Received: 13 April 2021 | Revised: 29 December 2021 | Accepted: 4 March 2022

IET Information Security

DOI: 10.1049/ise2.12061

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Improved rotational-XOR cryptanalysis of Simon-like block ciphers

Jinyu Lu[1] | Yunwen Liu[1] | Tomer Ashur[2] | Bing Sun[1] | Chao Li[1]

[1]College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, China

[2]imec-COSIC KU Leuven, Leuven, The Netherlands

Correspondence

Yunwen Liu, College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, China.
Email: univerlyw@hotmail.com

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 61772545, 61902414, 62002370; FWO post-doctoral fellow, Grant/Award Number: 12ZH420N; Natural Science Foundation of Hunan Province, Grant/Award Number: 2020JJ5667

Abstract

Rotational-XOR (RX) cryptanalysis is a cryptanalytic method aimed at finding distinguishable statistical properties in Addition-Rotation-XOR-C ciphers, that is, ciphers that can be described only by using modular addition, cyclic rotation, XOR and the injection of constants. In this study, we extend RX-cryptanalysis to AND-RX ciphers, a similar design paradigm where the modular addition is replaced by vectorial bitwise AND; such ciphers include the block cipher families Simon and Simeck. We analyse the propagation of RX-differences through AND-RX rounds and develop a closed form formula for their expected probability. Inspired by the MILP verification model proposed by Sadeghi et al., we develop a SAT/SMT model for searching compatible RX-characteristics in Simon-like ciphers, that is, that there is at least one right pair of messages/keys to satisfy the RK-characteristics. To the best of our knowledge, this is the first model that takes the RX-difference transitions and value transitions simultaneously into account in Simon-like ciphers. Meanwhile, we investigate how the choice of the round constants affects the resistance of Simon-like ciphers against RX-cryptanalysis. Finally, we show how to use an RX-distinguisher for a key recovery attack. Evaluating our model we find compatible RX-characteristics of up to 20, 27 and 34 rounds with respective probabilities of 2^{-26}, 2^{-44} and 2^{-56} for versions of Simeck with block sizes of 32, 48 and 64 bits, respectively, for large classes of weak keys in the related-key model. In most cases, these are the longest published distinguishers for the respective variants of Simeck. In the case of Simon, we present compatible RX-characteristics for round-reduced versions of all 10 instances. We observe that for equal block and key sizes, the RX-distinguishers cover fewer rounds in Simon than in Simeck. Concluding the paper, we present a key recovery attack on Simeck 64 reduced to 28 rounds using a 23-round RX-characteristic.

KEYWORDS

ARX, rotational-XOR cryptanalysis, round constants, Simeck, Simon

1 | INTRODUCTION

Rotational-XOR cryptanalysis is a cryptanalytic technique for Addition-Rotation-XOR (ARX) ciphers proposed by Ashur and Liu [1]. RX-cryptanalysis generalises rotational cryptanalysis by investigating the influence of round constants on the probabilistic propagation of rotational pairs passing through the ARX operations.

The successful application of RX-cryptanalysis to Speck [2] reveals that the round constants sometimes interact in a constructive way between the rounds, that is, that a broken symmetry caused by a round constant in round i may be restored—either fully or partially—by another constant injection in round $j > i$. As a result, new designs such as [3] now show resistance to RX-cryptanalysis as part of their security argument.

AND-RX ciphers, defined as a counterpart of ARX ciphers where the modular addition is replaced by bitwise AND, are of contemporary interest owing to the design of the block cipher Simon [4] which was followed by other Simon-like

ciphers such as Simeck [5]. Since the AND-RX operations in Simon-like ciphers are bitwise, the resulting statistical properties of individual bits remain independent of the bit-position. We say that such properties are rotation-invariant.

To break rotation-invariant properties, the round constants are usually injected into the state. In the case of Simon and Simeck, the constants are injected to the key schedule and propagate into the round function via the round subkey.

Searching for (related key) differential characteristics in most models based on MILP or SAT/SMT only involves differential transitions and they are considered independent in different rounds. Previous studies have reported exceptions where such characteristics are incompatible [6]. Sadeghi et al. [7] developed a MILP model to verify the existence of right pairs with respect to a given characteristic. Observing that sometimes this set is empty, they conclude that the concatenation of valid transitions may result in an incompatible characteristic due to globally contradicting constraints.

Our contribution. In this study, we extend the idea of RX-cryptanalysis to AND-RX ciphers with applications to Simon and Simeck. The propagation of RX-differences through the AND-RX operations is fully analysed and a closed algebraic formula is derived for its expected probability. We show that an RX-difference with translation α passes through the vectorial AND operation with the same probability as that of an α XOR-difference. Due to the different nature of RX-differences and XOR-differences, characteristics of the former type would depend more on the key schedule and choice of round constants than those of the latter type.

Inspired by the MILP verification model proposed by Sadeghi et al., we develop a SAT/SMT model to search compatible RX-characteristics, that is, that they are consistent with at least one right pair of messages/keys. This model can automatically avoid the incompatible problem when searching RX-characteristics. Using the automatic search model we find compatible RX-distinguishers for all versions of Simeck and Simon; these results are summarised in Tables 1 and 2.

The compatible RX-characteristics we found for Simeck variants with block sizes of 32-, 48- and 64-bit improve the previously longest published results by 5, 8 and 9 rounds, respectively, albeit sometimes in a weaker attack model. When comparing for the same number of rounds, our results offer different trade-offs between the size of the affected key class and the distinguisher's probability. For Simon2n/4n we present compatible RX-characteristics of up to 14, 15, 16, 18 and 22 rounds, respectively. These are not the longest distinguishers for the respective versions of Simon.

Finally, we show how to exploit an RX-characteristic in a key recovery attack and apply it to reduced-round versions of Simeck64. We present an attack on 28 out of 44 rounds for Simeck 64/128 with data complexity 2^{36} and time complexity $2^{79.5}$. To the best of our knowledge, this is the first extension of RX-cryptanalysis to key recovery.

This study is an extension of [6, 8]. The original paper included the following contributions:

TABLE 1 Comparison of RX-characteristics for rotation offset $\gamma = 1$ with the longest published distinguishers for Simeck32, Simeck48 and Simeck64[a]

Cipher	Attack rounds	Data complexity	Size of week key class	Type	Reference
Simeck32	13	2^{32}	Full	DC	[9]
	15	2^{31}	Full	ID	[10]
		2^{24}	2^{54}	RKDC	[11]
		2^{18}	2^{44}	RX	Section 5.1
	19	2^{24}	2^{30}	RX	Section 5.1
	20	2^{26}	2^{30}	RX	Section 5.1
Simeck48	16	2^{24}	2^{80}	RKDC	[11]
		2^{18}	2^{68}	RX	Section 5.1
	18	2^{47}	Full	ID	[10]
		2^{22}	2^{66}	RX	Section 5.1
	19	2^{48}	Full	DC	[9]
		2^{24}	2^{62}	RX	Section 5.1
	27	2^{44}	2^{46}	RX	Section 5.1
Simeck64	21	2^{63}	Full	ID	[10]
	25	2^{64}	Full	DC	[9]
		2^{34}	2^{80}	RX	Section 5.1
	34	2^{56}	2^{58}	RX	Section 5.1

[a]Differential characteristics, ID, integral distinguishers; RKDC, related-key differential characteristics; RX, RX-characteristics.

TABLE 2 RX-distinguishers and their probabilities in round-reduced Simon instances

Simon	32/64	48/72	48/96	64/96	64/128	96/96	96/144	128/128	128/192	128/256
Rds.	14	14	15	16	16	15	18	16	20	22
Pr.	2^{-32}	2^{-48}	2^{-46}	2^{-64}	2^{-64}	2^{-93}	2^{-96}	2^{-98}	2^{-120}	2^{-120}

Note: The distinguishers work in a related-key model for all keys.

- An analysis of the propagation of RX-characteristics in AND-RX algorithms;
- A SAT/SMT automatic search model for RX-characteristics in Simon-like ciphers;
- Applications to Simeck and Simon32;
- A preliminary exploration of how different key schedules affect the probability and automatic search of RX-characteristics.

The extended version includes the following additional contributions:

- To the best of our knowledge, for the first time we developed a SAT/SMT model capturing the RX-difference transitions and value transitions simultaneously in Simon-like ciphers. This model finds the compatible

characteristics as well as the right pairs simultaneously, moreover, it can be directly applied in the detection of incompatible characteristics. In contrast, the method of Sadeghi et al. is aimed at detecting the incompatibility distinguisher. In this paper, we consider more about how to obtain compatible distinguishers directly. This model makes the cryptanalyst get rid of the complex, tedious and time-consuming process from a large number of searches to detection, making the analysis more concise and efficient. Simultaneously, it plays a very important role for the distinguisher attack and the key recovery attack.

- Based on this model, we re-evaluate Simeck32, Simeck48 and Simeck64. We see that previously published distinguishers cover up to 15, 19 and 25 rounds of Simeck32, Simeck48 and Simeck64, respectively, whereas our RX-characteristics improve the number of distinguished rounds by 5, 8 and 9 rounds, albeit for a smaller key class than previous results. Benchmarking for the same number of rounds, detecting our distinguishers requires fewer data. At the same time, we also evaluated all the 10 versions of Simon. For Simon32/64, we find longer RX-characteristics than previously presented.
- We evaluate a sequence of round constants and see how different round constants reflect in the resistance of the resulting cipher against RX-cryptanalysis. We find that the ability of the ciphers to resist RX-cryptanalysis can be affected by the hamming weight of the $\Delta_\gamma c$ and the specific choices of the round constants. To circumvent RX-differential attack, we give some suggestions to design the round constants.
- Moreover, we show how an RX-characteristic can be used to mount a key recovery attack, and apply this method to the reduced-round version Simeck64.

