On the Effect of the Key Expansion Algorithm in Simon-like Ciphers

On the Effect of the Key-expansion Algorithm in Simon-like Ciphers

Jinyu Lu[1], Yunwen Liu[1⋆], Tomer Ashur[2], and Chao Li[1]

[1] College of Liberal Arts and Sciences, National University of Defense Technology, Hunan, Changsha, P. R. China
[2] imec-COSIC KU Leuven, Leuven, Belgium, and TU Eindhoven, The Netherlands
univerlyw@hotmail.com; tomer.ashur@esat.kuleuven.be

Abstract. In this work we investigate how the choice of the key-expansion algorithm and its interaction with the round function affect the resistance of Simon-like ciphers against rotational-XOR (RX) cryptanalysis. We observe that among the key-expansion algorithms we consider, Simon is most resistant, while Simeck is much less so. Implications on lightweight ciphers design are discussed and open questions are proposed.

Keywords: Rotational-XOR Cryptanalysis; Simeck; Simon; Key-expansion Algorithm

1 Introduction

Lightweight cryptography is a subfield of symmetric-key cryptography which presents a trade-off between suitable security and small implementations for resource-constrained devices. This approach mandates aggressive optimization of the components being used.

Key-expansion algorithms in particular seem to have been the target of such optimizations. Recent proposals often use a simple key-expansion algorithm, and sometimes even trivial. For examples of a simple key-expansion algorithm see Simon [4]; for algorithms not using a key-expansion algorithm at all see LED and Midori [3,10]. A middle ground between a "heavy" key-expansion algorithm and a simple one is to reuse the round function or some of its components for the key-expansion algorithm. For the former approach see Speck [4] and Simeck [36]; for the latter see [23,33].

The impact of the key schedule on cryptanalysis is important, yet in the context of lightweight block ciphers it remains to date understudied for the most part. Owing to the model of a Markov cipher due to Lai *et al.* [16], consecutive rounds of a cipher are assumed to be independent as long as the hypothesis of stochastic equivalence holds since the key will mask the relation between the output from the previous round and the input to the next one. Conversely, a study by Kranz *et al.* [15] showed the influence of a linear key-expansion

⋆ Corresponding author

algorithm on linear cryptanalysis in PRESENT. Along similar lines Abdelraheem *et al.* [1] showed how different choices of the key schedule result in different linear correlation distributions. It appears that there is no guarantee that for practical ciphers the stochastic equivalence hypothesis actually holds, especially when the key-expansion algorithm is simple or trivial. Information on how to design a good key-expansion algorithm and how it interacts with the round function remains scarce.

Lu *et al.* observed in [21] that the lightweight block cipher SIMON exhibits better resistance against RX-cryptanalysis than the lightweight block cipher SIMECK despite both belonging to the class of Simon-like ciphers. In this paper we set to understand the root cause for this gap. We observe that the difference between the two algorithms is twofold: (i) the rotation amounts used in the round function and (ii) the key-expansion algorithm; and set to isolate the determining factor.

Our starting point is the SMT model presented in [21] for finding RX-characteristics in Simon-like ciphers. This model was criticized by Sadeghi *et al.* who observed in [27] that the model will sometimes output incompatible RX-characteristics. We fix the model of Lu *et al.* by adding additional constraints ensuring the consistency of the RX-characteristic and apply the new model to a series of Simon-like ciphers with different parameters.

Our contribution:

- We correct the issues raised in [27] regarding the model devised in [21]. In that respect, we translate their MILP constraints into SAT/SMT and integrate them into our model;
- We evaluate the corrected model respective to SIMON32/64 and SIMECK32/64. We show that the RX-characteristic presented in [21] for SIMECK32/64 remains valid (as was also noted by [27]); For SIMON32/64 we find and validate longer RX-characteristics than those previously presented.
- We evaluate the corrected model respective to a sequence of the Simon-like ciphers and see how different design decisions reflect in the resistance of the resulting cipher against RX-cryptanalysis.

Organization. The paper is organized as follows: in Section 2 we recall the theory of RX-cryptanalysis, the structure of Simon-like ciphers, and the SMT model presented in [21]. In Section 3, we present the additional constraints required to ensure that the model is restricted to compatible RX-characteristics and evaluate this corrected model on SIMECK32 and SIMON32. Then, in Section 4, variants of Simon-like ciphers are presented and their resistance to RX-cryptanalysis is evaluated. We discuss possible interpretations of our results and directions for future research in Section 5 which concludes the paper.

2 Preliminaries

The table below presents the notation we use throughout the paper.

Notation	Description
$x = (x_{n-1}, \ldots, x_0)$	Binary vector of n bits; x_i is the bit in position i with x_0 the least significant one.
$\overline{x}$	Bitwise negation.
$x \odot y$	Bitwise AND between x and y.
$x \oplus y$	Bitwise XOR between x and y.
$x \| y$	Concatenation of x and y.
$x \vert y$	Bitwise OR between x and y.
$wt(x)$	Hamming weight of x.
$x \lll \gamma, S^{\gamma}(x)$	Circular left shift of x by γ bits.
$x \ggg \gamma, S^{-\gamma}(x)$	Circular right shift of x by γ bits.
$(I \oplus S^{\gamma})(x)$	$x \oplus S^{\gamma}(x)$.

2.1 Rotational-XOR Cryptanalysis

Rotational cryptanalysis is a related-key chosen-plaintext attack following the propagation of rotational pairs *i.e.*, pairs of the form $(x, x \lll \gamma)$. This attack is thwarted when a constant that is not rotation-invariant (*i.e.*, a constant c such that $c \neq c \lll \gamma$) is injected into the rotational pair; see *e.g.*, [5].

Rotational-XOR cryptanalysis is a generalized attack method taking such constants into account. Whereas the original technique was thwarted by the injection of round constants that are not rotational-invariant, RX-cryptanalysis overcomes this problem by integrating their effect into the analysis of the propagation probability. Rather than just considering a rotational pair as in the case of rotational cryptanalysis, RX-cryptanalysis considers an RX-pair of the form $(x, S^{\gamma}(x) \oplus \alpha)$ where α is called the *translation*. The technique was successfully applied to ARX-based primitives, including the block cipher Speck [18] and the PRF SipHash [35].

In [21] Lu *et al.* extended the applicability of RX-cryptanalysis also to AND-RX ciphers by showing that the RX-propagation probability through vectorial bitwise-AND is the same as the XOR-propagation probability through the same operation. This is captured by Theorem 1 which is reproduced from [21, Thm. 1].

Theorem 1. *Let $(x, (x \lll \gamma) \oplus \alpha)$ and $(y, (y \lll \gamma) \oplus \beta)$ be two RX-pairs where γ is the rotation offset and (α, β) the translations, respectively. Then, for an output translation Δ it holds that:*

$$\begin{aligned} \Pr[\,((x \odot y) \lll \gamma) \oplus \Delta = ((x \lll \gamma) \oplus \alpha) \odot ((y \lll \gamma) \oplus \beta)] \\ = \Pr[(x \odot y) \oplus \Delta = (x \oplus \alpha) \odot (y \oplus \beta)]\,, \end{aligned} \tag{1}$$

i.e., *the propagation probability of an RX-difference with translations (α, β) through $\odot$ is the same as that of a normal XOR-difference through the same operation when the translations are considered as input XOR-differences.*

Of particular interest in our paper is the RX-propagation probability through a Simon-like round which is given by Lemma 1 (reproduced from [21, Prop. 1])

Lemma 1. *For $S^a(x) \odot S^b(x)$ where $\gcd(n, a-b) = 1$, n is even, $a > b$ and $x = (x_{n-1}, \ldots, x_1, x_0) \in \mathbb{F}_2^n$, (α, β) as in Theorem 1, the difference propagation distribution table and RX propagation distribution are given by*

$$\Pr[\alpha \to \beta] = \begin{cases} 2^{-n+1} & \text{if } \alpha = \texttt{0xf}\cdots\texttt{f}, \\ & wt(\beta) \equiv 0 \bmod 2\,; \\ 2^{-A} & \text{if } \alpha \neq \texttt{0xf}\cdots\texttt{f}, \\ & \beta \odot \left(\overline{S^a(\alpha) \mid S^b(\alpha)}\right) = 0, \\ & \left(\beta \oplus S^{a-b}(\beta)\right) \odot \\ & \left(\overline{S^a(\alpha)} \odot S^{2a-b}(\alpha)\right. \\ & \left. \odot S^b(\alpha)\right) = 0; \\ 0 & \text{otherwise} \end{cases}$$

where $A = wt\Big(\big(S^a(\alpha) \mid S^b(\alpha)\big) \oplus \big(\overline{S^a(\alpha)} \odot S^{2a-b}(\alpha) \odot S^b(\alpha)\big)\Big)$.

