000000000001	000000000000000000000000
1	000000000000000*000001*0	000000000000000000000001
2	0000000*00000**00001**01	000000000000000*000001*0
3	00000**0000***0*01***0**	0000000*00000**00001**01
4	000***0*0**************1	00000**0000***0*01***0**
5	0***********************	000***0*0**************1
6	************************	0***********************

(b) For $n = 24$

	64-bit block	
Rounds	Left	Right
0	00000000000000000000000000000001	00000000000000000000000000000000
1	00000000000000000000000*000001*0	00000000000000000000000000000001
2	000000000000000*00000**00001**01	00000000000000000000000*000001*0
3	0000000*00000**0000***0*01***0*0	000000000000000*00000**00001**01
4	00000**0000***0*0******1******0*	0000000*00000**0000***0*01***0*0
5	000***0*0**********************0	00000**0000***0*0******1******0*
6	0*****************************0*	000***0*0**********************0
7	********************************	0*****************************0*

(c) For $n = 32$

Table 6: Truncated differential pattern propagation for SIMON using word sizes $n \in \{16, 24, 32\}$, with an input difference $(\texttt{0}\cdots\texttt{01} \parallel \texttt{0}\cdots\texttt{0})$

Taking the $n = 16$ case as an example, we see that after 5 rounds of SIMON, we have with probability 1 the truncated output difference

$$(\texttt{****************} \parallel \texttt{0******1******0*}).$$

By left rotating this right truncated difference by 7 or 9 positions, one of the 0's will be shifted to the position of the 1. Due to the symmetry of decryption and encryption of the Feistel scheme, we find that this provides us with two impossibility properties:

$$\Pr\left((\texttt{0001} \parallel \texttt{0000}) \to (\texttt{0001} \lll 7 \parallel \texttt{0000})\right) = 0 \qquad \text{and}$$
$$\Pr\left((\texttt{0001} \parallel \texttt{0000}) \to (\texttt{0001} \lll 9 \parallel \texttt{0000})\right) = 0,$$

where the impossible differential is over 10 rounds of SIMON. With this, we find two impossibility properties for each input difference of Hamming weight one, i.e. $2n$ in total. This property for the rotation by $q = 7$ is depicted in Figure 11 of Appendix A. In the further description of the attack, we denote by Q the set of indices for such rotations of the output difference, relative to the input difference, and hence $|Q|$ is the number of impossible differentials using one input difference. For example, for Simon32/64, $Q = \{7, 9\}$.

Note that the attack described so far uses an input difference of the form $(\alpha \parallel 0)$. Thus, the impossible differentials described in this section can trivially be extended by two rounds on top of probability 1, as described in Section 3.4, yielding an extra 2 rounds attacked.

Referring to Table 6, we see that for other values of n, we do not have both a 0 and 1 in the last truncated difference. Thus, we can not use this for obtaining an impossibility property, because we need to make a 0 overlap with a 1. We can, however, trace back to the last round where the truncated output difference on the right half contains a 1, and match this up with the last truncated output difference containing a 0. This sacrifice means the impossible differential covers less rounds.

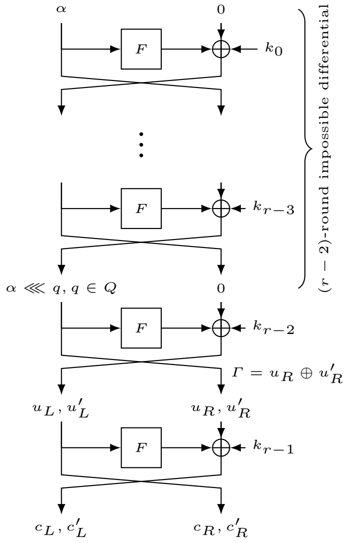


Fig. 7: Key recovery attack with impossible differentials on SIMON

4.1 Key Recovery

As it was described for key recovery using the standard differentials, we again encrypt for two rounds more than the property covers. Consider a pair of output ciphertexts $(c_L \parallel c_R)$ and $(c'_L \parallel c'_R)$. The first filter in the recovery we can apply, is to test if

$$\Gamma := F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L \tag{2}$$

equals the right half of one of the $|Q|$ impossible differentials, i.e. if it equals some $\alpha \lll q, q \in Q$.

If it does, we try all values v of the last round key and partially decrypt for one round to obtain the 1-round decrypted pair $(u_L \parallel u_R)$ and $(u'_L \parallel u'_R)$. We may now test if

$$F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L \tag{3}$$

equals 0. If it does, then v can be discarded forever as a possible last round key. The attack procedure is presented as Algorithm 1 and we refer to Figure 7 for an illustration of the attack.