Organisation: In Section 2, we recall RX-cryptanalysis, the structure of Simon-like ciphers with emphasis on the key schedule of Simon and Simeck, and relevant previous work. In Section 3, we generalise RX-cryptanalysis to AND-RX algorithms. We give a closed-form algebraic formula for the probabilistic propagation of an RX-difference in AND-RX and for the special case of Simon-like ciphers. In Section 4, we devise an automated search model for finding good RX-characteristics in Simon-like ciphers. This model is then evaluated in Section 5 on all versions of Simeck and Simon. In Section 6, we investigate the effect of the round constants on the resistance of Simon-like ciphers against RX-cryptanalysis. In Section 7, we employ our distinguishers in key recovery attacks on reduced-round versions of Simeck64. The attacks work in the related-key chosen-plaintext model. Section 8 concludes the paper.

2 | PRELIMINARIES

In this section, we give a brief overview of the structure of Simon-like ciphers and recall the general idea of rotational-XOR cryptanalysis. Table 3 presents the notation we use.

TABLE 3 The notations used throughout the paper

Notation	Description
$x = (x_{n-1}, \ldots, x_1, x_0)$	Binary vector of n bits; x_i is the bit in position i with x_0 the least significant one
$x \odot y$	Vectorial bitwise AND between x and y
$x \oplus y$	Vectorial bitwise XOR between x and y
$x \Vert y$	Concatenation of x and y
$x \vert y$	Vectorial bitwise OR between x and y
$\mathrm{wt}(x)$	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
$(I \oplus S^\gamma)\ (x)$	$x \oplus S^\gamma(x)$
$\overline{x}$	Bitwise negation
$\underset{\sim}{x}$	$x \lll 1$

2.1 | Simon-like ciphers

Simon is a family of block ciphers following the AND-RX design paradigm, that is, 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 generalise the structure of Simon's round function with parameters different from the original ones.

2.1.1 | The round function

Simon is a family of lightweight block ciphers designed by the NSA in 2013. A member of the family is denoted by Simon2n/mn, to specify a block size of 2n for $n \in \{16, 24, 32, 48, 64\}$, and key size of mn for $m \in \{2, 3, 4\}$. The round function of Simon is defined as

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

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

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

Of particular interest in this paper is the Simeck family of lightweight block ciphers designed by Yang et al. [5], aiming at improving the hardware implementation cost of Simon. Simeck2n/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 Simeck2n throughout the study. The rotation amounts for all Simeck versions are $(a, b, c) = (5, 0, 1)$.

2.1.2 | The key schedule

The non-linear key schedule 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 $4n$-bit master key for Simeck2n. The registers of the key schedule 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

$$k^{r+1} = t^r$$

where

$$t^{r+3} = k^i \oplus f_{5,0,1}(t^r) \oplus c^r,$$

and the round constants are $c^r \in \{\texttt{0xfffc}, \texttt{0xfffd}\}$. A single round of Simeck is depicted in Figure 1a.

Simon, conversely, uses a linear key schedule to generate the round keys. Let $K = (k^{m-1}, \ldots, k^1, k^0)$ be a master key for Simon2n, where $k^i \in \mathbb{F}_2^n$. The sequence of round keys k^r is generated by

$$k^{i+m} = \begin{cases} k^r \oplus \left(I \oplus S^{-1}\right) S^{-3} k^{r+1} \oplus c^r, & \text{if } m = 2 \\ k^r \oplus \left(I \oplus S^{-1}\right) S^{-3} k^{r+2} \oplus c^r, & \text{if } m = 3 \\ k^r \oplus \left(I \oplus S^{-1}\right)\left(S^{-3} k^{r+3} \oplus k^{r+1}\right) \oplus c^r, & \text{if } m = 4 \end{cases}$$

where $c^r \in \{\texttt{0xfffc}, \texttt{0xfffd}\}$, and $0 \le r \le (T-1)$. A single round of Simon with $m = 4$ is depicted in Figure 1b.

2.1.3 | Previous work

The security of Simon-like ciphers has been widely explored over the last few years and a large number of cryptanalytic techniques were applied to it. To name just a few: linear cryptanalysis [12, 13], differential cryptanalysis [9, 12, 14], impossible differential cryptanalysis [15], related-key differential cryptanalysis [11], integral cryptanalysis and the division property [16–19]. For a comparison of our results with relevant previous work see Table 1.

2.2 | Rotational-XOR cryptanalysis

Rotational cryptanalysis [20, 21] is a related-key chosen-plaintext attack investigating the propagation of rotational pairs, that is, pairs of the form $(x, x \lll \gamma)$. This attack is thwarted when a constant that is not rotation-invariant, that is, a constant c such that $c \neq (c \lll \gamma)$ is injected into the rotational pair.

Rotational-XOR cryptanalysis [1] is a generalized attack method which takes these 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 considering just 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 [2] and the PRF SipHash [22].

3 | ROTATIONAL-XOR CRYPTANALYSIS OF AND-RX CONSTRUCTIONS

AND-RX constructions are similar in concept to ARX constructions where the non-linear operation (i.e., modular addition) is replaced with a vectorial bitwise AND. Since all operations are now bit oriented, such constructions are always rotation-invariant. More generally, they are structurally invariant under any affine transformation of the bit-indices as was shown in [12]. Superficially, it is believed that this invariance cannot be preserved over a large number of rounds if non-invariant constants are injected into the state since they will break the symmetry between bits in different positions. Despite their close relation to ARX constructions, the security of AND-RX ciphers against RX-cryptanalysis has not received much attention. Now we are set to rectify this omission.

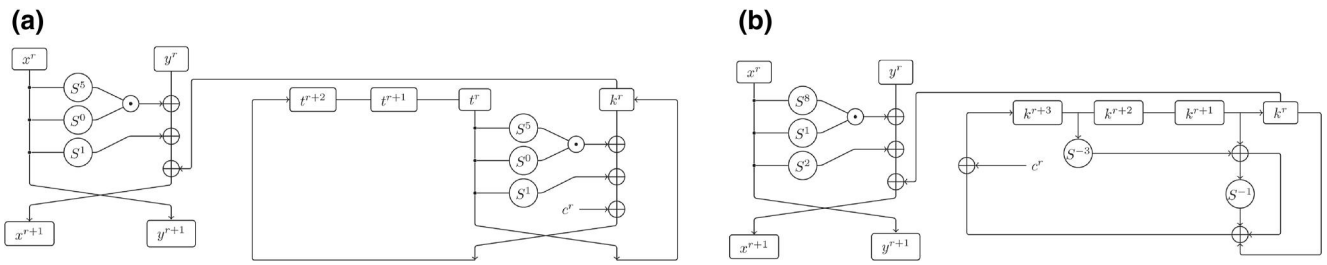

FIGURE 1 Illustration of the Simeck and Simon ciphers. (a) One round of Simeck. (b) One round of Simon with $m = 4$

3.1 | The expected probability of an RX-transition

In [1], an RX-pair was defined to be a rotational pair with rotational offset γ under translations δ_1 and δ_2, that is, it is the pair $(x \oplus \delta_1, (x \lll \gamma) \oplus \delta_2)$. We opt for a slightly different notation with x and $x' = (x \lll \gamma) \oplus \delta$, or $(x, (x \lll \gamma) \oplus \delta)$ as an RX-pair.

Definition 3.1 ([1] (adapted)). The RX-difference of x and $x' = (x \lll \gamma) \oplus \delta$ is denoted by

$$\boldsymbol{\Delta}_\gamma(x, x') = x' \oplus (x \lll \gamma),$$

where $\delta \in \mathbb{F}_2^n$ is a constant and γ is the rotational offset with $0 < \gamma < n$.

The propagation of an RX-difference $\Delta_r(x, x')$ through linear operations of the AND-RX structure is deterministic and follows these rules:

- **XOR.** For two input RX-pairs $(x, (x \lll \gamma) \oplus \delta_1)$ and $(y, (y \lll \gamma) \oplus \delta_2)$, their XOR is the RX-pair $(z, z') = (x \oplus y, ((x \oplus y) \lll \gamma) \oplus \delta_1 \oplus \delta_2)$;
- **Cyclic rotation by η bits.** The cyclic rotation of each of the values in $(x, (x \lll \gamma) \oplus \delta)$ by η bits is the RX-pair $(z, z') = (x \lll \eta, (x \lll (\gamma + \eta)) \oplus (\delta \lll \eta))$;
- **XOR with a constant c**. The XOR of a constant c to each of the values in the RX-pair $(x, (x \lll \gamma) \oplus \delta)$ is the RX-pair $(z, z') = (x \oplus c, (x \lll \gamma) \oplus \delta \oplus c)$, the corresponding RX-difference is $\Delta_\gamma c = c \oplus (c \lll \gamma)$

all with probability 1.

Intuitively, the bitwise nature of the AND operation restricts the propagation of an RX-difference compared to modular addition. When two rotational pairs enter into the vectorial AND operation, the rotational relation is preserved with probability 1 due to the localised nature of bit-oriented operations. If the inputs form an RX-pair with translation $\delta \neq 0$ the propagation of the RX-difference through the vectorial AND is probabilistic and its probability is given by the following theorem.

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. $\alpha, \beta \in \mathbb{F}_2^n$ are independent constants, and $0 < \gamma < n$. Then, for an output translation Δ (Δ is a constant) it holds that:

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

$$\Pr[(x \odot y) \oplus \Delta = (x \oplus \alpha) \odot (y \oplus \beta)], \tag{2}$$

that is, 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.