Sadeghi *et al.* presented in [27] an MILP model for finding a right-pair for a given (RX-)characteristic. Surprisingly, they found that some RX-characteristics, although constructed using locally valid transitions, are incompatible *i.e.*, they do not allow for the propagation of any right pair due to global contradictions. This appears to be a general problem for automated search models which are set to track the propagation of differences without ensuring that the corresponding values can propagate simultaneously.

2.2 Simon-like Ciphers

Simon is a family of block ciphers following the AND-RX design paradigm, *i.e.*, members of the family can be described using only the bitwise operations AND ($\odot$), XOR ($\oplus$), and cyclic rotation by γ bits (S^γ). Simon-like ciphers generalize the structure of Simon's round function with parameters different from the original ones.

The round function

Simon is a family of lightweight block ciphers designed by the US NSA [4]. A member of the family is denoted by Simon$2n/mn$, to specify a block size of $2n$ for $n \in \{16, 24, 32, 48, 64\}$, and key size of mn for $m = \{2, 3, 4\}$. Since the key size can only be 64 when the block size $2n$ is equal to 32, we simply write it as Simon32 instead of Simon32/64 hereinafter. The round function of Simon is defined as

$$f(x) = \left(S^8(x) \odot S^1(x)\right) \oplus S^2(x)\,.$$

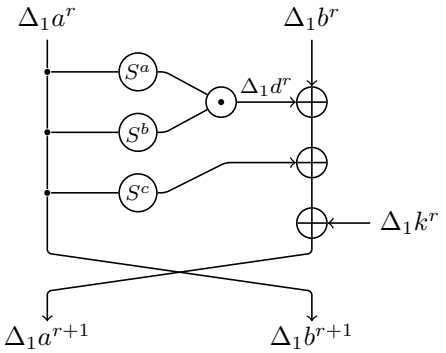


Fig. 1: The round function of a Simon-like cipher and notation for the propagation of RX-differences through it

Simon-like ciphers are ciphers that share the same round structure as SIMON, but generalize it to arbitrary rotation amounts (a, b, c) such that the round function becomes

$$f_{a,b,c}(x) = \left(S^a(x) \odot S^b(x)\right) \oplus S^c(x)\,.$$

This round function is depicted in Figure 1.

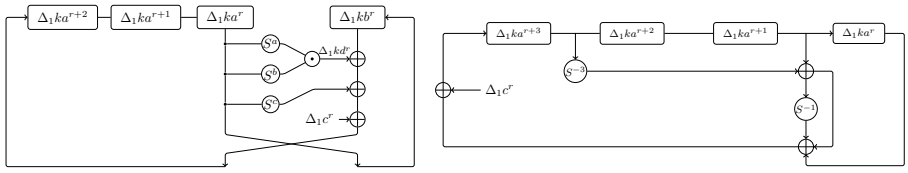
Of particular interest in this paper is the SIMECK family of lightweight block ciphers designed by Yang *et al.* [36], aiming at improving the hardware implementation cost of SIMON. SIMECK$2n/4n$ denotes an instance with a $2n$-bit block and a $4n$-bit key for $n \in \{16, 24, 32\}$. Since the key length of SIMECK is always $4n$ we use lazy writing in the sequel and simply write SIMECK$2n$ throughout the paper. The rotation amounts for all SIMECK versions are $(a, b, c) = (5, 0, 1)$.

Tracking the RX-propagation Lu *et al.* devised an SMT model for tracking the propagation of RX-differences in Simon-like ciphers. They have determined that the following set of constraints respective to the notation in Figure 1 is sufficient for finding a valid RX-characteristic:

$$\begin{aligned}
0 &= \Delta_1 d^r \odot (\overline{S^a(\Delta_1 a^r) \mid S^b(\Delta_1 a^r)})\,;\\
0 &= (\Delta_1 d^r \oplus S^{a-b}(\Delta_1 d^r)) \odot (\overline{S^a(\Delta_1 a^r)}\\
&\qquad \odot S^{2a-b}(\Delta_1 a^r) \odot S^b(\Delta_1 a^r))\,;\\
\Delta_1 b^{r+1} &= \Delta_1 a^r\,;\\
\Delta_1 a^{r+1} &= \Delta_1 d^r \oplus \Delta_1 b^r \oplus S^c(\Delta_1 a^r) \oplus \Delta_1 k^r\,.
\end{aligned}$$

If the propagation is valid, the probability in round r is given by $2^{-w_d^r}$, where

$$\begin{aligned}
w_d^r = wt((&S^a(\Delta_1 a^r) \mid S^b(\Delta_1 a^r)) \oplus (\overline{S^a(\Delta_1 a^r)}\\
&\odot S^{2a-b}(\Delta_1 a^r) \odot S^b(\Delta_1 a^r)))\,,
\end{aligned}$$



(a) Simeck32 (b) Simon32

Fig. 2: The key-expansion algorithms of Simeck32 and Simon32 and notation for their RX-difference propagation

is said to be the weight of the non-linear transition in round r.

The key-expansion algorithm The non-linear key-expansion algorithm of Simeck reuses the cipher's round function to generate the round keys. Let $K = (t_2, t_1, t_0, k_0)$ be the master key for Simeck$2n$, where $t_i, k_0 \in \mathbb{F}_2^n$. The registers of the key-expansion algorithm are loaded with

$$K = k_3||k_2||k_1||k_0$$

for K the master key, and the sequence of round keys $(k_0, \ldots, k_{T-1})$ is generated with

$$k_{i+1} = t_i$$

where

$$t_{i+3} = k_i \oplus f_{5,0,1}(t_i) \oplus c \oplus (z_j)_i,$$

and $c \oplus (z_j)_i \in \{\texttt{0xfffc}, \texttt{0xfffd}\}$ a round constant. A single round of Simeck is depicted in Figure 2a.

The SMT model given by Lu *et al.* for this key-expansion algorithm is analogous to that of the round function respective to the notation in Figure 2a:

$$\begin{aligned}
&0 = \Delta_1 kd^r \odot \overline{S^a(\Delta_1 ka^r) \mid S^b(\Delta_1 ka^r)};\\
&0 = (\Delta_1 kd^r \oplus S^{a-b}(\Delta_1 kd^r)) \odot (\overline{S^a(\Delta_1 ka^r)}\\
&\qquad \odot S^{2a-b}(\Delta_1 ka^r) \odot S^b(\Delta_1 ka^r))\,;\\
&\Delta_1 kb^{r+1} = \Delta_1 ka^r\,;\\
&\Delta_1 ka^{r+3} = \Delta_1 kd^r \oplus \Delta_1 kb^r \oplus S^c(\Delta_1 ka^r) \oplus \Delta_1 c^r\,;\\
&\Delta_1 k^r = \Delta_1 kb^r.
\end{aligned}$$

with weight w_k^r set as

$$\begin{aligned}
w_k^r = &wt((S^a(\Delta_1 ka^r) \mid S^b(\Delta_1 ka^r)) \oplus (\overline{S^a(\Delta_1 ka^r)}\\
&\odot S^{2a-b}(\Delta_1 ka^r) \odot S^b(\Delta_1 ka^r)))\,.
\end{aligned}$$

SIMON, conversely, uses a linear key-expansion algorithm to generate the round keys. Let $K = (k_{m-1}, \ldots, k_1, k_0)$ be a master key for SIMON$2n$, where $k_i \in \mathbb{F}_2^n$. The sequence of round keys k_i is generated by

$$k_{i+m} = \begin{cases} k_i \oplus & (I \oplus S^{-1})S^{-3}k_{i+1} \oplus c \oplus (z_j)_i, & \text{if } m = 2 \\ k_i \oplus & (I \oplus S^{-1})S^{-3}k_{i+2} \oplus c \oplus (z_j)_i, & \text{if } m = 3 \\ k_i \oplus & (I \oplus S^{-1})(S^{-3}k_{i+3} \oplus k_{i+1}) \oplus c \oplus (z_j)_i, & \text{if } m = 4 \end{cases}$$

for $0 \leq i \leq (T - m)$. The key-expansion algorithm of SIMON with $m = 4$ is depicted in Figure 2b.