Algorithm 1: Impossible differential key recovery pseudo-code for SIMON

```
   Data: $Q$ : set of rotation indices relative to input difference $\alpha$ giving
         impossible differentials
   Result: $\mathcal{K}$ : set of remaining key candidates for last round key
1  $\mathcal{K} \leftarrow \mathbb{F}_2^n$
2  Construct a "basis" of plaintexts $\mathcal{M}$ of size $2^\ell$
3  foreach $\alpha = (0\cdots01) \lll j, j = 0, \dots, n-1$ do
4      foreach $m \in \mathcal{M}$ do
5          $m' = (m'_L \parallel m'_R) \leftarrow (m_L \oplus \alpha \parallel m_R \oplus F(m_L) \oplus F(m_L \oplus \alpha))$
6          Look up $c = (c_L \parallel c_R)$ and query $c' = (c'_L \parallel c'_R) = E_K(m')$
7          $\Gamma \leftarrow F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L$
8          if $\Gamma \in \{\alpha \lll q \mid q \in Q\}$ then
9              foreach $v \in \mathcal{K}$ do
10                 A $(u_L \parallel u_R) \leftarrow (c_R \parallel F(c_R) \oplus c_L \oplus v)$
11                 $(u'_L \parallel u'_R) \leftarrow (c'_R \parallel F(c'_R) \oplus c'_L \oplus v)$
12                 if $F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L = 0$ then
13                     $\mathcal{K} \leftarrow \mathcal{K} \backslash \{v\}$
14                 end
15             end
16         end
17     end
18 end
19 return $\mathcal{K}$
```

4.2 Complexity

In the following, we give our analysis of the key recovery complexity for the impossible differential attack, in terms of data (which we define as the number of encryption oracle queries), memory and computational (time) complexity, given in terms of equivalent number of r-round encryption queries. During our analysis, we refer to the line numbers of Algorithm 1, as well as Equations (2) and (3).

As the plaintexts of the basis $\mathcal{M}$ of size 2^ℓ are queried once and stored in memory, the data and memory complexity for line 2 is 2^ℓ data and 2^ℓ memory. By choosing $\mathcal{M}$ in a way that we avoid using a particular pair twice in the form of (m, m') and (m', m), the total number of plaintext pairs used for the attack is

$$n \cdot 2^\ell,$$

where the factor n comes from the possible rotations of the input difference $\alpha = (0\cdots01) \lll j, j = 0, \dots, n-1$.

As the number of input differences we iterate over in line 3 is n, and $|\mathcal{M}| = 2^\ell$, the number of m' constructed and queried in lines 5 and 6 is $n \cdot 2^\ell$. These m' are used once and not stored in memory, hence the total memory complexity of the attack is 2^ℓ for storing $\mathcal{M}$, and the total data complexity is $2^\ell + n \cdot 2^\ell = (n+1)2^\ell$.

Expected Size of $\mathcal{K}$ When using a particular plaintext pair (m, m') with corresponding ciphertext pair (c, c') in lines 5 through 16, we first check if the difference Γ matches one of the right halves of the $|Q|$ impossible differences. Assuming that Γ is uniformly distributed with probability mass function 2^{-n}, the probability of entering the **if** statement of line 8 is

$$\frac{|Q|}{2^n},$$

and as such, the expected number of pairs passing the filtering of Equation (2) is

$$n2^\ell \cdot \frac{|Q|}{2^n}.$$

Consider now a wrong guess v for the key under attack. We know already that for the correct key, the probability of the *if statement* of line 12 being true is zero, due to the miss-in-the-middle property of the impossible differential attack. However, under the assumption that for a wrong key guess v, the difference of Equation (3) is uniformly distributed, the probability of discarding a wrong key, using a single pair, is 2^{-n}, and thus the probability of not discarding it is

$$(1 - 2^{-n}).$$

Assuming independency of the probabilities of discarding a wrong key, for each of the $n2^\ell$ pairs, the expected number of remaining keys $|\mathcal{K}|$ after using all pairs is

$$\mathbb{E}[|\mathcal{K}|] = 2^n \left(1 - 2^{-n}\right)^{n2^\ell |Q| 2^{-n}} .$$