Proof To prove the theorem, we distribute the right hand side of (1) as

$$\begin{aligned}&((x \lll \gamma) \oplus \alpha) \odot ((y \lll \gamma) \oplus \beta)\\ &= ((x \odot y) \lll \gamma) \oplus ((x \lll \gamma) \odot \beta) \oplus\\ &\quad ((y \lll \gamma) \odot \alpha) \oplus (\alpha \odot \beta).\end{aligned}$$

Similarly, distributing the right hand side of (2) we get

$$(x \oplus \alpha) \odot (y \oplus \beta) = (x \odot y) \oplus (x \odot \beta) \oplus (y \odot \alpha) \oplus (\alpha \odot \beta).$$

Rewriting Theorem 1 as

$$\Pr[((x \odot y) \lll \gamma) \oplus \Delta = ((x \odot y) \lll \gamma) \oplus ((x \lll \gamma) \odot \beta) \oplus ((y \lll \gamma) \odot \alpha) \oplus (\alpha \odot \beta)] = \tag{3}$$

$$\Pr[(x \odot y) \oplus \Delta = (x \odot y) \oplus (x \odot \beta) \oplus (y \odot \alpha) \oplus (\alpha \odot \beta)], \tag{4}$$

the proof is completed by observing that $(x \odot y) \lll \gamma$, $x \lll \gamma$, and $y \lll \gamma$ have the same probability distribution as $x \odot y$, x, and y, respectively, due to the rotation-invariance of bit-oriented operations. □

Kölbl et al. showed in [12] that in the special case of Simon-like ciphers (e.g., Simon and Simeck) where $y = S^{a-b}(x)$, the difference propagation distribution (and thus, the RX-propagation distribution) is given by the following proposition.

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

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

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

Proof The proof for the difference propagation distribution was given in [12]. The case for RX-propagation follows then from Theorem 1. □

3.2 | Discussion

Based on Theorem 1, it can be seen that the RX-difference passes through the vectorial AND component of a cipher with the same probability as an XOR-difference. However, the resulting RX-characteristics are in general different from the corresponding (related-key) differential characteristics, due to the XOR of constants in the round function which affects the propagation.

It is interesting to see that in ARX ciphers, the probability for the rotational-transition part of the RX-transition is maximised with $2^{-1.415}$ when $\gamma \in \{1, n-1\}$ and decreases for other γ. Conversely, the same transition passes with probability 1 through the vectorial AND in AND-RX ciphers. In other words, a rotational pair would propagate with probability 1 through all AND-RX operations, but only with some probability $p < 1$ through the ARX operations. We conclude that in general, round constants are more critical in AND-RX constructions compared to ARX ones, and hence that the former are more susceptible to RX-cryptanalysis than the latter.

4 | AUTOMATED SEARCH OF RX-CHARACTERISTICS IN SIMON-LIKE CIPHERS

Similar to other statistical attacks, RX-cryptanalysis works in two phases: offline and online. In the offline phase, the adversary is searching for a distinguishable property with respect to the algorithm's structure. Having found such a property, the adversary tries to detect it from data collected in the online phase.

Automated search methods are a common way to assist finding such a property (i.e., Phase 1). The idea behind these tools is to model the search problem as a set of constraints and solve it using one of the available constraint solvers. For ciphers using Boolean and arithmetic operations, the search problem can be converted into a Boolean Satisfiability Problem (SAT) or a satisfiability module problem (SMT). The respective solver then returns an answer on whether all constraints can be satisfied simultaneously, and if the answer is positive it also returns a valid assignment. A number of ARX and AND-RX ciphers were studied using automatic search tools, in the context of differential cryptanalysis, linear cryptanalysis, division property and RX-cryptanalysis [2, 23–27].

In this section, we give a detailed description of an automatic search model for RX-characteristics in Simon-like ciphers.

4.1 | The common round function

From Theorem 1, we learn that the propagation of RX-differences through the AND operation follows a probabilistic rule, with a probability distribution as in Proposition 1. We use $\Delta_1 a^r$ and $\Delta_1 b^r$ to denote the two n-bit vectors representing RX-differences at the beginning of round r, and $\Delta_1 d^r$ the n-bit vector representing the RX-difference at the output of the vectorial AND at the same round. A schematic view of this notation is depicted in Figure 2.

Then, the following two Boolean equations should be satisfied simultaneously for the propagation of RX-differences through the vectorial AND to be valid

$$0 = \Delta_1 d^r \odot \Big(\overline{S^a(\Delta_1 a^r) \mid S^b(\Delta_1 a^r)}\Big); \tag{5}$$

$$0 = \Big(\Delta_1 d^r \oplus S^{a-b}(\Delta_1 d^r)\Big) \odot \Big(\overline{S^a(\Delta_1 a^r)} \odot S^{2a-b}(\Delta_1 a^r) \odot S^b(\Delta_1 a^r)\Big). \tag{6}$$

In simple words, (5) ensures that any active bit in $\Delta_1 d^r$ results from at least one active bit in the corresponding position of $\Delta_1 a^r$. If a bit $\Delta_1 d^r$ was activated by exactly one bit from $\Delta_1 a^r$, (6) ensures that either a second bit in $\Delta_1 d^r$ is active, or that another active bit in $\Delta_1 a^r$ had deactivated the said bit. This encodes the implicit expansion function, that is, the dependency between the bit in position i and that in position $i + a - b$ before they enter the vectorial AND.

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

$$w_d^r = wt\Big(\Big(S^a(\Delta_1 a^r) \mid S^b(\Delta_1 a^r)\Big) \oplus \Big(\overline{S^a(\Delta_1 a^r)} \odot S^{2a-b}(\Delta_1 a^r) \odot S^b(\Delta_1 a^r)\Big)\Big), \tag{7}$$

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

In addition, the propagation of an RX-difference through the linear operations is described by the following constraints:

$$\Delta_1 b^{r+1} = \Delta_1 a^r; \tag{8}$$

$$\Delta_1 a^{r+1} = \Delta_1 d^r \oplus \Delta_1 b^r \oplus S^c(\Delta_1 a^r) \oplus \Delta_1 k^r. \tag{9}$$

4.2 | The key schedule of Simeck

The key schedule of Simeck is modelled analogously to the round function. Let $\Delta_1 ka^r$, $\Delta_1 kb^r$ and $\Delta_1 kd^r$ be n-bit variables in round r which denote the left input RX-difference, the right input RX-difference, and the output RX-difference of the vectorial AND (see Figure 3a). As before, the following two Boolean equations should be satisfied simultaneously for the

propagation of RX-differences through the non-linear part of Simeck's key schedule to be valid:

$$0 = \Delta_1 kd^r \odot \left(\overline{S^a(\Delta_1 ka^r) \mid S^b(\Delta_1 ka^r)}\right); \tag{10}$$

$$0 = \left(\Delta_1 kd^r \oplus S^{a-b}(\Delta_1 kd^r)\right) \odot \left(\overline{S^a(\Delta_1 ka^r)} \odot S^{2a-b}(\Delta_1 ka^r) \odot S^b(\Delta_1 ka^r)\right), \tag{11}$$

with weight w_k^r set as

$$w_k^r = wt\left(\left(S^a(\Delta_1 ka^r) \mid S^b(\Delta_1 ka^r)\right) \oplus \left(\overline{S^a(\Delta_1 ka^r)} \odot S^{2a-b}(\Delta_1 ka^r) \odot S^b(\Delta_1 ka^r)\right)\right). \tag{12}$$

The propagation of RX-difference through the linear operations of the key schedule is modelled by the following constraints:

$$\Delta_1 kb^{r+1} = \Delta_1 ka^r; \tag{13}$$

$$\Delta_1 ka^{r+3} = \Delta_1 kd^r \oplus \Delta_1 kb^r \oplus S^c(\Delta_1 ka^r) \oplus \Delta_1 c^r. \tag{14}$$

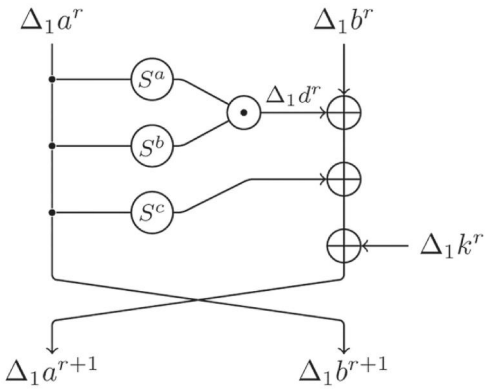

FIGURE 2 Notation of the RX-differences in the encryption function

Finally, the key schedule and the round function are linked via the following constraint:

$$\Delta_1 k^r = \Delta_1 kb^r. \tag{15}$$

4.3 | The key schedule of Simon

The key schedule of Simon is different. For brevity, we only describe the model for Simon2n/4n; translating this model to other variants is straightforward. Let $\Delta_1 ka^r$, $\Delta_1 ka^{r+1}$, $\Delta_1 ka^{r+2}$ and $\Delta_1 ka^{r+3}$ be n-bit variables denoting the RX-differences in the state of the key expansion function at the beginning of round r, and let $\Delta_1 ka^{r+4}$ denote the RX-difference fed back to the leftmost register at the end of the round (see Figure 3b); then, the propagation of the key RX-differences is given by

$$\Delta_1 ka^{r+4} = S^{-3}\left(\Delta_1 ka^{r+3}\right) \oplus \Delta_1 ka^{r+1} \oplus S^{-1}\left(S^{-3}\left(\Delta_1 ka^{r+3}\right) \oplus \Delta_1 ka^{r+1}\right) \oplus \Delta_1 ka^r \oplus \Delta_1 c^r, \tag{16}$$

and the injection of the subkey into the state in round r by

$$\Delta_1 k^r = \Delta_1 ka^r. \tag{17}$$

4.4 | The objective function

To evaluate the model, we define an objective function, that is, a quantity that the model is trying to optimise and which can be used to compare the 'quality' of different solutions. The original model in [2], which was the first model to search for RX-differences in ciphers with a non-linear key schedule, operated in two steps. First, a good key RX-characteristic was sought. Then, a good RX-characteristic was sought for the state with respect to the selected key RX-characteristic.