Lu *et al.* modeled this key-expansion algorithm as:

$$\begin{aligned} \Delta_1 ka^{r+4} = & S^{-3}(\Delta_1 ka^{r+3}) \oplus \Delta_1 ka^{r+1} \\ & \oplus S^{-1}(S^{-3}(\Delta_1 ka^{r+3}) \oplus \Delta_1 ka^{r+1}) \\ & \oplus \Delta_1 ka^r \oplus \Delta_1 c^r \\ \Delta_1 k^r = & \Delta_1 ka^r \,. \end{aligned}$$

Since it is linear, there is no need to track the weight of the RX-propagation in this key-expansion algorithm.

2.3 SAT/SMT Automated Search Method

Since Matsui first proposed a program for automatically searching linear characteristics for DES block cipher at EUROCRYPT 1993 [22], automatic tools for cryptanalysis play an important role in the design and cryptanalysis of symmetric ciphers. The main automatic analysis methods are based on Boolean Satisfiability Problem (SAT)/Satisfiability Modulo Theories (SMT) search method [12, 18, 24, 28], Mixed Integer Linear Programming (MILP) search method [8, 25, 29, 31, 32, 34], and Constraint Programming (CP) search method [9, 30]. The idea behind these search methods is to model the search problem as a set of constraints and solve it using one of the available constraint solvers.

The SAT problem is the problem of determining if there exists an instantiation that satisfies a given Boolean formula. The SMT problem is to determine the satisfiability of the first-order logic formula under a specific theory. Compared with the SAT problem, the SMT problem can be expressed with richer languages (theories) than boolean formulas. In particular, a formula in the bit-vector theory can contain bit-vectors (a vector of boolean variables) and the usual operations of bit-vectors such as bitwise operations (AND, XOR, OR, etc.) arithmetic operations (addition, multiplication, etc.), cyclic operations and so on. For SMT problems, the main solvers are Boolector [7], Z3 [26], STP[3], etc.

Since an ARX/AND-RX cipher only contains basic Boolean operations: modular addition, AND, cyclic shift and XOR, it is natural to describe in SMT-LIB. Thus, we use the automatic searching tool relying on SAT/SMT instead of MILP. Nevertheless, it is also applicable to adopt an MILP solver, see for example [11, 27].

[3] http://stp.github.io/

3 Compatibility

Sadeghi *et al.* presented in [27] an MILP model outputing a solution respective to a given RX-characteristic *i.e.*, a pair of related keys and a right pair satisfying said characteristic. They observed that some of the RX-characteristics in [21] cannot produce right pairs respective to any key pair due to global contradictions; such RX-characteristics are said to be incompatible.

Following their work, we adapted the work of Lu *et al.* with some additional constraints ensuring the compatibility of the output RX-characteristic. Let $k^{r+1} = f_{ks}(t_2^r, t_1^r, t_0^r, k^r, c^r)$, where $f_{ks}(t_2^r, t_1^r, t_0^r, k^r, c^r)$ denotes the function deriving the subkey k^{r+1} from the state of the key-expansion algorithm in round r and the round constant c^r. Further let k^r and $(k^r)'$ denote the n-bits subkeys to round r, respective to the master keys K and K'.

Then, the following constraints should be satisfied for the RX-characteristic to be compatible:

$$k^{r+1} = f_{ks}(t_2^r, t_1^r, t_0^r, k^r, c^r); \tag{2}$$

$$(k^{r+1})' = f_{ks}((t_2^r)', (t_1^r)', (t_0^r)', (k^r)', c^r); \tag{3}$$

$$\Delta_1 k^r = (k^r \lll 1) \oplus (k^r)'; \tag{4}$$

$$\Delta_1 k^{r+1} = (k^{r+1} \lll 1) \oplus (k^{r+1})'. \tag{5}$$

In simple words, Constraints (2)–(3) ensure that the subkey can be derived from the master key (consistency) while Constraints (4)–(5) ensure that they have the appropriate RX-difference.

Once the RX-characteristic for the key-expansion algorithm is determined to be compatible, let $(x^{r+1}, y^{r+1}) \hat{=} R(x^r, y^r, k^r) = (f_{a,b,c}(x^r) \oplus y^r \oplus k^r, x^r)$ denote the encryption function for round r taking the pair (x^r, y^r) as left and right inputs, respectively, k^r the subkey; and returning (x^{r+1}, y^{r+1}) as the left and right outputs, respectively.

Then, the following constraints should be satisfied for the RX-characteristic to be compatible respective to the subkeys found in (2)–(5):

$$(x^{r+1}, y^{r+1}) = R(x^r, y^r, k^r); \tag{6}$$

$$((x^{r+1})', (y^{r+1})') = R((x^r)', (y^r)', (k^r)'); \tag{7}$$

$$\Delta_1 x^r = (x^r \lll 1) \oplus (x^r)'; \tag{8}$$

$$\Delta_1 y^r = (y^r \lll 1) \oplus (y^r)'; \tag{9}$$

$$\Delta_1 x^{r+1} = (x^{r+1} \lll 1) \oplus (x^{r+1})'; \tag{10}$$

$$\Delta_1 y^{r+1} = (y^{r+1} \lll 1) \oplus (y^{r+1})'. \tag{11}$$

3.1 Running the Model

With the above model in mind, we begin by searching RX-characteristics for Simon32 and Simeck32. We describe the model using the SMT-LIB language and apply the Boolector solver with several parameter sets. Our experiments

were carried out on a laptop with Intel Core i5-7300U CPU running at 2.60GHz and a server with Intel Xeon(R) Core E5-2609 v2 CPU running at 2.50GHz. The source code for this and subsequent searches can be found in [20]. The results are presented in Table 1.

Table 1: The probability of compatible RX-characteristics found for SIMON32 and SIMECK32 with $\gamma = 1$. The probabilities are given in two columns for SIMECK32 distinguishers, where the "Data Prob." is the probability of the round function part and the "Key Prob." is that of the key schedule part. All probabilities p are given as $-\log_2 p$. The column "time" provides the time needed to find a RX-characteristic in seconds or hours ("s" and "h" for short). For instance, the found RX-characteristic covering 20-round SIMECK32 has a data probability of 2^{-26} and a weak key size $2^{64-34} = 2^{30}$ with 255.40 seconds. We see that RX-characteristics for SIMON32 cover significantly fewer rounds. Entries marked with an asterisk are not necessarily optimal *i.e.*, an RX-characteristics covering the same number of rounds with better probabilities may still exist.

	SIMON32		**SIMECK32**		
Rounds	**Prob.**	**Time**	**Data Prob.**	**Key Prob.**	**Time**
6	0	0.10s	0	0	0.06s
7	4	1.82s	2	4	0.47s
8	6	2.94s	4	4	0.67s
9	10	106.01s	4	6	0.78s
10	14	2.90h	6	8	1.52s
11	22*	55.56h	10	12	7.00s
12	26*	78.55h	12	12	8.34s
13	30*	13.98h	12	18	39.03s
14	32*	6.66h	16	18	24.14s
15			18	20	26.31s
16			18	28	420.45s
17			18	32	180.67s
18			22	30	305.07s
19			24	34	900.13s
20			26	34	255.40s

The longest RX-characteristic we found for SIMON32 covers 14 rounds and its probability is 2^{-32} for the entire key space. This RX-characteristic is not necessarily optimal, yet for 15 rounds we were only able to find an RX-characteristic with probability 2^{-36} which would require more data than what is allowed by the block size. The full description of this RX-characteristic is presented in A, Table 10.