Time Complexity For every pair used in lines 9 through 15, i.e. those pairs satisfying $\Gamma \in \{\alpha \lll q \mid q \in Q\}$, we must try as many keys as there are currently in $\mathcal{K}$. The fraction of the set $\mathcal{K}$ which is not discarded by using a single such pair equals the probability that some pair does not discard some wrong key. This probability is computed as

$$\begin{aligned}
& 1 - \Pr\left(\text{wrong key } v \text{ discarded by some pair}\right) \\
= \;& 1 - \Pr\left(\text{pair discards } v \mid \Gamma \in \{\alpha \lll q \mid q \in Q\}\right) \cdot \Pr\left(\Gamma \in \{\alpha \lll q \mid q \in Q\}\right) \\
= \;& 1 - 2^{-n} \cdot \frac{|Q|}{2^n} \\
= \;& 1 - \frac{|Q|}{2^{2n}}.
\end{aligned}$$

As such, the expected number of 1-round partial decryptions we will do during the course of the attack, using $n2^\ell$ pairs, is determined as

$$\begin{aligned}
& 2^n + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right) + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right)^2 + \cdots + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^\ell - 1} \\
= \;& 2^n \sum_{i=0}^{n2^\ell - 1} \left(1 - \frac{|Q|}{2^{2n}}\right)^i \\
= \;& 2^n \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^\ell}}{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)} \\
= \;& 2^{3n} \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^\ell}}{|Q|} && (4)
\end{aligned}$$

Evaluating this expression numerically is very computationally intensive for larger values of ℓ and n. For the numerator of Equation (4), we can use the fact that $\lim_{x \to \pm\infty} \left(1 - \frac{k}{x}\right)^x = e^{-k}$. We write 2^ℓ as $2^\ell = c2^{2n}$ for some constant c. Then

$$\begin{aligned}
\lim_{x \to \pm\infty} 2^{3n} \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^\ell}}{|Q|} &= 2^{3n} \cdot \frac{1 - e^{-|Q|nc}}{|Q|} \\
&= 2^{3n} \cdot \frac{1 - e^{-|Q|n2^{\ell - 2n}}}{|Q|}. && (5)
\end{aligned}$$

We use the approximation of Equation (5), when computing Equation (4) is too intensive. For the attack, the time complexity is determined as the total effort spent in the 1-round partial decryption phase, converted to the equivalents of r-round encryption queries. This is done, since 2^n r-round encryption queries would be the effort required to brute-force the key. As such, the total complexity in terms of r-round encryptions equals the expression from either Equation (4) or (5), multiplied by $\frac{2}{r}$. In Table 7 we present our results on key recovery attacks using impossible differentials, for all variants of SIMON, such that the expected number of remaining subkeys is 1% of the whole key space. We note that the complexities for some of the variants of SIMON are higher than brute-force effort, and hence is not considered an attack. However, as the complexities are independent of the master key size, we do have attacks on most variants.

Cipher	Rounds		$\|Q\|$	Pairs	Data	Memory	Time
	Total	Attacked		$n2^\ell$	$2^\ell + n2^\ell$	2^ℓ	
Simon32/64	32	14	2	$2^{33.203}$	$2^{33.291}$	$2^{29.203}$	$2^{44.183}$
Simon48/72	36	15	1	$2^{50.203}$	$2^{50.262}$	$2^{45.618}$	$2^{69.079\dagger}$
Simon48/96	36	15	1	$2^{50.203}$	$2^{50.262}$	$2^{45.618}$	$2^{69.079\dagger}$
Simon64/96	42	16	2	$2^{65.203}$	$2^{65.248}$	$2^{60.203}$	$2^{91.986\dagger}$
Simon64/128	44	16	2	$2^{65.203}$	$2^{65.248}$	$2^{60.203}$	$2^{91.986\dagger}$
Simon96/92	52	19	2	$2^{97.203}$	$2^{97.233}$	$2^{91.618}$	$2^{139.738\dagger}$
Simon96/144	54	19	2	$2^{97.203}$	$2^{97.233}$	$2^{91.618}$	$2^{139.738\dagger}$
Simon128/128	68	22	2	$2^{129.203}$	$2^{129.226}$	$2^{123.203}$	$2^{187.527\dagger}$
Simon128/192	69	22	2	$2^{129.203}$	$2^{129.226}$	$2^{123.203}$	$2^{187.527\dagger}$
Simon128/256	72	22	2	$2^{129.203}$	$2^{129.226}$	$2^{123.203}$	$2^{187.527\dagger}$

Table 7: Results on key recovery attack on SIMON using $|Q|\cdot n$ impossible differentials. The number of pairs used, $n2^\ell$ is determined such that the expected size of $\mathcal{K}$, i.e. the remaining key candidates, is 1% of the total subkey space 2^n. The complexities indicated with a † are computed using the approximation of Equation (5).