In this study, we take a different approach. Rather than considering the two search problems separately, we generate good RX-characteristics 'on-the-fly' without a priori fixing the key characteristic. We start by searching for RX-characteristics

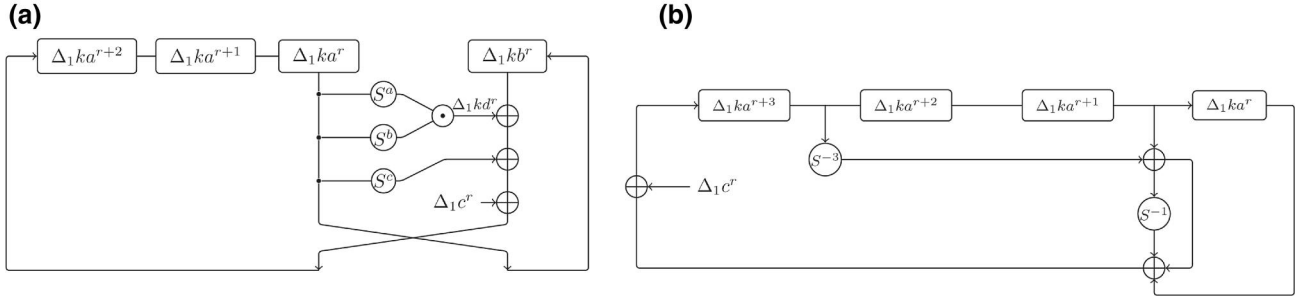

FIGURE 3 Notations of the RX-differences. (a) Notation of the RX-differences with a non-linear key schedule. (b) Notation of the RX-differences with a linear key schedule

minimising the total weight in both the data and key parts, namely $w_d + w_k$. Then, conditioned on the total weight $w_d + w_k$ fixed to the minimum found, we further minimise the weight in the data part w_d in order to improve the data complexity of the attack.

For Simon, this strategy would yield the same results as the strategy in [2] since $w_k^r = 0$ for all r due to the linear key schedule. The objective function for the R-round Simon model is expressed as

$$\begin{aligned} &\min(w)\text{s.t.}\\ &\quad \max(R) \quad \text{s.t.}\\ &\qquad (w = \left(\sum_{r=1}^{R} w_d^r\right)) \wedge (w \le 2n). \end{aligned} \tag{18}$$

For Simeck, we first observe that the key difference injected in round r is actually generated in round $r-4$ where its cost is 'paid'. As a result, the total probability of an R-round characteristic in the key schedule part only needs to take into account the cost of rounds 1 to $R-4$. Hence, we set the objective function as follows:

$$\begin{aligned} &\min(w_d)\ \text{s.t.}\\ &\quad \min(w)\ \text{s.t.}\\ &\qquad \max(R)\ \text{s.t.}\\ &\qquad\quad (w = \left(\sum_{r=1}^{R} w_d^r + \sum_{r=1}^{R-4} w_k^r\right)) \wedge (w \le 4n)\\ &\qquad\qquad \wedge \left(w_d = \sum_{r=1}^{R} w_d^r\right) \wedge (w_d \le 2n). \end{aligned} \tag{19}$$

4.5 | Compatibility

In [7], Sadeghi et al. presented a MILP model for outputting a solution with respect to a given RX-characteristic, that is, a pair of related keys and a right pair satisfying the said characteristic. They observed that some of the RX-characteristics in [6, 8] cannot produce right pairs with respect to any key due to global contradictions; such RX-characteristics are said to be incompatible.

In this study, we try to construct a model which can describe the RX-difference transitions and value transitions simultaneously. The basic idea is straightforward, the models to describe the RX-difference transitions and value transitions will be independently constructed. Then, construct a model of the difference-value relations in each round and use it to connect the RX-difference transitions and value transitions. Once such a model is constructed, the found characteristics are guaranteed to be valid.

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 sub-key k^{r+1} from the state of the key schedule in round r and the round constant c^r. Further, let k^r and $(k^r)'$ denote the n-bits sub-keys to round r, with respect to the master keys K and K'. Then, the following constraints should be satisfied for the value transitions of keys:

$$k^{r+1} = f_{ks}(t^{r+2}, t^{r+1}, t^r, k^r, c^r); \tag{20}$$

$$\left(k^{r+1}\right)' = f_{ks}\left((t^{r+2})', (t^{r+1})', (t^r)', (k^r)', c^r\right). \tag{21}$$

In order to describe the difference-value relations in key schedule we added the following constraints:

$$\Delta_1 k^r = \overleftarrow{k^r} \oplus (k^r)'; \tag{22}$$

$$\Delta_1 k^{r+1} = \overleftarrow{k^{r+1}} \oplus \left(k^{r+1}\right)'. \tag{23}$$

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

Then, the following constraints should be satisfied for the value transitions of messages:

$$\left(x^{r+1}, y^{r+1}\right) = h_{a,b,c}(x^r, y^r, k^r); \tag{24}$$

$$\left((x^{r+1})', (y^{r+1})'\right) = h_{a,b,c}((x^r)', (y^r)', (k^r)'). \tag{25}$$

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

$$\Delta_1 x^r = \overleftarrow{x^r} \oplus (x^r)'; \tag{26}$$

$$\Delta_1 y^r = \overleftarrow{y^r} \oplus (y^r)'; \tag{27}$$

$$\Delta_1 x^{r+1} = \overleftarrow{k^{r+1}} \oplus \left(x^{r+1}\right)'; \tag{28}$$

$$\Delta_1 y^{r+1} = \overleftarrow{y^{r+1}} \oplus \left(y^{r+1}\right)'. \tag{29}$$

5 | RX-DIFFERENTIAL CHARACTERISTICS IN Simeck AND SIMON

Now that we have a model for finding compatible RX-characteristics in AND-RX constructions, we can use an SMT solver to evaluate it. We describe the model using the

SMT-LIB language and apply the Boolector solver with several parameter settings. Our experiments were carried out on a laptop having an Intel Core i7-7700HQ CPU running at 2.80 GHz with an 8 GB RAM and a server with Intel Xeon(R) Core E5-2609 v2 CPU running at 2.50 GHz. The source code can be found in [28].

5.1 | Simeck

Using the above model, we found RX-characteristics that cover up to 20, 27 and 34 rounds for variants of Simeck with block size of 32, 48 and 64 bits, respectively. These results are presented in Table 4. Whereas distinguishers of similar length were presented in [6, 8], this section replaces some of them with compatible ones (for a discussion on compatibility, see Section 4.5). We further prove 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 gives a tight bound on the number of rounds that can be distinguished using RX-cryptanalysis.

Recalling the previous results in Table 1 we see that previously published distinguishers cover up to 15, 19 and 25 rounds of Simeck32, Simeck48 and Simeck64, respectively, whereas our RX-characteristics improve the number of distinguished rounds by 5, 8 and 9 rounds, albeit for a smaller key class than the previous results. Benchmarking for the same number of rounds, detecting our distinguishers requires fewer data.

5.1.1 | Experimental verification

To empirically validate our results, we implemented the 15-round RX-characteristic presented in Table 5. We first sample a random 64-bit master key $K = (k^3 \| k^2 \| k^1 \| k^0)$ and obtain its respective matching key $K' = S^1(K) \oplus (0001 \| 0004 \| 0008 \| 0014)$. We then check if the resulting subkeys satisfy the required RX-difference. If not, a new K is picked and the above process is repeated until a good pair (K, K') is found. This pair of related keys is used to encrypt 2^{32} plaintext pairs. For each encrypted plaintext pair, we check if the intermediate RX-differences match those of the RX-characteristic.

We sampled about $2^{33.6} = 2^{26.6+7}$ keys, out of which 2^7 satisfied the requested key RX-difference. For these keys, the average probability that a randomly selected plaintext satisfies the RX-characteristic was around $2^{-18.005}$. These figures confirm our claims.

TABLE 4 Weights of the best found RX-characteristics for round-reduced Simeck32, Simeck48 and Simeck64 with $\gamma = 1$[a]

SIMECK32											
Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Data	6	10	12	12	16	18	18	18	22	24	26
Key	8	12	12	18	18	20	28	32	30	34	34

SIMECK48													
Rounds	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Data	18	18	18	22	24	26	30	30	32	36	38	40	44
Key	20	28	32	30	34	34	36	40	44	46	46	48	50

SIMECK64													
Rounds	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**
Data	28	30	32	34	40	40	42	42	46	48	50	50	56
Key	40	44	46	48	50	54	58	60	64	66	70	72	70

[a]For each of the ciphers we report the results in three rows: number of distinguished rounds, weights of the round function part, and weights of the key schedule part. For instance, the best found RX-characteristic covering 20-round Simeck32 have a data probability of 2^{-26} and a key probability of 2^{-34}.