For SIMECK32 the longest RX-characteristic we found covers 20 rounds. The probability of this RX-characteristic is 2^{-26} and it applies to a weak-key class of size 2^{30}. We further found that there exists no RX-characteristic with $w_d + w_k \leq 64$ for more than 20 rounds of SIMECK32; therefore, our 20-round RX-characteristic is tight with respect to the number of rounds and optimal respective to the objective function. The full description of this RX-characteristic is presented in A, Table 11.

4 The Effect of the Key-expansion Algorithm on the Resistance Against RX-cryptanalysis of Simon-like Constructions

Lu *et al.* observed that SIMECK appears to be more vulnerable to RX-cryptanalysis than its counterpart SIMON. This, albeit to a lesser degree, is also the conclusion from Table 1. There are two main differences between SIMON and SIMECK: (i) the key-expansion function, and (ii) the rotation amounts in the round function. In this section we set to investigate how each of these differences affects the overall resistance against RX-cryptanalysis. This is done by introducing new variants which isolate the property we are interested in. To compare the design components in Simon-like ciphers, here we concentrate on discussing four main features:

- (A) The type of key-expansion algorithm. We consider the SIMECK and SIMON key-expansion algorithms with $m = 4$, as well as two additional key-expansion algorithms derived from each (see Figures 3–4).
- (B) The round function's rotation amounts (a, b, c). Three rotation amounts are considered: $(8, 1, 2)$ from SIMON, $(5, 0, 1)$ from SIMECK, and $(12, 5, 3)$ suggested in [13, 14] and deemed optimal against certain attacks;
- (C) The key-expansion function's rotation amounts (a, b, c) when it reuses the round function following the design philosophy of SIMECK;
- (D) The round constants used in the key-expansion algorithm, where the Sparkle-like round constants are provided in B.

A variant is denoted by SIM-(A,B,C,D) where SIM means that the cipher is Simon-like and the tuple (A,B,C,D) defines the controlled variables such that (A) defines the type of the key-expansion algorithm; (B) the rotation amounts for the round; (C) the rotation amounts for the key-expansion algorithm when reusing the round function, and (D) the round constants. The legend for interpreting the tuple (A,B,C,D) is given in Table 2. For example, SIMECK32 is SIM-(1,1,1,1). When a certain parameter is not relevant it is denoted by a dash *e.g.*, SIMON32 is SIM-(2,2,-,2).

4.1 Non-Linear Key-expansion Algorithms (Parameter (A))

We begin by investigating the effect of different parameters on the resistance against RX-cryptanalysis in non-linear key-expansion algorithms. First, in Section 4.1 we investigate how the resistance of SIMECK against RX-cryptanalysis

Table 2: The full parameter set for SIM-(A,B,C,D), where z_0–z_4 is the constant sequence of SIMON. For example, SIMON32 is SIM-(2,2,-,2) since it uses the SIMON32 key-expansion algorithm (Parameter (A)), the round function's rotation amounts are $(8, 1, 2)$ (Parameter (B)), the key-expansion function does not reuse the round function hence it has no Parameter (C), and $\texttt{0xfffc} \oplus z_0$ for the round constants (Parameter (D)).

Parameter	(A) KS-type	(B) Rotation amounts (round)	(C) Rotation amounts (KS)	(D) Round constants
1	SIMECK	(5,0,1)	(5,0,1)	$\texttt{0xfffc} \oplus z_{\text{SIMECK32}}$
2	SIMON32	(8,1,2)	(8,1,2)	$\texttt{0xfffc} \oplus z_0$
3	Fig 3a	(12,5,3)	(12,5,3)	$\texttt{0xfffc} \oplus z_1$
4	Fig 3b			$\texttt{0xfffc} \oplus z_2$
5	Fig 4a			$\texttt{0xfffc} \oplus z_3$
6	Fig 4b			$\texttt{0xfffc} \oplus z_4$
7				round counter
8				Sparkle-like

is affected when different rotation amounts are used. Note that here, similar to the original design, the rotation amounts in the round function and the key-expansion algorithm are the same, *i.e.*, Parameter (B) and Parameter (C) are equal. Then, in Section 4.1 we analyze two novel non-linear key-expansion algorithms. In Section 4.1 we break the symmetry between the round function and the key-expansion algorithm and investigate how different combinations for Parameters (B)–(C) affect the overall resistance of the cipher against RX-cryptanalysis. Finally, in Section 4.1 we isolate the round constants and investigate their effect.

Controlling for the rotation amounts (Parameters (B)–(C)) Our starting point is SIMECK32 *i.e.*, SIM-(1,1,1,1). The first two variants we investigate, SIM-(1,2,2,1) and SIM-(1,3,3,1), follow the same design philosophy and reuse the round function in the key-expansion algorithm. The three variants differ only in the rotation amounts, which are $(5, 0, 1)$, $(8, 1, 2)$ and $(12, 5, 3)$, respectively; and are kept the same for the round function and the key-expansion algorithms in each respective cipher. The round constants are always the same as those of SIMECK.

The results are presented in Table 3. The difference appears to be meaningful and we conclude that Simon-like ciphers reusing the round function in the key-expansion algorithm are highly sensitive to Parameters (B) and (C) (*i.e.*, the rotation amounts in the round function and the key-expansion algorithm, respectively). We further see that SIMECK is the most vulnerable of the three variants.

Controlling for non-linearity (Parameter A) Recalling that the longest RX-characteristic we could find for SIMON32 covered only 14 rounds, we now want to determine if the relatively weak resistance of the SIMECK design philosophy as reflected in Table 3 is specific to reusing the Simon-like round in

Table 3: The effect of different rotation amounts on Simon-like ciphers reusing the round function in the key-expansion algorithm. We fix the SIM parameters such that $(A) = (D) = 1$, and evaluate for different $(B) = (C) \in \{1, 2, 3\}$. We see that the resistance of the resulting ciphers against RX-cryptanalysis is highly sensitive to this decision.

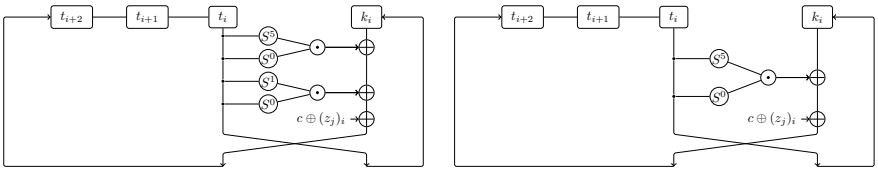
Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
SIM-(1,1,1,1)											
Data	6	10	12	12	16	18	18	18	22	24	26
Key	8	12	12	18	18	20	28	32	30	34	34
SIM-(1,2,2,1)											
Data	12	16	18	20	22	24	28	28			
Key	10	12	18	18	20	22	26	32			
SIM-(1,3,3,1)											
Data	12	16	14	24	28	30					
Key	12	14	24	22	26	32					

the key-expansion algorithm or more generally to non-linear key-expansion algorithms.

We define two more variants with novel non-linear key-expansion algorithms inspired by Simon-like rounds. SIM-(3,1,-,1) and SIM(4,1,-,1) use the SIMECK rotation amounts and round constants, with the key-expansion algorithms depicted in Figures 3a–3b, respectively.

Note that the point here is not to analyze the security of these two key-expansion algorithms but to ascertain if the key schedule of SIMECK is an outlier to the resistance offered by an arbitrary non-linear one. Here we consider two key-expansion algorithms that reuse (part of) the round function as variants of SIMECK, one with two non-linear branches (Figure 3a) and the other with only one non-linear branch (Figure 3b).