4.3 Practical Tests

For the case $n = 16$, the block size is small enough that we may actually implement and verify the attack. Thus, we provide in [19] among other cryptanalytic functionalities, our C++ implementation of the key-recovery attack on 14 rounds of Simon32/64, using the 12-round impossible differential.

In Table 8, we present the results of 10 experimental runs, the time for each run and the size of the output $|\mathcal{K}|$, with its corresponding percentage of the full round key space. Figure 8 shows how the size of $|\mathcal{K}|$ progressed over the course of the attack, when using difference rotation amounts on the input difference.

5 Further Observations

In this section we present other observations on SIMON, that currently have not led to immediate attacks, but are interesting topics for further analysis. Specifically, we consider SIMON from a rotational cryptanalysis perspective, and consider analysis of repeating patterns in the key schedule.

Size of $\mathcal{K}$	Time (sec.)	% of 2^n
3805	1619	5.81
789	1636	1.20
2455	1655	3.75
607	1615	0.93
1600	1634	2.44
344	1152	0.52
1536	1190	2.34
2937	1172	4.48
3170	1268	4.84
5259	1207	8.02

Table 8: Results from key recovery experiments on Simon32/64, using the parameters of Table 7. Note, that half the tests were run during the night, where the server was under less load, hence the difference in the runtimes.

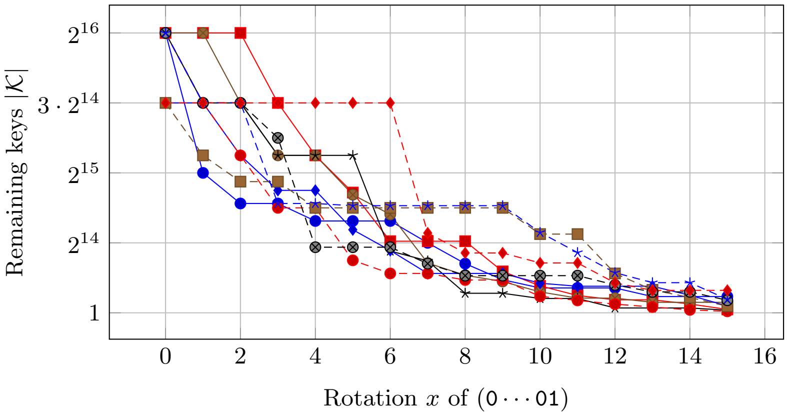


Fig. 8: Progression of the size of $|\mathcal{K}|$ for the key recovery attack on Simon32/64 using the parameters of Table 7, as a function of the rotation amount on the input difference (input difference used is $\alpha = (\texttt{0}\cdots\texttt{01}) \lll x, x = 0, \ldots, n-1$. The progressions are from the experimental results of Table 8.

5.1 Rotational Cryptanalysis

For block ciphers using round functions built around the components

- Addition modulo $2^n - 1$,
- Rotation of binary strings and
- Exclusive-or,

rotational cryptanalysis has proven efficient. Examples include cryptanalysis of Keccac by Morawiecki et al. [21] and a rotational rebound attack on Skein by Khovratovich et al. [17]. The basic idea is focused around constructing rotational pairs of words where one is the rotation of the other, for a certain rotation amount. The propagation of these pairs is traced throughout the different rounds of the primitive, knowing that rotational pairs will disclose information about specific key bits in every- or certain key words. It is worth noting that in [16], the authors state that systems with XORs and rotations can always be broken.

While the SIMON family of block ciphers does not use modular addition, it does use $\oplus$ and rotation. Thus, an interesting question is, for a random input x to the round function F, what is the probability that

$$x = F(x) \lll j, \quad 0 \leq j < n.$$

For $n \in \{16, 32\}$, the answer is given by the Table 9. We see here, that for $j = n - 2$ we get the best probabilities which is $p_{16} = \frac{2207}{2^{16}} \approx 2^{-4.89}$ for $n = 16$ and $p_{32} = (p_{16})^2 \approx 2^{-9.78}$ for $n = 32$. In Appendix B, we describe an observation in which we combine the rotational properties from Table 9 with differential cryptanalysis.