TABLE 5 A 15-round RX-characteristics in Simeck32/64

Round	RX-difference in key	RX-difference in data
0	0014	(0000‖0010)
1	0008	(0004‖0000)
2	0004	(0000‖0004)
3	0001	(0000‖0000)
4	0002	(0001‖0000)
5	0002	(0001‖0001)
6	0000	(0000‖0001)
7	0003	(0001‖0000)
8	0002	(0000‖0001)
9	0007	(0003‖0000)
10	0001	(0000‖0003)
11	0002	(0002‖0000)
12	0008	(0004‖0002)
13	0002	(0002‖0004)
14	0000	(0000‖0002)
15		(0002‖0000)
Prob.	2^{-26}	2^{-18}

5.2 | Simon

Interestingly, despite their similar structure, finding good RX-characteristics for Simon seems to be harder than for Simeck. The average run-time for finding solutions or proving unsatisfiability appears to be much longer in the former than the latter for the same number of rounds and block size. For the 10 instances of Simon, we found RX-distinguishers covering a range of 14–22 rounds; these results are presented in Table 6.

TABLE 6 The weights of the optimal and best-found RX-characteristic in round-reduced variants of all Simon versions with $\gamma = 1$[a]

Rounds	6	7	8	9	10	11	12	13	14	15	16	17	18
Simon32/64	0	4	6	10	14	20*	24*	30*	32*	-	-	-	-
Simon48/72	2	4	8	12	16	26*	36*	40*	48*	-	-	-	-
Simon48/96	0	4	6	10	14	24*	32*	32*	38*	46*	-	-	-
Simon64/96	4	8	10	16	18	24*	36*	40*	52*	54*	64*	-	-
Simon64/128	0	4	6	10	14	22*	34*	36*	40*	48*	64*	-	-
Rounds	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
Simon96/96	28*	40*	48*	64*	80*	93*	-	-	-	-	-	-	-
Simon96/144	16	26*	32*	40*	50*	66*	76*	84*	96*	-	-	-	-
Simon128/128	30*	40*	50*	60*	76*	92*	98*	-	-	-	-	-	-
Simon128/192	16*	30*	36*	40*	50*	66*	76*	84*	96*	108*	120*	-	-
Simon128/256	12*	24*	32*	40*	44*	56*	66*	70*	82*	90*	94*	104*	120*

[a]When an optimal solution is out-of-reach after a reasonable amount of time, we provide the best-found RX-characteristics labelled with *. For instance, the best found RX-characteristic covering 14-round Simon32/64 has a data probability of 2^{-32} for all key pairs with a specified difference.

TABLE 7 The full parameter set for Sim(A,B)

Parameter	(A) Cipher	(B) Round constants
1	Simeck32	$RC_{Simeck32}$
2	Simon32	$RC_{Simon32}$
3		Round counter
4		Alzette-like
5		Prince-like
6		0×5555/0×aaaa

6 | ON THE EFFECT OF THE ROUND CONSTANTS

Lightweight block ciphers often use a simple or even trivial key expansion algorithm to present a trade-off between suitable security and small implementations for resource-constrained devices. Using round constants to break the similarities between round functions is a typical countermeasure to protect block ciphers against self-similarity cryptanalysis. For instance, the key schedule of Simon and Simeck employ round constants specifically to eliminate slide properties and rotational attacks, different round constants in the key schedule of rectangle [29] prevent slide attacks and so on. Since the values of round constants do not affect the propagation of differences in differential cryptanalysis of ciphers, however, in RX-differential the XOR of round constants introduce RX-differences into the propagation. This implies that the effect of round constants cannot be ignored in RX-cryptanalysis, as opposed to differential cryptanalysis.

As shown in Section 3, it can be seen that the RX-difference $\Delta_\gamma(x, x')$ passes through the 'XOR with a constant c' operation is deterministic with probability 1 and the corresponding RX-difference is $\Delta_\gamma c = c \oplus (c \lll \gamma)$. There are two components in $\Delta_\gamma c$: (1) the rotational offset γ; and (2) the value of c. To understand how each of these two components affects the resistance of the resulting cipher to RX-cryptanalysis, we define additional variants:

- (A) The Simon-like ciphers. We consider ciphers with the same specification as Simon and Simeck except that the round constants vary.
- (B) The round constants used in the key expansion algorithm. Here we consider six types of constants, where the Alzette-like and Prince-like round constants are provided in Appendices A and B, respectively. The last type of round constants is 0×5555/0×aaaa, where we replace the alternatively round constants 0×fffc/0×fffd in Simeck by 0×5555/0×aaaa.

As shown in Table 7, a variant is denoted by Sim(A, B) where Sim means that the Simon-like ciphers and the tuple (A, B) define the controlled variables such that (A) defines the type of the cipher algorithm and (B) the round constants. For example, Simeck32 can be denoted by Sim(1, 1), Simon32 can be denoted by Sim(2, 2).

6.1 | The rotation offset γ

We begin by investigating the effect of rotation offset γ on the resistance against RX-cryptanalysis. When the round constant is given, the value of $\Delta_\gamma c$ with different rotation offset γ ($0 < \gamma < n$) can be calculated. To study the influence of the rotation offset, we first fix the parameters such that (A) = {1}, (B) = {1} and evaluate for $\gamma \in \{1, 2, \ldots, 15\}$, that is evaluate Simeck32 for $\gamma \in \{1, 2, \ldots, 15\}$.

The results are shown in Table 8. It can be seen that the length of the RX-characteristics is maximised with 20 rounds

TABLE 8 The effect of different rotation offset γ on the resistance of the Sim(1, 1) (Simeck32) against RX-cryptanalysis[a]

	Rounds	10	11	12	13	14	15	16	17	18	19	20
$\gamma = 1$	Data	6	10	12	12	16	18	18	18	22	24	26
	Key	8	12	12	18	18	20	28	32	30	34	34
$\gamma = 2$	Data	8	12	12	12	16	20	22	26	28	28	
	Key	8	10	16	20	24	28	28	28	32	36	
$\gamma = 3$	Data	10	12	14	16	20	22	24	22	24		
	Key	10	12	16	22	22	26	30	36	38		
$\gamma = 4$	Data	8	12	12	16	20	22	22	26	24		
	Key	10	10	14	18	22	24	28	30	36		
$\gamma = 5$	Data	10	12	14	18	20	22	28				
	Key	6	14	18	20	26	30	30				
$\gamma = 6$	Data	10	14	16	20	22	24	26				
	Key	12	14	20	22	28	36	38				
$\gamma = 7$	Data	10	14	16	20	24	26*					
	Key	14	16	20	24	30	38*					
$\gamma = 8$	Data	8	14	16	18	18	24					
	Key	16	16	22	28	36	38					
$\gamma = 9$	Data	12	14	16	20	24	26*					
	Key	8	16	20	24	30	38*					
$\gamma = 10$	Data	10	14	14	20	18	22	26*				
	Key	12	14	20	22	34	38	38*				
$\gamma = 11$	Data	8	12	14	18	20	22	26				
	Key	12	14	18	20	26	30	32				
$\gamma = 12$	Data	10	12	12	16	20	22	22	24	24		
	Key	6	10	14	18	22	24	28	34	36		
$\gamma = 13$	Data	10	12	14	16	20	20	20	22	26		
	Key	10	12	16	22	22	28	34	36	38		
$\gamma = 14$	Data	8	12	12	16	20	24	22	20	24		
	Key	8	10	16	18	22	24	28	36	38		
$\gamma = 15$	Data	6	10	12	12	16	18	16	18	22	24	26
	Key	8	12	12	18	18	20	30	32	30	34	34

[a]For each of the γ we report the results in two rows: probability of the round function part, and probability of the key schedule part. When an optimal solution is out-of-reach after a reasonable amount of time, we provide the best-found RX-characteristics labelled with *. For instance, the best found RX-characteristic covering 16-round $\mathrm{Sim}_{\gamma=10}(1, 1)$ have a data probability of 2^{-26} and a key probability of 2^{-38}.

when $\gamma \in \{1, 15\}$. For simplicity, we choose $\gamma = 1$ in this paper. As shown in Table 9, the hamming weight of $\Delta_1 c$ and $\Delta_{15} c$ is lower than the other case and we conjecture that the RX-characteristics penetrate more rounds when the hamming weight of the RX-difference $\Delta_\gamma c$ is lower. In fact, compared with differential cryptanalysis, the round constants generate RX-differences $\Delta_\gamma c$ with a large hamming weight will make the hamming weight of the input difference of the next round become larger, which makes the propagation probability become lower, as predicted by Proposition 1.

To further verify the conjecture, we choose round constants such that the resulting RX-difference $\Delta_\gamma c$ has hamming weight of either 16 or 0, for instance the constants 0×5555/0×aaaa. The result is shown in Table 10 and we can clearly see that the significant difference for Sim(1, 6) with odd γ and Sim(1, 6) with even γ.

Meanwhile, note that for an algorithm to resist against RX-cryptanalysis, it needs to resist all possible options of γ. In other words, the new variant Sim(1, 6) is more vulnerable to RX-cryptanalysis than Simeck32 under some rotational amounts. This means that the ability of the ciphers to resist RX-cryptanalysis can be affected by the hamming weight of the $\Delta_\gamma c$.

6.2 | The constant value c

From Section 6.1, we can see that the ability of the ciphers to resist RX-cryptanalysis can be affected by the hamming weight of the $\Delta_\gamma c$ when c is fixed. In this section, we fix the rotation offset $\gamma = 1$ to determine the effect of specific round constants in Simon-like ciphers. We first consider a sequence of variants Sim(1, 2), Sim(1, 3), Sim(1, 4) and Sim(1, 5). These variants differ from Simeck32 in the round constants where the constants are $\mathrm{RC}_{\mathrm{Simon32}}$, the round counter, Alzette-like and Prince-like round constants, respectively. Similarly, we consider the variants Sim(2, 1), Sim(2, 3), Sim(2, 4) and Sim(2, 5).