The results are presented in Table 4. We conclude that reusing the round function, by itself, is not a bad design approach for lightweight ciphers.



(a) SIM-(3,1,-,1) (b) SIM-(4,1,-,1)

Fig. 3: Novel non-linear key-expansion algorithms for SIM-(3,1,-,1) and SIM-(4,1,-,1)

Table 4: The effect on the resistance of Simon-like ciphers when different approaches to designing a non-linear key-expansion function are taken. We fix the SIM parameters such that $(B) = (D) = 1$ and evaluate for $(A) \in \{3, 4\}$. These are compared to SIM-(1,1,1,1) and SIM-(1,3,3,1) which are the worst and the best rotation amounts from Table 3. The novel key-expansion algorithms do not appear to be particularly good or bad compared to reusing the round function.

Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
SIM-(1,1,1,1)														
Data	6	10	12	12	16	18	18	18	22	24	26			
Key	8	12	12	18	18	20	28	32	30	34	34			
SIM-(1,3,3,1)														
Data	12	16	14	24	28	30								
Key	12	14	24	22	26	32								
SIM-(3,1,-,1)														
Data	4	10	10	12	14	16	16	20	16	18				
Key	12	12	18	24	24	30	34	34	42	46				
SIM-(4,1,-,1)														
Data	6	8	8	10	12	14	14	18	18	20	22	24	28	26
Key	8	12	14	16	20	20	24	24	28	28	30	32	32	38

Controlling for the self-similarity between Parameter (B) and Parameter (C) We now break the link between Parameter (B) and Parameter (C) by allowing $(B) \neq (C)$. The six variants we investigate, SIM-(1,1,2,1), SIM-(1,1,3,1), SIM-(1,2,1,1), SIM-(1,2,3,1), SIM-(1,3,1,1), and SIM-(1,3,2,1), all reuse the round function and the round constants of SIMECK and differ in their combination of rotation amounts for the round function and the key-expansion algorithm.

The results are presented in Table 5. Most notably we observe that when the $(12, 5, 3)$-parameter is used in the round function (*i.e.*, variants with (B)=3), the resistance against RX-cryptanalysis is relatively stable regardless of rotation amounts in the key-expansion algorithm.

It is interesting to see that SIM-(1,1,2,1) and SIM-(1,2,1,1) which are mirrored versions of each another offer entirely different resistance against RX-cryptanalysis. Note that the results we report here are optimal, *i.e.*, for a fixed number of rounds there are no better RX-characteristics than those in Table 5.

The self-similarity between the round function and the key-expansion algorithm does not seem to have a meaningful effect. SIM-(1,1,1,1) (*i.e.*, SIMECK32) remains the most vulnerable parameter choice, while the other self-similar variants, SIM-(1,2,2,1) and SIM-(1,3,3,1), fare relatively well compared to SIM-(1,1,1,1), as well as some of their non-self-similar counterparts.

We conclude that not only the ciphers' overall resistance against RX-cryptanalysis is sensitive to the choice of Parameters (B) and (C) but the round function and the key-expansion algorithm appear to be sensitive separately to these decisions.

Controlling for the round constants (Parameter (D)) To ensure the effect of the round constants on the key-expansion algorithm of SIMECK32, we fix the parameters such that (A) = 1, (B) = (C) $\in \{1, 2, 3\}$ and evaluate for (D)

Table 5: A comparison of Simon-like variants with different parameter choices for Parameters (B) and (C). We see that the cipher's resistance against RX-cryptanalysis is sensitive to this choice and that self-similar key-expansion algorithms do not pose a particular risk.

Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Sim-(1,1,1,1)											
Data	6	10	12	12	16	18	18	18	22	24	26
Key	8	12	12	18	18	20	28	32	30	34	34
Sim-(1,1,2,1)											
Data	6	10	14	16	20	20	26	28	32		
Key	10	14	14	18	20	24	24	28	32		
Sim-(1,1,3,1)											
Data	10	14	14	22	24	28					
Key	8	12	18	22	28	34					
Sim-(1,2,1,1)											
Data	8	12	12	16	24	18	20	24	24	26	
Key	10	12	18	22	16	30	32	32	36	38	
Sim-(1,2,2,1)											
Data	12	16	18	20	22	24	28	28			
Key	10	12	18	18	20	22	26	32			
Sim-(1,2,3,1)											
Data	12	14	16	22	24	26					
Key	10	16	24	24	30	34					
Sim-(1,3,1,1)											
Data	12	16	16	20	22	26	30				
Key	10	10	18	20	24	26	28				
Sim-(1,3,2,1)											
Data	14	16	18	22	24	28	30				
Key	10	14	20	20	28	26	32				
Sim-(1,3,3,1)											
Data	12	16	14	24	28	30					
Key	12	14	24	22	26	32					

$\in \{1, 2, 7, 8\}$, where the parameter round constants (D) = 1 and (D) = 2 are only different in the order of `0xfffc` and `0xfffd`, (D) = 7 is the round constants on the key-expansion algorithm of SPECK and (D) = 8 is the round constants in the SPARKLE permutation.

The results are presented in Table 6. When (D) takes a value in $\{1, 2, 7\}$, there is a minor difference in the round constants. However, there is a significant difference for (D) = 8, this means that the ability of the ciphers to resist RX-cryptanalysis can be affected by the different choices of the round constants.

Since the propagation of an RX-difference through the round constant c of the key schedule in round r is modeled by XOR $\Delta_1 c^r$, where $\Delta_1 c^r = c \oplus (c \lll \gamma)$. It can be pointed out that when the $wt(\Delta_1 c^r)$ is heavier, zeros and ones in $\Delta_1 c^r$ are distributed more evenly, and the RX-differences $\Delta_1 c^i$ and $\Delta_1 c^j$ are independent of each other for $i \neq j, i, j \in \{0, 1, \cdots, r\}$, then the resistance against RX-cryptanalysis is stronger.

4.2 Linear Key-expansion Algorithms (Parameter (A))

In the previous subsection we saw that non-linear key-expansion algorithms are highly sensitive to all manners of design choices, with SIMECK32 (*i.e.*, SIM-(1,1,1,1)) being in particular vulnerable to RX-cryptanalysis among the variants we considered.

The previous subsection can be used to explain the gap between SIMECK and SIMON observed by Lu *et al.* To determine whether this explanation accounts for the entire gap or just for a part of it we now focus on the key-expansion algorithm of SIMON and other linear key-expansion algorithms.

We begin in Section 4.2 by comparing different rotation amounts respectively to the key-expansion algorithm of SIMON32. Then, in Section 4.2 we consider two novel linear key-expansion algorithms. Finally, in Section 4.2 we compare the effect of different round constants.

Controlling for the rotation amounts in the round function (Parameter (B)) We compare different rotation amounts for the round function, using the key-expansion algorithm of SIMON32. The diffusion of different rotation amounts was previously investigated in *e.g.*, [13, 14] and "good" rotation amounts were determined respectively to certain attacks. However, whereas the key-expansion algorithm was abstracted in all previous work analyzing the effect of the rotation amounts, it is the main focus of this paper. That is to say that we are interested in rotation amounts that produce fast diffusion only insofar they do so respective to the selected key-expansion algorithm. This approach appears intuitive in our case since RX-cryptanalysis is a related-key attack. However, following the work of Sadeghi *et al.* in [27] and the additional constraints we add in Section 3 it is important to reconsider if conclusions of previous work were not based on incompatible or phantom (see [17, 19]) characteristics.

The first two variants we consider, SIM-(2,1,-,2) and SIM-(2,3,-,2) use the key-expansion algorithm of SIMON32 with the original round constants, but differ in the rotation amounts for the round function.

Table 6: A comparison of Simon-like ciphers with different round constants. We see there is a significant difference when the parameters (D) $\in \{1, 2, 7\}$ change to the (D) $= 8$ round constants.

Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
SIM-(1,1,1,1)											
Data	6	10	12	12	16	18	18	18	22	24	26
Key	8	12	12	18	18	20	28	32	30	34	34
SIM-(1,1,1,2)											
Data	6	10	12	14	16	18	20	18	22	24	26
Key	8	12	12	18	18	22	22	30	30	34	36
SIM-(1,1,1,7)											
Data	6	10	12	12	16	18	18	22	22	26	
Key	8	12	12	18	18	22	28	30	36	34	
SIM-(1,1,1,8)											
Data	18	24	26*								
Key	20	24	30*								
SIM-(1,2,2,1)											
Data	12	16	18	20	22	24	28	28			
Key	10	12	18	18	20	22	26	32			
SIM-(1,2,2,2)											
Data	12	16	18	20	22	26	28	26	28		
Key	10	12	18	20	20	22	24	32	36		
SIM-(1,2,2,7)											
Data	8	12	14	16	20	26	26				
Key	8	14	16	20	24	24	30				
SIM-(1,2,2,8)											
Data	18	22	24*								
Key	20	24	30*								
SIM-(1,3,3,1)											
Data	12	16	14	24	28	30					
Key	12	14	24	22	26	32					
SIM-(1,3,3,2)											
Data	12	16	20	24	26	30					
Key	12	14	18	22	28	32					
SIM-(1,3,3,7)											
Data	12	16	18	24	28	30					
Key	14	18	22	24	28	32					
SIM-(1,3,3,8)											
Data	20	22	26*								
Key	20	26	30*								

The results are compared respectively to SIMON32 in Table 7. We see that the rotation amounts in the round function do not impose a significant difference on the resistance to RX-cryptanalysis. However, it is important to note that these rotation amounts are not arbitrary and are the result of previous work suggesting they would be good ones.

Table 7: The effect of different rotation amounts (Parameter (B)) on the resistance of the cipher against RX-cryptanalysis when parameters (A), (C)–(D) are the same as in SIMON32. No significant difference is observed. Entries marked with an asterisk are not necessarily optimal *i.e.*, an RX-characteristics covering the same number of rounds with better probabilities may still exist.

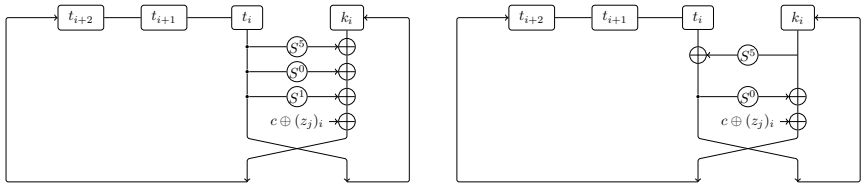
Rounds	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
SIM-(2,2,-,2)	0	4	6	10	14	22*	26*	30*	32*
SIM-(2,1,-,2)	0	4	6	8	12	16	24*	28*	32*
SIM-(2,3,-,2)	0	4	6	8	15	22*	26*	31*	

Controlling for linearity and the round constants (Parameters (A) and (D)) We have seen that the key-expansion algorithm of SIMON offers good resistance against RX-cryptanalysis independently of the specific choice [4] of the rotation amounts in the round function. To determine if this property is particular to this key-expansion algorithm or general in linear key-expansion algorithms, we define two novel linear key-expansion algorithms. Meanwhile, to compare the linear and non-linear key-expansion algorithms in a reasonable sense, the algorithms proposed are two constructions that can be regarded as linear variants of the key-expansion algorithms of SIMECK. For instance, the one in Figure 4a replaces the AND operation by an XOR, and Figure 4b is a simplified construction from Figure 4a.

SIM-(5,1,-,2) and SIM-(6,1,-,2) differ from SIM-(2,1,-,2) analyzed in the previous subsection only in their key-expansion algorithm. SIM-(5,1,-,2) uses the key-expansion algorithm in Figure 4a and SIM-(6,1,-,2) uses the key-expansion algorithm in Figure 4b. Meanwhile, to determine the effect of the round constants $(D) \in \{1, 2\}$ on the resistance to RX-cryptanalysis we define SIM-(5,1,-,1) and SIM-(6,1,-,1). These are respectively analog to SIM-(5,1,-,2), SIM-(6,1,-,2), but they use the round constants of SIMECK.

The results can be found in Table 8 and we can clearly see that the new variants are more vulnerable to RX-cryptanalysis than SIM-(2,1,-,2) which is in itself comparable to SIMON32. We conclude that the key-expansion algorithm of SIMON offers particular resistance against RX-cryptanalysis.

[4] Based on the offsets that are strong in differential/linear cryptanalysis.



(a) SIM-(5,1,-,2) (b) SIM-(6,1,-,2)

Fig. 4: Novel linear key-expansion algorithms for SIM-(5,1,-,2) and SIM-(6,1,-,2)

Table 8: The effect on the resistance of Simon-like ciphers when different approaches to designing the linear key-expansion algorithm are taken. We fix the SIM parameters such that $(B) = 1$ and evaluate for $(A) \in \{5, 6\}$, $(D) \in \{1, 2\}$. These are compared to SIM-(2,2,-,2) (*i.e.*, SIMON32). The difference between the algorithms appears to be meaningful and the novel key-expansion algorithms are categorically worse than SIMON. Entries marked with an asterisk are not necessarily optimal *i.e.*, an RX-characteristics covering the same number of rounds with better probabilities may still exist.

Rounds	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
SIM-(2,1,-,1)	0	4	6	8	12	18*	26*	28*	32*					
SIM-(2,1,-,2)	0	4	6	8	12	16	24*	28*	32*					
SIM-(5,1,-,1)	0	2	4	6	8	11	16	18	23*	27*	30*			
SIM-(5,1,-,2)	0	2	4	6	8	11	15	17	22	25	31*			
SIM-(6,1,-,1)	0	2	2	4	6	8	11	14	17	18	21	25	27	32*
SIM-(6,1,-,2)	0	2	2	4	6	8	11	14	17	18	21	25	27	32*

Controlling for the round constants in the key-expansion algorithm of Simon32 (Parameter (D)) Finally, to determine the effect of the round constants in the key-expansion algorithm of SIMON32 we consider a sequence of variants SIM-(2,2,-,3), SIM-(2,2,-,4), SIM-(2,2,-,5) and SIM-(2,2,-,6). These variants are the same as SIMON32 in all but the round constants which are taken from SIMON48/96, SIMON64/96, SIMON64/128, SIMON128/256, respectively.

In addition, we consider the variant SIM-(2,2,-,7), which is the same as the previous ones, but uses the round number as the round constant. This strategy was used in the key-expansion algorithm of SPECK, SIMON's counterpart. At the same time, we consider the variant SIM-(2,2,-,8), which is the same as SIMON32, but uses the Sparkle-like round constants.

The results are presented in Table 9. In C, Table 12 we present results for the same experiment but with rotation amounts $(5, 0, 1)$; the trend appears to be the same. We are unable to conclude why different versions of SIMON use different round constants, nor why the round number is not used for this as in SPECK.

However, there is an obvious difference when the round constants (D) $\in \{1-7\}$ change to the Sparkle-like round constants, which is similar to Section 4.1. This also verifies the conclusion of Section 4.1 that "When the $wt(\Delta_1 c^r)$ is heavier, zeros and ones in $\Delta_1 c^r$ are distributed more evenly, and the RX-differences $\Delta_1 c^i$ and $\Delta_1 c^j$ are independent of each other for $i \neq j$, $i, j \in \{0, 1, \cdots, r\}$, then the resistance against RX-cryptanalysis is stronger".

Table 9: A comparison of Simon-like ciphers with (B) = 2 and varying over Parameter (D). Entries marked with an asterisk are not necessarily optimal *i.e.*, an RX-characteristics covering the same number of rounds with better probabilities may still exist.