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Count	3	5	7	1	3	5	47	1	3	5	7	1	3	5	2207	1

(a) $n = 16$

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Count	3	5	7	1	3	5	47	1	3	5	7	1	3	5	8671	1
j	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Count	3	5	7	1	3	5	47	1	3	5	7	1	3	5	4870847	1

(b) $n = 32$

Table 9: Rotational approximations to $x = F(x) \lll j$ for $n \in \{16, 32\}, j = 0, \ldots, n-1$

Another observation regarding rotational cryptanalysis is that the SIMON F function is invariant under rotation, i.e.

$$\forall x \in \mathbb{F}_2^n, \forall j = 0, \ldots, n-1 : F(x \lll j) = F(x) \lll j. \tag{6}$$

Consider a plaintext pair $m = (m_L \parallel m_R)$ and

$$\begin{aligned} m' &= (m'_L \parallel m'_R) \\ &= (m_L \lll j \parallel m_R \lll j), \end{aligned}$$

with corresponding ciphertexts $c = (c_L \parallel c_R)$ and $c' = (c'_L \parallel c'_R)$. Assume that *each round key* k'_i used to encrypt m' equals k_i left rotated by j, where k_i are the round keys for encrypting m, i.e.

$$k'_i = k_i \lll j, \quad i = 0, \ldots, T-1.$$

In this case, a consequence of Equation (6) is, that

$$\begin{aligned} c'_L &= c_L \lll j, \qquad \text{and} \\ c'_R &= c_R \lll j. \end{aligned}$$

To that end, we define the following problem.

Rotational Related Key Problem Given a master key K for a variant of the SIMON block cipher on m key words of n bit each, i.e. $K \in \underbrace{\mathbb{F}_2^n \times \cdots \times F_2^n}_{m \text{ times}}$, with associated round keys k_i, $i \in \{0, \ldots, T-1\}$, determine a related key $K' \in \underbrace{\mathbb{F}_2^n \times \cdots \times F_2^n}_{m \text{ times}}$, with $K' \neq K$, s.t. for the associated round keys k'_i, it holds for all k'_i that it is as close as possible (in some measure) to $(k_i \lll j)$, for some $j = 0, \ldots, n-1$.

While solving this problem to the point where it provides ground for a cryptographic attack seems hard, it poses an interesting problem. Fixing $j = 2$, we tried for different variants of SIMON to investigate the implications on the expanded round keys, of fixing the master key K' to the rotation of K by j positions. Specifically, we generate a random K and expand the round keys. We then define K' as the rotated K, and expand those round keys. The latter expanded round keys are rotated back by j positions to the right, and we compute the length of their longest common subsequence. The result is presented in Figure 9. The figure shows how the length of the LCS rapidly converges, but whether different approaches work better is an open question.

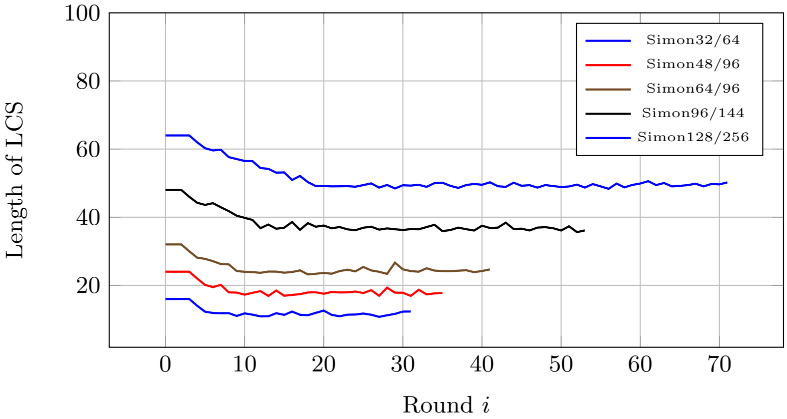


Fig. 9: Average longest common subsequence for round keys k_i and k'_i, where the master key relation is rotated by $j = 2$ positions to the left in K' compared to K

5.2 Weak Keys

The heavy constant c used in the SIMON key schedule ascertains that the weight of round keys quickly converge to an average of $\frac{n}{2}$. However, referring to the key schedule of Figure 2, and considering the case $m \neq 4$, if we can ensure that

$$(k_{i-1} \ggg 3) \oplus (k_{i-1} \ggg 4)$$

cancels many bits of the constant c, i.e.

$$c \oplus (k_{i-1} \ggg 3) \oplus (k_{i-1} \ggg 4) \tag{7}$$

has low Hamming weight, then for each of these bits, the round key k_i will equal k_{i-m}. When $m = 4$, the corresponding requirement becomes striving for a low Hamming weight in

$$c \oplus (k_{i-1} \ggg 3) \oplus k_{i-3} \oplus (((k_{i-1} \ggg 3) \oplus k_{i-3}) \ggg 1).$$