The results are presented in Tables 11 and 12. When the parameter (B) takes a value from {1, 2, 3}, the optimal RX-distinguishers have similar strength in the number of covered rounds. However, there is a significant difference for (B) ∈ {4, 5} when using Alzette-like and Prince-like constants instead of the round constants of Simeck, Simon and Speck which provide stronger resistance against RX-cryptanalysis. This means that the resistance of the ciphers against RX-cryptanalysis can be also affected by the specific choices of the round constants.

Since the propagation of an RX-difference through the round constant in round r is modelled by injecting a difference $\Delta_\gamma c^r$, where $\Delta_\gamma c^r = c^r \oplus (c^r \lll \gamma)$. To circumvent RX-differential attack, we suggest to design the round constants that satisfy the following conditions. (1) The hamming weights of c^r and $\Delta_\gamma c^r$ are as large as possible; (2) zeroes and ones in c^r and $\Delta_\gamma c^r$ are distributed more evenly; (3) the round constants are independent of each other. Under such conditions, the cipher is more likely to have a stronger resistance against RX-cryptanalysis.

7 | A KEY RECOVERY ATTACK FROM AN RX-CHARACTERISTIC

In this section, we use a 23-round RX-characteristic in a 28-round key recovery attack on Simeck64. This key recovery algorithm is similar to that of other statistical attacks. First, a

TABLE 9 The value of $\Delta_\gamma c$ corresponding to different round constants 0×fffc, 0×fffd, 0×5555, 0×aaaa

c\γ	1	2	3	4	5	6	7	8
0×fffc	0×0005	0×000f	0×001b	0×0033	0×0063	0×00c3	0×0183	0×0303
0×fffd	0×0006	0×000a	0×0012	0×0022	0×0042	0×0082	0×0102	0×0202
0×aaaa	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000
0×5555	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000
c\γ	9	10	11	12	13	14	15	
0×fffc	0×0603	0×0c03	0×1803	0×3003	0×6003	0×c003	0×8002	
0×fffd	0×0402	0×0802	0×1002	0×2002	0×4002	0×8002	0×0003	
0×aaaa	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000	0×ffff	
0×5555	0×ffff	0×0000	0×ffff	0×0000	0×ffff	0×0000	0×ffff	

TABLE 10 The effect of different rotation offset γ on the resistance of the Sim(1, 6) against RX-cryptanalysis

Rounds	10	11	12	13	14	15	16	17	18	19	20	21
$\gamma = 1$												
Data	22*	20*	24*									
Key	22*	34*	38*									
$\gamma = 2$												
Data	8	8	10	12	14	16	18	20	20	22	24	26
Key	8	12	14	14	16	18	22	24	30	32	36	38

TABLE 11 Fix the parameter (A) = 1, and evaluate the effect of different round constants [Parameter (B)] on the resistance of the cipher against RX-cryptanalysis with $\gamma = 1$

Rounds	10	11	12	13	14	15	16	17	18	19	20
Sim(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)											
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, 3)											
Data	6	10	12	12	16	18	18	22	22	26	
Key	8	12	12	18	18	22	28	30	36	34	
Sim(1, 4)											
Data	18	18	26*								
Key	20	28	30*								
Sim(1, 5)											
Data	16	22*	22*								
Key	22	24*	32*								

*Indicates when an optimal solution is out-of-reach after a reasonable amount of time, we provide the best-found RX-characteristics.

TABLE 12 : Fix the parameter (A) = 2, and evaluate the effect of different round constants [Parameter (B)] on the resistance of the cipher against RX-cryptanalysis with $\gamma = 1$

Rounds	6	7	8	9	10	11	12	13	14
Sim(2, 1)	0	4	6	10	14	22*	26*	28*	
Sim(2, 2)	0	4	6	10	14	20*	24*	30*	32*
Sim(2, 3)	0	4	6	10	15	24*	26*	30*	
Sim(2, 4)	0	5	9	13	18*	25*	31*		
Sim(2, 5)	0	4	8	13	19*	24*	30*		

*Indicates when an optimal solution is out-of-reach after a reasonable amount of time, we provide the best-found RX-characteristics.

set of plaintexts is encrypted to cover all the distinguisher rounds plus a few extra ones. Then, enough key bits are guessed to excavate the distinguisher under the assumption that only right guesses will allow to detect it, whereas wrong guesses will bury it deeper in. In the context of linear cryptanalysis this is called the wrong-key-randomisation hypothesis due to Harper et al. [30] and we believe that it is reasonable to extend the term also to differential- and RX-cryptanalysis.

We begin by describing the attack algorithm then continues to apply it to Simeck64.

7.1 | Attack procedure

Let $Enc_k^r(P)$ be an r-round encryption of a plaintext p under a key k and let D_d be an RX-characteristic with probability p covering $r_1 < r$ rounds of the cipher; D_k is its corresponding RX-characteristic for the key part. The attack procedure is as follows: for a small constant q we encrypt $q \cdot p^{-1}$ plaintext pairs

$$C = Enc_K^{r_1+r_2}(P),$$
$$C' = Enc_{K'}^{r_1+r_2}(P').$$

such that (p, p') are chosen plaintexts satisfying the input RX-difference of D_d, and (C, C') their corresponding ciphertexts after being encrypted over $r_1 + r_2$ rounds under the related keys (K, K') following the RX-characteristic D_k. Then, to detect the distinguisher, the key bits of (k, k') involved in the r_2 rounds are guessed, and the ciphertexts are partially decrypted to expose the bits involved in the output RX-difference of D_d. If indeed the RX-difference of the partially decrypted ciphertexts matches that of D_d, the guessed bits are put forward as candidates to be the right guess. Adjusting the small constant c allows controlling the number of candidate keys at the end of the attack.

Since k and k' are related keys, a guess for a bit-value in one fully determines the corresponding bit in the other. However, in non-linear key schedules some bit-relations are masked by the non-linear operations; thus a second guess is required.

The data complexity of the attack is $O(q \cdot p^{-1})$, and the time complexity is $\frac{r_2}{r_1+r_2} \cdot 2^b \cdot 2^\kappa + 2^{mn-\kappa}$ where q is a small constant, p is the probability of the RX-characteristic, 2^h is the number of plaintext pairs remaining after the filtering phase, κ is the number of guessed bits, and mn is the size of the master key.

7.2 | Testing the attack procedure

To demonstrate the effectiveness of the attack, we implement a key recovery attack on 12-round Simeck32 using a 9-round RX-distinguisher:

$$D_d : (0020, 0464) \xrightarrow{9} (0000, 0000),$$
$$D_k : (0000, 0006, 0008, 0014) \xrightarrow{5} (0000, 0000, 0000, 0002).$$

The probabilities in the encryption part and the key schedule part are 2^{-12} and 2^{-10}, respectively. Note that they are not the optimal RX-characteristics for 9-round Simeck32.

The characteristic is extended by three rounds to obtain the difference pattern shown in Table 13. According to Appendix C, the process of key guessing is shown in Table 14, that is, the required key bits are $k^{11} = \{4, 5, 6, 10, 11, 12\}$ to verify the RX-difference at the beginning of round 9.

Using 2^{15} plaintext pairs with input RX-difference (0020,0464), we collect the encrypted pairs after 12 rounds under a pair of keys

$$K = (E56F, 221F, 4E01, 9C61)$$
$$K' = (CADF, 4438, 9C0A, 38D7)$$

which satisfies the RX-distinguisher D_k. On guessing the six key bits in Table 14 and setting arbitrary values in the other key bits of k^{11}, k^{10} and k^9, we decrypt the last three rounds and count the number of right pairs for the output difference (0000,0000).

In 16 experiments, we observe that the number of right pairs is 8, which matches the attack hypothesis: $2^{15} \times 2^{-12} = 2^3$. We further observe that in all 16 experiments, wrong keys always result in an insignificant number of right pairs.

7.3 | Attacking 28-round Simeck64

We now present a key recovery attack on round-reduced Simeck64. When the choice of the key of the cryptosystem is restricted to a weak key class, the attack succeeds if it is faster than the exhaustive search over this restricted key-class. And the larger the weak key class, the better the attack would be. Thus, our starting point is RX-characteristic for Simeck64 covering 23 rounds with probability 2^{-34}, for a weak key class of size 2^{84}. The characteristic in the data part is

$$D_d : (00000000, 00000113) \xrightarrow{23} (00000001, 00000002)$$

and in the key schedule it is

$$D_k : (00000000, 00000004, 00000008, 00000117) \xrightarrow{19} (00000007, 00000002, 00000000, 00000000).$$

TABLE 13 Truncated RX-differential of bits obtained by extending the path of 9-round Simeck32 $(0020, 0464) \xrightarrow{9} (0000, 0000)$ in the bottom direction

0	$\Delta_1 L^0$	0000 0000 0010 0000
	$\Delta_1 R^0$	0000 0100 0110 0100
	$\Delta_1 k^0$	0000 0000 0001 0100
9 rounds		
9	$\Delta_1 L^9$	0000 0000 0000 0000
	$\Delta_1 R^9$	0000 0000 0000 0000
	$\Delta_1 k^9$	0000 0000 0000 0111
10	$\Delta_1 L^{10}$	0000 0000 0000 0111
	$\Delta_1 R^{10}$	0000 0000 0000 0000
	$\Delta_1 k^{10}$	0000 0000 0000 0101
11	$\Delta_1 L^{11}$	0000 0000 ***0 1***
	$\Delta_1 R^{11}$	0000 0000 0000 0111
	$\Delta_1 k^{11}$	0000 0000 0000 0101
12	$\Delta_1 L^{12}$	000* **0* ***1 ****
	$\Delta_1 R^{12}$	0000 0000 ***0 1***

TABLE 14 The process of key guessing for the key recovery attack on the 12-round Simeck32 using the 9-round RX-distinguisher