Rounds	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
SIM-(2,2,-,1)	0	4	6	10	14	22*	26*	28*	
SIM-(2,2,-,2)	0	4	6	10	14	22*	26*	30*	32*
SIM-(2,2,-,3)	0	3	6	10	13	22*	26*	30*	
SIM-(2,2,-,4)	0	3	6	9	13	22*	26*	32*	
SIM-(2,2,-,5)	0	4	6	10	12	22*	26*	30*	
SIM-(2,2,-,6)	0	4	6	10	13	22*	26*	32*	
SIM-(2,2,-,7)	0	4	6	10	15	24*	26*	30*	
SIM-(2,2,-,8)	0	5	9	13	20*	28*	32*		

5 Discussion and Conclusion

The design of key-expansion algorithms has not received much attention in recent years. In fact, the opposite seems to be the case and recent lightweight proposals often give up on designing the key-expansion algorithm as a component of importance and opt for trivial key-expansion or a minimal one.

In this paper we investigated how the choice of the key-expansion algorithm and the way it interacts with the round function, affect the cipher's resistance against RX-cryptanalysis. This was done by suggesting a sequence of key-expansion algorithms to be used in multiple variants of Simon-like ciphers and analyzing the resulting resistance.

The so-called design rationale of SIMON [5, P. 5, footnote *] recites Zhang *et al.* [37] in saying "*we conclude that it is not advisable for Simon-like ciphers to re-use the round function in the key schedule*". Our experiments do not support this assertion. First, Table 3 reveals that certain parameter choices can offer resistance levels coming close to those of SIMON. Then, from Table 4 it appears that there is nothing inherently weak in reusing the round function for the key-expansion algorithm, and that other non-linear key-expansion algorithms do not offer categorically better resistance against RX-cryptanalysis.[5] Table 5 drives the point home by observing that the resistance is sensitive to seemingly trivial choices. Using the rotation amounts (5,0,1) and (8,1,2) for the round function and key-expansion algorithm, respectively, offers better resistance than using them counter-respectively. From Table 6 we conclude that the choice of round constants has a significant impact on the resistance against RX-cryptanalysis. Finally and surprisingly, it appears that compared to all parameter sets we investigated in this paper, the one from [36] (*i.e.*, SIMECK) appears to be especially vulnerable against RX-cryptanalysis.

Despite being an object of research for more than seven years, remarkably little is known about the key-expansion algorithm of SIMON. In [5], Beaulieu *et al.* do not provide any reproducible claims about how the key-expansion algorithms were selected. Our experiments reveal that the resistance against RX-cryptanalysis when using the key-expansion algorithm of SIMON is for the most part independent of any parameter we could control, yet it offers better resistance compared to all other linear and non-linear key-expansion algorithms we considered. We were not able to determine why different versions of SIMON use entirely different schemes to generate the round constants. Using the round number as the round constants, as the designers chose to do for SPECK, could have further improved efficiency for the cipher and it would be interesting to understand why the designers did not opt for this approach. Overall, SIMON's key-expansion algorithm offers the best resistance among those surveyed here and it is evident that an undisclosed design security criterion was used to select it.

Whereas claims have previously been made that [5], pertaining to detail the design rationale of SIMON, does not offer any new insight and merely repeats

[5] And the same holds when taking Table 8 into account.

published work (see *e.g.*, [2]), this paper is the first to provide an affirmative example for this and show that the effect can be significant. Although a superficial conclusion from our results would be that SIMON is robust, Ashur's point was that the undisclosed nature of the security criteria leading to these decisions, and the manipulation by Beaulieu *et al.* in the language of the so-called design rationale, are by themselves reasons for concern.

Caveats and future work. Ours is an exploratory research covering a small part of the research domain. We have only considered Simon-like ciphers, only against one type of attack, and mostly in the relation of SIMON-like or SIMECK-like key-expansion algorithms. However, we do believe that this exploration is of value.

First of all, although RX-cryptanalysis is often presented as a generalization of rotational cryptanalysis, it can also be viewed as a type of related-key differential cryptanalysis where the differences are respective to different bit positions. As such, we believe that our results are indicative of related-key differential attacks, suggesting that further investigation would be appropriate.

Secondly, that the key-expansion algorithm itself, as well as the way it interacts with the round function, affect the security of the resulting cipher is evident already from our limited exploration on Simon-like structures and there is no reason to believe that this would not generalize. It also appears that reusing the round function for the key-expansion algorithm is not inherently bad, and that the effect of round constants does appear to be significantly understood as a single factor in an overall design. Designers of lightweight ciphers should be aware of these points. It would be interesting to extend our research to other common key-expansion algorithms and key schedules for lightweight ciphers.

Conclusion. In this paper we corrected the SMT model devised by Lu *et al.* according to the critique raised by Sadeghi *et al.* We found again the 20-round RX-characteristic for SIMECK32 found by Lu *et al.* and a 14-round RX-characteristics for SIMON32, extending theirs by 3 rounds. We then compared how different key-expansion algorithms affect said resistance in Simon-like ciphers and found that SIMECK is especially vulnerable among those we tested while SIMON is especially resistant among them. We discussed possible implications for lightweight ciphers' design and cryptanalytic strength. Ideas for future work are proposed.

Acknowledgement

This paper was supported by National Natural Science Foundation of China (NSFC) under grants 61902414, 61672530, 62002370, Natural Science Foundation of Hunan Province under grant 2020JJ5667. Tomer Ashur is an FWO post-doctoral fellow under Grant Number 12ZH420N.

A Reported RX-Characteristics for Simon32 and Simeck32

Table 10 presents the 14-round RX-characteristic for SIMON32 found in Section 3. Table 11 presents the 20-round RX-characteristic for SIMECK32 found in Section 3.

Table 10: A 14-round RX-characteristics for SIMON32

Round	key RX-difference	data RX-difference
0	a380	(0000||a780)
1	7784	(0400||0000)
2	5505	(6780||0400)
3	5402	(0023||6780)
4	8000	(1008||0023)
5	9005	(c003||1008)
6	c000	(8000||c003)
7	8004	(0001||8000)
8	0001	(0000||0001)
9	0000	(0000||0000)
10	0000	(0000||0000)
11	0000	(0000||0000)
12	0004	(0000||0000)
13	c005	(0004||0000)
14		(c015||0004)
Prob.	1	2^{-32}

B Sparkle-like Round Constants

At 2019, Beierle et al. presented Schwaemm and Esch lightweight authenticated encryption and hashing using the Sparkle permutation family submitted to the NIST lightweight cryptography standardization process [6]. They chose the round constants c_i as follows:

$$c_0 = \texttt{b7e15162},\ c_1 = \texttt{bf715880},$$
$$c_2 = \texttt{38b4da56},\ c_3 = \texttt{324e7738},$$
$$c_4 = \texttt{bb1185eb},\ c_5 = \texttt{4f7c7b57},$$
$$c_6 = \texttt{cfbfa1c8},\ c_7 = \texttt{c2b3293d}.$$

In this paper, We truncated these 64-bit round constants into two 32-bits to fit our test parameters and cycle every 16 rounds (In fact, no more than 16 rounds of RX-characteristics were produced in the algorithm we tested.), *i.e.*, the Sparkle-like round constants used in this paper are:

Table 11: A 20-round RX-characteristic for SIMECK32

Round	key RX-difference	data RX-difference
0	0004	(0000||0004)
1	0000	(0000||0000)
2	0001	(0000||0000)
3	0002	(0001||0000)
4	0002	(0000||0001)
5	0005	(0003||0000)
6	0001	(0000||0003)
7	0002	(0002||0000)
8	000a	(0004||0002)
9	0002	(0000||0004)
10	0000	(0006||0000)
11	0013	(000a||0006)
12	000a	(0001||000a)
13	0004	(0002||0001)
14	0000	(0001||0002)
15	0001	(0000||0001)
16	0000	(0000||0000)
17	0002	(0000||0000)
18	0006	(0002||0000)
19	0007	(0000||0002)
20		(0005||0000)
Prob.	2^{-34}	2^{-26}

$$
\begin{aligned}
c_0 = c_{16} &= \texttt{b7e1},\ c_1 = c_{17} = \texttt{5162},\\
c_2 = c_{18} &= \texttt{bf71},\ c_3 = c_{19} = \texttt{5880},\\
c_4 = c_{20} &= \texttt{38b4},\ c_5 = c_{21} = \texttt{da56},\\
c_6 = c_{22} &= \texttt{324e},\ c_7 = c_{23} = \texttt{7738},\\
c_8 = c_{24} &= \texttt{bb11},\ c_9 = c_{25} = \texttt{85eb},\\
c_{10} = c_{26} &= \texttt{4f7c},\ c_{11} = c_{27} = \texttt{7b57},\\
c_{12} = c_{28} &= \texttt{cfbf},\ c_{13} = c_{29} = \texttt{a1c8},\\
c_{14} = c_{30} &= \texttt{c2b3},\ c_{15} = c_{31} = \texttt{293d}.
\end{aligned}
$$

C Using the Round Constants of Other Simon Variants

Similar to Section 4.2 we analyze Sim-(2,1,-,1), Sim-(2,1,-,3), Sim-(2,1,-,4), Sim-(2,1,-,5), Sim-(2,1,-,6), Sim-(2,1,-,7), Sim-(2,1,-,8) and compare them in Table 12 to Sim-(2,1,-,2). In line with Table 9 there seems to be a meaningful difference attributed to the round constants.