Focusing on the case $m \in \{2, 3\}$, i.e. (7), we see that the rotational difference in the XOR of k_{i-1} is a single position. Thus, for any $j \in \{2, \ldots, n-1\}$ ($j = 0$ being the least significant bit), we know that bit j of k_i will equal bit j of k_{i-m} *if and only if* bit $j+3$ and bit $j+4$ of k_{i-1} are different. For $j = 1$, we need that bit 4 and 5 of k_{i-1} are the same, and for $j = 0$ we need the XOR of bit 3 and 4 of k_{i-1} to equal z^i_j. Note, that all indices are computed modulo n. Intuitively, for each pair of consecutive alternating bits of k_{i-1}, we have one bit of k_i equal one bit of k_{i-m}.

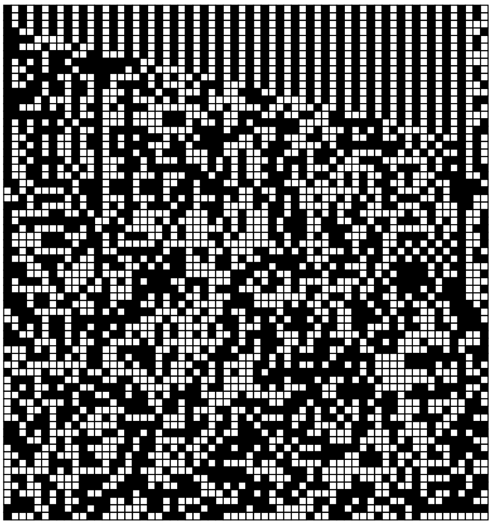


Fig. 10: Round keys (one per row) using master key ($\texttt{1010}\cdots\texttt{10} \parallel \texttt{1010}\cdots\texttt{10}$) for Simon128/128. A black square represents a 1, a white square 0.

As such, an obvious experiment is to set k_{i-1} equal to either ($\texttt{0101}\cdots\texttt{01}$) or ($\texttt{1010}\cdots\texttt{10}$). This implies that (7) will have 0's on the $n-2$ most significant bit positions. Consequently, k_i will equal k_{i-m} on those bits. If we thus choose to load all master key words with strings of this pattern, we expect that the pattern will repeat in some round keys. The resulting round keys for this experiment with the Simon128/128 cipher is depicted in Figure 10.

While this observation does not suggest any weakness in the cipher itself, it is an open question whether (parts of) this round key pattern can be combined with a property exploiting it for some number of rounds, and weak key space. In particular, whether a trade-off can be defined between the following is still an open and interesting question:

- The size of a weak key space where round keys repeat partially in a particular pattern,
- The extent of the key repetition, and
- The number of rounds where this exploitable property lasts

6 Conclusion

In this work, we have considered the lightweight block cipher family SIMON from a cryptanalytic perspective. The specification paper provided by the NSA does not include any form of security assessment, and with our analysis we have taken the first step towards an openly available cryptanalysis of the cipher family SIMON, using various commonly applied cryptanalytic techniques as summarized in Table 1. All our cryptanalysis uses the simple assumption of a chosen plaintext setting, i.e. no chosen ciphertext oracle is required, nor is any known- chosen- or related-key settings used.

We have determined iterative differentials for Simon32/64, and general differentials for all variants of SIMON, that yield differential attacks on reduced versions with at least half the total rounds of the cipher in all cases. This analysis provided the grounds for our best results. An interesting observation in Section 3.3 is that Simon32/64 exhibits a strong differential effect. This suggests that bounding the expected differential probability (EDP) by the expected maximum characteristic probability is not well-founded in this case.

Furthermore, we considered using truncated differentials to construct impossible differentials over a number of rounds, which yielded attacks on reduced versions of most of the cipher variants, however not stronger than the ordinary differential attacks.

Finally, we provided a number of other observations on the cipher structure with regards to rotational cryptanalysis and possible weak keys patterns. This may provide starting ground for future analysis in this direction.

Open Question and Future Work The analysis provided on SIMON variants has shed light on the differential properties of the cipher family. However, it is worth exploring the possibility of improving these differential/impossible differential attacks. Also, cryptanalysis with respect to higher-order differential attacks and meet-in-the-middle attacks would be interesting. In addition to that, it can be profitable to consider the effect of linear cryptanalysis on SIMON, in particular zero-correlation attacks. Furthermore, an analysis of the key schedule and whether one can obtain related-key properties that can be exploited in a combination with rotational cryptanalysis, is an interesting open question.