$\Delta_1 R_j^{11}$	0000 0000 0000 0111	Guessing k_{10}^{11}, k_{11}^{11}, k_{12}^{11}
$\Delta_1 R_{j-5}^{11}$	0000 0000 1110 0000	Guessing k_4^{11}, k_5^{11}, k_6^{11}
$\Delta_1 R_j^{10}$	0000 0000 0000 0000	No guessing required
$\Delta_1 R_{j-5}^{10}$	0000 0000 0000 0000	No guessing required
$\Delta_1 k_j^{11}$	0000 0000 0000 0101	No guessing required
$\Delta_1 k_j^{10}$	0000 0000 0000 0101	No guessing required
$\Delta_1 k_j^{9}$	0000 0000 0000 0111	No guessing required

TABLE 15 Truncated RX-differential of bits obtained by extending the path of 23-round Simeck64 $(00000000, 00000113) \xrightarrow{23} (00000001, 00000002)$ in the bottom directions

0	$\Delta_1 L^0$	0000 0000 0000 0000 0000 0000 0000 0000
	$\Delta_1 R^0$	0000 0000 0000 0000 0000 0001 0001 0011
	$\Delta_1 k^0$	0000 0000 0000 0000 0000 0001 0001 0111
23 rounds		
23	$\Delta_1 L^{23}$	0000 0000 0000 0000 0000 0000 0000 0001
	$\Delta_1 R^{23}$	0000 0000 0000 0000 0000 0000 0000 0010
	$\Delta_1 k^{23}$	0000 0000 0000 0000 0000 0000 0000 0101
24	$\Delta_1 L^{24}$	0000 0000 0000 0000 0000 0000 00*0 010*
	$\Delta_1 R^{24}$	0000 0000 0000 0000 0000 0000 0000 0001
	$\Delta_1 k^{24}$	0000 0000 0000 0000 0000 0000 0*00 00*1
25	$\Delta_1 L^{25}$	0000 0000 0000 0000 0000 0*00 ***0 1***
	$\Delta_1 R^{25}$	0000 0000 0000 0000 0000 0000 00*0 010*
	$\Delta_1 k^{25}$	0000 0000 0000 0000 0000 0000 ***0 1***
26	$\Delta_1 L^{26}$	0000 0000 0000 0000 *00* **0* ***1 ****
	$\Delta_1 R^{26}$	0000 0000 0000 0000 0000 0*00 ***0 1***
	$\Delta_1 k^{26}$	0000 0000 0000 0000 0000 0000 *0*0 1*0*
27	$\Delta_1 L^{27}$	0000 0000 000* 00** *0** **** **** ****
	$\Delta_1 R^{27}$	0000 0000 0000 0000 *00* **0* ***1 ****
	$\Delta_1 k^{27}$	0000 0000 0000 0000 0000 *000 ***0 0***
28	$\Delta_1 L^{28}$	0000 00*0 0*** 0*** **** **** **** ****
	$\Delta_1 R^{28}$	0000 0000 000* 00** *0** **** **** ****

A full description of this characteristic can be found in Table E3 of Appendix E.

Recall from Figures 1b and 2 that k^r is the round key in round r $(0 \leq r \leq r_1 + r_2)$, $\Delta_1 K^r$ is the RX-difference of the round key, $\Delta_1 a^r$ $(\Delta_1 b^r)$ is the input RX-difference in the left (right) part of the state. Extending D_d in the bottom direction as described in Table 15 we obtain the 28-round RX-differential which will be used in the attack.

Collection Phase We use a set ϱ of 2^{36} plaintext pairs (p, p'), such that their RX-difference is (00000000, 00000113). We ask for the 28-round encryption of all the pairs in ϱ, under the related-key pair (K, K′) in the weak key class to obtain their respective ciphertexts

$$C = Enc_K^{28}(P),$$
$$C' = Enc_{K'}^{28}(P').$$

Filtering Phase Observing that the truncated output RX-difference after 28 rounds is preserved in 23 of the ciphertext bits, we directly filter some of the wrong pairs, leaving $2^{36-23} = 2^{13}$ pairs to be used in the key guessing phase.

Key Guessing Phase We guess the necessary round key bits for partial decryption and verify the RX-difference at the end of round 23. A total 50 bits in k^{24}, k^{25}, k^{26}, k^{27} must be guessed along with 19 additional bits of (k^{24}), $(k^{25})'$, $(k^{26})'$, $(k^{27})'$. The guessed key bits of k and k' are (A detailed deduction on the guessed key bits in RX-attacks is shown in Appendix C)

$$\begin{aligned} k^{24} &= \{0, 4, 5, 26\}, \\ k^{25} &= \{0, 1, 4, 5, 6, 9, 26, 28, 31\}, \\ k^{26} &= \{0, 1, 4, 5, 6, 7, 9, 10, 11, 14, 26, 27, 28, 29, 31\}, \\ k^{27} &= \{0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 19, 26, 27, \\ &\quad 28, 29, 30, 31\}, \end{aligned}$$

and

$$\begin{aligned} (k^{24})' &= \{1, 6\}, \\ (k^{25})' &= \{0, 1, 2, 5, 6, 7\}, \\ (k^{26})' &= \{0, 2, 5, 7\}, \\ (k^{27})' &= \{0, 1, 2, 5, 6, 7, 11\}. \end{aligned}$$

Attack Complexity The time complexity of this attack is $\frac{5}{28} \cdot 2^{13} \cdot 2^{(50+19)} + 2^{128-50} \approx 2^{79.5}$, with data complexity $O(2^{36})$ and the attack covers 28 rounds of Simeck64.

8 | CONCLUSION

This study generalised the idea of rotational-XOR cryptanalysis to AND-RX ciphers by showing that an RX-difference has the same propagation probability as a corresponding XOR-difference going through the same function. Especially, we present a novel model capturing compatible distinguishers directly. As far as we know, this is the first SAT/SMT model to search for an RX-characteristic involving the value transitions in Simon-like ciphers. We found RX-characteristics for the Simeck family covering up to 20, 27 and 34 rounds for block sizes 32, 48, 64, respectively. These are the longest distinguishers for this cipher family. For the Simon family we found RX-characteristics for 14 rounds of Simon32/64; 14 and 15 for Simon48 with key sizes 72 and 96, respectively; 16 rounds for all versions of Simon64, 15 and 18 rounds for Simon96 with key sizes 96 and 144, respectively; and 16, 20 and 22, for Simon128 with key sizes 128, 192 and 256, respectively. We then studied how different round constants affect the resistance of Simon-like ciphers against RX-cryptanalysis. Moreover, we presented for the first time a procedure for using an RX-characteristic in a key recovery attack and applied it to 28-round Simeck64.

As future work, since both RX-cryptanalysis and related-key differential cryptanalysis are suitable for weak key environments and they are the variant of differential cryptanalysis, we recommend future work to further explore the differences between them. Also, we consider the search for longer distinguishers on all versions of Simon.

ACKNOWLEDGEMENTS

This study was supported by the National Natural Science Foundation of China (NSFC) under grants 61902414, 61772545 and 62002370, and the Natural Science Foundation of Hunan Province under grant 2020JJ5667. Tomer Ashur is an FWO post-doctoral fellow under Grant Number 12ZH420N.

CONFLICT OF INTEREST

All authors do not have a conflict of interest to disclose.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Jinyu Lu https://orcid.org/0000-0002-7299-0934