Table 12: A comparison of Simon-like ciphers with (B) = 1 and varying over Parameter (D). Entries marked with an asterisk are not necessarily optimal *i.e.*, an RX-characteristics covering the same number of rounds with better probabilities may still exist.

Rounds	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Sim-(2,1,-,1)	0	4	6	8	12	18*	26*	28*	32*
Sim-(2,1,-,2)	0	4	6	8	12	16	24*	28*	32*
Sim-(2,1,-,3)	0	4	6	10	12	22*	25*	27*	30*
Sim-(2,1,-,4)	0	4	6	10	12	22*	26*	30*	
Sim-(2,1,-,5)	0	4	6	10	12	22*	26*	30*	32*
Sim-(2,1,-,6)	0	4	6	10	12	22*	26*	30*	32*
Sim-(2,1,-,7)	0	4	6	10	14	20*	26*	30*	32*
Sim-(2,1,-,8)	0	5	9	12	20*	28*	32*		

References

1. Abdelraheem, M.A., Ågren, M., Beelen, P., Leander, G.: On the distribution of linear biases: Three instructive examples. In: CRYPTO. Lecture Notes in Computer Science, vol. 7417, pp. 50–67. Springer (2012)
2. Ashur, T.: [PATCH v2 0/5] crypto: SPECK support. Linux kernel mailing list (June 2018), `https://marc.info/?l=linux-crypto-vger&m=152788106609224&w=2`
3. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015)

4. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. ePrint Arch. 2013, 404 (2013)
5. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: Notes on the design and analysis of SIMON and SPECK. IACR Cryptol. ePrint Arch. 2017, 560 (2017)
6. Beierle, C., Biryukov, A., dos Santos, L.C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q.: Lightweight AEAD and hashing using the Sparkle permutation family. IACR Trans. Symmetric Cryptol. 2020(S1), 208–261 (2020)
7. Brummayer, R., Biere, A.: Boolector: An efficient SMT solver for bit-vectors and arrays. In: TACAS. Lecture Notes in Computer Science, vol. 5505, pp. 174–177. Springer (2009)
8. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: MILP-based automatic search algorithms for differential and linear trails for SPECK. In: FSE. Lecture Notes in Computer Science, vol. 9783, pp. 268–288. Springer (2016)
9. Gérault, D., Minier, M., Solnon, C.: Constraint programming models for chosen key differential cryptanalysis. In: CP. Lecture Notes in Computer Science, vol. 9892, pp. 584–601. Springer (2016)
10. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: CHES. Lecture Notes in Computer Science, vol. 6917, pp. 326–341. Springer (2011)
11. Ito, R., Shiba, R., Sakamoto, K., Liu, F., Isobe, T.: Bit-wise cryptanalysis on AND-RX permutation Friet-PC. Cryptology ePrint Archive, Report 2021/212 (2021), https://eprint.iacr.org/2021/212
12. Kamal, A.A., Youssef, A.M.: Applications of SAT solvers to AES key recovery from decayed key schedule images. In: SECURWARE. pp. 216–220. IEEE Computer Society (2010)
13. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 9215, pp. 161–185. Springer (2015)
14. Kondo, K., Sasaki, Y., Todo, Y., Iwata, T.: On the design rationale of SIMON block cipher: Integral attacks and impossible differential attacks against SIMON variants. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 101-A(1), 88–98 (2018)
15. Kranz, T., Leander, G., Wiemer, F.: Linear cryptanalysis: Key schedules and tweakable block ciphers. IACR Trans. Symmetric Cryptol. 2017(1), 474–505 (2017)
16. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 547, pp. 17–38. Springer (1991)
17. Liu, F., Isobe, T., Meier, W.: Automatic verification of differential characteristics: Application to reduced Gimli. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 12172, pp. 219–248. Springer (2020)
18. Liu, Y., Witte, G.D., Ranea, A., Ashur, T.: Rotational-XOR cryptanalysis of reduced-round SPECK. IACR Trans. Symmetric Cryptol. 2017(3), 24–36 (2017)
19. Liu, Y., Zhang, W., Sun, B., Rijmen, V., Liu, G., Li, C., Fu, S., Cao, M.: The phantom of differential characteristics. Des. Codes Cryptogr. 88(11), 2289–2311 (2020)
20. Lu, J.: Rotational-XOR cryptanalysis of Simon-like block ciphers (2020), https://github.com/JIN-smile/The-source-code-for-searching-compatible-RX-characteristics/

21. Lu, J., Liu, Y., Ashur, T., Sun, B., Li, C.: Rotational-XOR cryptanalysis of Simon-like block ciphers. In: ACISP. Lecture Notes in Computer Science, vol. 12248, pp. 105–124. Springer (2020)
22. Matsui, M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993)
23. Matsui, M.: New block encryption algorithm MISTY. In: FSE. vol. 1267, pp. 54–68. Springer (1997)
24. Mouha, N., Preneel, B.: Towards finding optimal differential characteristics for ARX: Application to Salsa20. Cryptology ePrint Archive, Report 2013/328 (2013)
25. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Inscrypt. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011)
26. de Moura, L.M., Bjørner, N.: Z3: an efficient SMT solver. In: TACAS. Lecture Notes in Computer Science, vol. 4963, pp. 337–340. Springer (2008)
27. Sadeghi, S., Rijmen, V., Bagheri, N.: Proposing an MILP-based method for the experimental verification of difference trails. IACR Cryptol. ePrint Arch. 2020, 632 (2020)
28. Song, L., Huang, Z., Yang, Q.: Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In: ACISP (2). Lecture Notes in Computer Science, vol. 9723, pp. 379–394. Springer (2016)
29. Sun, L., Wang, W., Wang, M.: MILP-aided bit-based division property for primitives with non-bit-permutation linear layers. IET Inf. Secur. 14(1), 12–20 (2020)
30. Sun, S., Gérault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., Hu, L.: Analysis of AES, SKINNY, and others with constraint programming. IACR Trans. Symmetric Cryptol. 2017(1), 281–306 (2017)
31. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014)
32. Wu, S., Wang, M.: Security evaluation against differential cryptanalysis for block cipher structures. IACR Cryptol. ePrint Arch. 2011, 551 (2011)
33. Wu, W., Zhang, L.: Lblock: A lightweight block cipher. In: ACNS. Lecture Notes in Computer Science, vol. 6715, pp. 327–344 (2011)
34. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10031, pp. 648–678 (2016)
35. Xin, W., Liu, Y., Sun, B., Li, C.: Improved cryptanalysis on SipHash. In: CANS. Lecture Notes in Computer Science, vol. 11829, pp. 61–79. Springer (2019)
36. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The SIMECK family of lightweight block ciphers. In: CHES. Lecture Notes in Computer Science, vol. 9293, pp. 307–329. Springer (2015)
37. Zhang, H., Wu, W.: Structural evaluation for Simon-like designs against integral attack. In: ISPEC. Lecture Notes in Computer Science, vol. 10060, pp. 194–208 (2016)