7 Acknowledgement

The authors would like to thank Andrey Bogdanov, Christian Rechberger and Lars R. Knudsen for their helpful and constant remarks, guidance, explanations, discussions and reviews throughout this work. Also, we would like to thank Elias Yarrkov for discussing iterated differentials, and Tyge Tiessen for discussing differential properties of the SIMON F function.

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, 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, Alex Biryukov, and Adi Shamir. Miss in the Middle Attacks on IDEA and Khufu. In *FSE*, pages 124–138, 1999.
4. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In AlfredJ. Menezes and ScottA. Vanstone, editors, *Advances in Cryptology-CRYPT0' 90*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer Berlin Heidelberg, 1991.
5. A. Bogdanov, L. R. Knudsen, G. Le, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *the proceedings of CHES 2007*. Springer, 2007.

6. Julia Borghoff, Anne Canteaut, Tim Güneysu, ElifBilge Kavun, Miroslav Knezevic, LarsR. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, SørenS. Thomsen, and Tolga Yalçın. Prince – a low-latency block cipher for pervasive computing applications. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer Berlin Heidelberg, 2012.
7. Christophe Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN — A Family of Small and Efficient Hardware-Oriented Block Ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer Berlin Heidelberg, 2009.
8. D. Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. *IBM Journal of Research and Development*, 38(3):243–250, 1994.
9. Joan Daemen, Mario Lamberger, Norbert Pramstaller, Vincent Rijmen, and Frederik Vercauteren. Computational aspects of the expected differential probability of 4-round aes and aes-like ciphers. *Computing*, 85(1-2):85–104, 2009.
10. Joan Daemen and Vincent Rijmen. The Block Cipher Rijndael. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications*, volume 1820 of *Lecture Notes in Computer Science*, pages 277–284. Springer Berlin Heidelberg, 2000.
11. Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers, 2005.
12. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. How to Efficiently and Simultaneously Compute the Probabilities of All Iterative Characteristics. Eurocrypt 2013 Rump Session, 2013.
13. Zheng Gong, Svetla Nikova, and YeeWei Law. KLEIN: A New Family of Lightweight Block Ciphers. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg.
14. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The LED Block Cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer Berlin Heidelberg, 2011.
15. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang 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, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer Berlin Heidelberg, 2006.
16. Dmitry Khovratovich and Ivica Nikolic. Rotational Cryptanalysis of ARX. In Seokhie Hong and Tetsu Iwata, editors, *FSE*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, 2010.
17. Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational Rebound Attacks on Reduced Skein. In *ASIACRYPT*, pages 1–19, 2010.
18. Lars R. Knudsen. DEAL - A 128-bit Block Cipher, 1998.
19. Martin M. Lauridsen and Hoda A. Alkhzaimi. SIMON and SPECK cryptanalysis code repository. https://github.com/mmeh/simon-speck-cryptanalysis.
20. ChaeHoon Lim and Tymur Korkishko. mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In Joo-Seok Song, Taekyoung Kwon, and Moti Yung, editors, *Information Security Applications*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer Berlin Heidelberg, 2006.
21. Pawel Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced Keccak. Cryptology ePrint Archive, Report 2012/546, 2012. http://eprint.iacr.org/.
22. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An Ultra-Lightweight Blockcipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 342–357. Springer Berlin Heidelberg, 2011.
23. Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG : An Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 279–298. Springer Berlin Heidelberg, 2004.

24. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A Lightweight Block Cipher for Multiple Platforms. In LarsR. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer Berlin Heidelberg, 2013.
25. Huihui Yap, Khoongming Khoo, Axel Poschmann, and Matt Henricksen. EPCBC - A Block Cipher Suitable for Electronic Product Code Encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *Cryptology and Network Security*, volume 7092 of *Lecture Notes in Computer Science*, pages 76–97. Springer Berlin Heidelberg, 2011.

A Addenda to Impossible Differentials Cryptanalysis

	96-bit block	
Rounds	Left	Right
0	000000000000000000000000000000000000000000000001	000000000000000000000000000000000000000000000000
1	000000000000000000000000000000000000000*000001*0	000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000*00000**00001**01	000000000000000000000000000000000000000*000001*0
3	00000000000000000000000*00000**0000***0*01***0*0	0000000000000000000000000000000*00000**00001**01
4	000000000000000*00000**0000***0*0******1******01	00000000000000000000000*00000**0000***0*01***0*0
5	0000000*00000**0000***0*0********************1*0	000000000000000*00000**0000***0*0******1******01
6	00000**0000***0*0*****************************0*	0000000*00000**0000***0*0********************1*0
7	000***0*0**************************************0	00000**0000***0*0*****************************0*
8	0***0*	000***0*0**************************************0
9	**	0***0*

(a) For $n = 48$

	128-bit block	
Rounds	Left	Right
0	0000000000000000000000000000000000000000000000000000000000000001	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000*000001*0	0000000000000000000000000000000000000000000000000000000000000001
2	00000000000000000000000000000000000000000000000*00000**00001**01	0000000000000000000000000000000000000000000000000000000*000001*0
3	000000000000000000000000000000000000000*00000**0000***0*01***0*0	00000000000000000000000000000000000000000000000*00000**00001**01
4	0000000000000000000000000000000*00000**0000***0*0******1******01	000000000000000000000000000000000000000*00000**0000***0*01***0*0
5	00000000000000000000000*00000**0000***0*0********************1*0	0000000000000000000000000000000*00000**0000***0*0******1******01
6	000000000000000*00000**0000***0*0*****************************01	00000000000000000000000*00000**0000***0*0********************1*0
7	0000000*00000**0000***0*0************************************0*0	000000000000000*00000**0000***0*0*****************************01
8	00000**0000***0*0***0*	0000000*00000**0000***0*0************************************0*0
9	000***0*0**0	00000**0000***0*0***0*
10	0***0*	000***0*0**0
11	**	0***0*

(b) For $n = 64$

Table 10: Truncated differential pattern propagation for SIMON using word sizes $n \in \{48, 64\}$, with an input difference $(\mathtt{0} \cdots \mathtt{01})$ on the left half and a 0-difference on the right half

B Combining Rotational and Differential Cryptanalysis

Referring to Table 9, we saw that $\Pr(x = F(x) \lll j)$ is maximized when $j = 2$. This probability equals $\Pr(F(x) = x \ggg 2)$. Now consider a plaintext pair $(m_L \parallel m_R)$ and $(m'_L \parallel m'_R)$.

If $m'_L = m_R \lll 2$ and $m'_R = m_L \ggg 2$, then in the first round, we have that $F(m_L) = m_L \ggg 2$ and $F(m'_L) = F(m_R \lll 2) = m_R$ with a certain probability. If these are both fulfilled, then the difference on the left block after 1 round ends up being $(m_L \ggg 2) \oplus m_R \oplus k \oplus (m_L \ggg 2) \oplus m_R \oplus k = 0$, and on the right block it will be $m_L \oplus (m_R \lll 2)$. As the input difference is 0 to F in round 2, we find that the output difference on the left block is $m_L \oplus (m_R \lll 2)$ and a 0 difference on the right block, after 2 rounds.

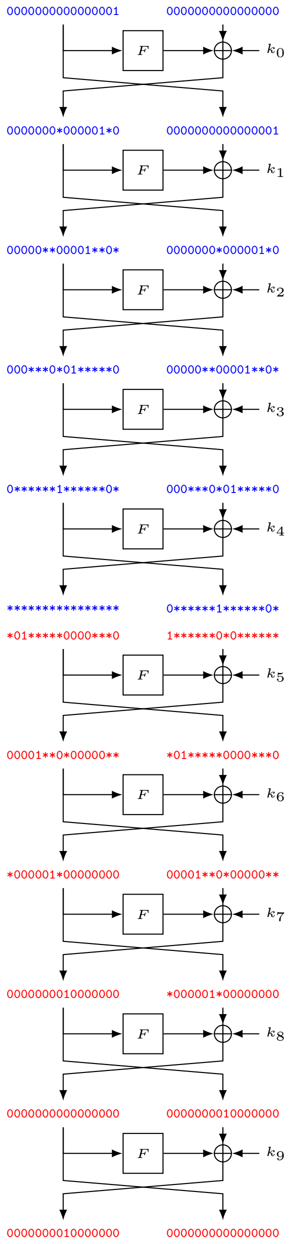


Fig. 11: A 10-round impossible differential for Simon32/64. Tracing truncated output differences in respectively forward and backward directions give a contradiction on the right half truncated mask after 5 rounds, where a 0 overlaps a 1.

Note, that the first observation above can be used in the beginning of the 6-round iterated differential described in Section 3 or the impossible differentials described in Section 4, as it yields an output difference of the form $(\alpha \parallel 0)$, which is exactly the starting point needed.

However, compared to the generic extension by two rounds on top, described in Section 3.4, this approach is impaired in two ways:

1. It has probability $p < 1$, and
2. For a fixed required difference α, the number of pairs obtainable is only 2^n, i.e. square root the number of pairs otherwise obtainable.