REFERENCES

1. Ashur, T., Liu, Y.: Rotational cryptanalysis in the presence of constants. IACR Trans. Symmet. Cryptol. 1(1), 57–70 (2016)
2. Liu, Y., et al.: Rotational-XOR cryptanalysis of reduced-round SPECK. IACR Trans. Symmet. Cryptol. 3(3), 24–36 (2017)
3. Koo, B., et al.: A family of lightweight block ciphers for resource-constrained devices. Lect. Notes Comput. Sci. 10779, 3–25 (2017)
4. Beaulieu, R., et al.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. 2013, 1–45 (2013). http://eprint.iacr.org/2013/404
5. Yang, G., et al.: The SIMECK family of lightweight block ciphers. Lect. Notes Comput. Sci. 9293, 307–329 (2015)
6. Lu, J., et al.: Rotational-XOR cryptanalysis of Simon-like block ciphers. Lect. Notes Comput. Sci. 12248, 105–124. Springer (2020)
7. Sadeghi, S., Rijmen, V., Bagheri, N.: Proposing an MILP-based method for the experimental verification of difference trails. IACR Cryptol. 2020, 632 (2020). https://eprint.iacr.org/2020/632
8. Lu, J., et al.: Rotational-XOR cryptanalysis of Simon-like block ciphers. IACR Cryptol. 2020, 486 (2020)
9. Liu, Z., Li, Y., Wang, M.: Optimal differential trails in SIMON-like ciphers. IACR Trans. Symmet. Cryptol. (1), 358–379 (2017)
10. Wang, S., et al.: MILP-aided method of searching division property using three subsets and applications. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology—ASIACRYPT 2019. Lecture Notes in Computer Science, pp. 398–427. Springer (2019)
11. Wang, X., et al.: Automatic search for related-key differential trails in SIMON-like block ciphers based on MILP. Lect. Notes Comput. Sci. 11060, 116–131 (2018)
12. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. Lect. Notes Comput. Sci. 1, 161–185 (2015)
13. Liu, Z., Li, Y., Wang, M.: The security of SIMON-like ciphers against linear cryptanalysis. IACR Cryptol. 576, 1–27 (2017)
14. Beierle, C.: Pen and paper arguments for SIMON and SIMON-like designs. Lect. Notes Comput. Sci. 9841, 431–446 (2016)
15. Sadeghi, S., Bagheri, N.: Improved zero-correlation and impossible differential cryptanalysis of reduced-round SIMECK block cipher. IET Inf. Secur. 12(4), 314–325 (2018)
16. Kondo, K., et al.: On the design rationale of SIMON block cipher: integral attacks and impossible differential attacks against SIMON variants. IEICE Trans. 101-A(1), 88–98 (2018)
17. Todo, Y., Morii, M.: Bit-based division property and application to SIMON family. Lect. Notes Comput. Sci. 9783, 357–377 (2016)
18. Wang, Q., et al.: Cryptanalysis of reduced-round SIMON32 and SIMON48. Lect. Notes Comput. Sci. 8885, 143–160 (2014)
19. Zhang, H., Wu, W.: Structural evaluation for SIMON-like designs against integral attack. Lect. Notes Comput. Sci. 10060, 194–208 (2016)
20. Khovratovich, D., Nikolić, I.: Rotational cryptanalysis of arx. In: International Workshop on Fast Software Encryption, pp. 333–346. Springer (2010)
21. Khovratovich, D., et al.: Rotational cryptanalysis of arx revisited. In: International Workshop on Fast Software Encryption, pp. 519–536. Springer (2015)
22. Xin, W., et al.: Improved cryptanalysis on SipHash. Lect. Notes Comput. Sci. 11829, 61–79 (2019)
23. Bagherzadeh, E., Ahmadian, Z.: MILP-based automatic differential search for LEA and HIGHT block ciphers. IET Inf. Secur. 14(5), 595–603 (2020)
24. Liu, Y., Wang, Q., Rijmen, V.: Automatic search of linear trails in ARX with applications to SPECK and Chaskey. In: International Conference on Applied Cryptography and Network Security - ACNS 2016, pp. 485–499. Springer (2016)
25. Sun, L., et al.: MILP-aided bit-based division property for ARX-based block cipher. IACR Cryptol. 1101 (2016)
26. Sun, L., et al.: MILP-aided bit-based division property for ARX ciphers. Sci. China Inf. Sci. 61(111), 1–3 (2018)
27. Sun, L., Wang, W., Wang, M.: Automatic search of bit-based division property for ARX ciphers and word-based division propertyASIACRYPT (1). Lect. Notes Comput. Sci. 10624, 128–157 (2017)
28. Lu, J.: Rotational-XOR cryptanalysis of Simon-like block ciphers (2020). https://github.com/JIN-smile/Simon32-and-Simeck32
29. Zhang, W., et al.: Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. Sci. China Inf. Sci. 58(12), 1–15 (2015)
30. Harpes, C., Kramer, G.G., Massey, J.L.: A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. Lect. Notes Comput. Sci. 921, 24–38 (1995)
31. Beierle, C., et al.: Alzette: a 64-bit arx-box. In: Annual International Cryptology Conference, pp. 419–448. Springer (2020)
32. Beierle, C., et al.: Schwaemm and Esch: lightweight authenticated encryption and hashing using the sparkle permutation family. NIST Round. 2 (2019)
33. Borghoff, J., et al.: Prince—a low-latency block cipher for pervasive computing applications. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 208–225. Springer (2012)

How to cite this article: Lu, J., et al.: Improved rotational-XOR cryptanalysis of Simon-like block ciphers. IET Inf. Secur. 16(4), 282–300 (2022). https://doi.org/10.1049/ise2.12061

APPENDICES

A ALZETTE-LIKE ROUND CONSTANTS

Alzette is a 64-bit ARX-based S-box designed by Beierle et al. [31] at CRYPTO 2020. It is also used as a crucial nonlinear component of Sparkle permutation family which submitted to the NIST lightweight cryptography standardisation process [32]. Alzette has eight variants, and the parameters are as follows:

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

In this study, we truncated these 32-bit constants into two 16-bits to fit our test parameters and cycle every 16 rounds (In fact, no more than 16 rounds of RX-characteristics are produced in the algorithm we test.), that is, the Alzette-like round constants used in this study are:

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

B PRINCE-LIKE ROUND CONSTANTS

Prince [33] is a block cipher that is optimised with respect to latency when implemented in hardware, which is introduced by Borghoff et al. at ASIACRYPT 2012. It is a 64-bit block cipher which is symmetric around the middle round. The special choice of round constants is as follows:

c_0 = 0000000000000000, c_1 = 13198a2e03707344,
c_2 = a4093822299f31d0, c_3 = 082efa98ec4e6c89,
c_4 = 452821e638d01377, c_5 = be5466cf34e90c6c,
c_6 = 7ef84f78fd955cb1, c_7 = 85840851f1ac43aa,
c_8 = c882d32f25323c54, c_9 = 64a51195e0e3610d,
c_{10} = d3b5a399ca0c2399, c_{11} = c0ac29b7c97c50dd.

For all $0 \le i \le 11$, $c_i \oplus c_{11-i}$ is the constant α = c0ac29b7c97c50dd, and that c_1, c_2, …, c_5 and α are derived from the fraction part of π = 3.14159….

In this study, we truncated c_1, c_2, …, c_8 constants into 8 × 4 16-bits to fit our test parameters, that is, the Prince-like round constants used in this study are:

c_0 = 7344, c_1 = 0370, c_2 = 8a2e, c_3 = 1319,
c_4 = 31d0, c_5 = 299f, c_6 = 3822, c_7 = a409,
c_8 = 6c89, c_9 = ec4e, c_{10} = fa98, c_{11} = 082e,
c_{12} = 1377, c_{13} = 38d0, c_{14} = 21e6, c_{15} = 4528,
c_{16} = 0c6c, c_{17} = 34e9, c_{18} = 66cf, c_{19} = be54,
c_{20} = 5cb1, c_{21} = fd95, c_{22} = 4f78, c_{23} = 7ef8,
c_{24} = 43aa, c_{25} = f1ac, c_{26} = 0851, c_{27} = 8584,
c_{28} = 3c54, c_{29} = 2532, c_{30} = d32f, c_{31} = c882.

C KEY RECOVERY ATTACK OF SIMECK WITH RX-DISTINGUISHERS

Consider a Simeck cipher with $2n$ block size and $4n$-bit key, assume that it has a round function $f_{5,0,1}$ and a key schedule that reuses the round function structure to produce the round keys. Assume that a d-round RX-distinguisher is available for Simeck$2n/4n$ with input difference $(\Delta_1 L^0, \Delta_1 R^0)$ and output difference $(\Delta_1 L^d, \Delta_1 R^d)$. Extending the distinguisher forward to attack $(d+2)$-round cipher, we denote the RX-differences and values according to the following figure (Figure C1). The input RX-difference to the ith round is denoted by $(\Delta_1 L^{i-1}, \Delta_1 R^{i-1})$ and the round key difference by $\Delta_1 k^{i-1}$; the corresponding intermediate values are $(L^{i-1}, R^{i-1}, k^{i-1})$ and $((L^{i-1})', (R^{i-1})', (k^{i-1})')$, the RX-differences between the values are computed by $((\overleftarrow{L^{i-1}} \oplus (L^{i-1})')$, $((\overleftarrow{R^{i-1}} \oplus (R^{i-1})')$ and $((\overleftarrow{k^{i-1}} \oplus (k^{i-1})')$.

Given the encryption of plaintext pairs with the input difference $(\Delta_1 L^0, \Delta_1 R^0)$ under a pair of related keys following the RX-characteristic in the key schedule, the encrypted values after the $(d+2)$-round are accessible to the attacker, that is, (L^{d+2}, R^{d+2}) and $((L^{d+2})', (R^{d+2})')$ are known.

To partially decrypt, the RX-difference $\Delta_1 R_j^d = R_{j-1}^d \oplus \left(R_j^d\right)'$ can be obtained by the ciphertexts with some key guesses.

$$
\begin{aligned}
\Delta_1 R_j^d &= R_{j-1}^d \oplus \left(R_j^d\right)' \\
&= \left(R_{j-1}^{d+1} \cdot R_{j-6}^{d+1}\right) \oplus R_{j-2}^{d+1} \oplus R_{j-1}^{d+2} \oplus k_{j-1}^d \\
&\quad \oplus \left(\left(R_j^{d+1}\right)' \cdot \left(R_{j-5}^{d+1}\right)'\right) \oplus \left(R_{j-1}^{d+1}\right)' \oplus \\
&\quad \left(R_j^{d+2}\right)' \oplus \left(k_j^d\right)' \\
&= \left(R_{j-1}^{d+1} \cdot R_{j-6}^{d+1}\right) \oplus R_{j-2}^{d+1} \oplus R_{j-1}^{d+2} \oplus k_{j-1}^d \\
&\quad \oplus \left(\left(\Delta_1 R_j^{d+1} \oplus R_{j-1}^{d+1}\right) \cdot \left(\Delta_1 R_{j-5}^{d+1} \oplus R_{j-6}^{d+1}\right)\right) \\
&\quad \oplus \left(\Delta_1 R_{j-1}^{d+1} \oplus R_{j-2}^{d+1}\right) \oplus \left(\Delta_1 R_j^{d+2} \oplus R_{j-1}^{d+2}\right) \oplus \\
&\quad \left(\Delta_1 k_j^d \oplus k_{j-1}^d\right) \\
&= \Delta_1 R_j^{d+1} \cdot \boldsymbol{R_{j-6}^{d+1}} \oplus \Delta_1 R_{j-5}^{d+1} \cdot \boldsymbol{R_{j-1}^{d+1}} \oplus \Delta_1 R_j^{d+1} \cdot \Delta_1 R_{j-5}^{d+1} \\
&\quad \oplus \Delta_1 R_{j-1}^{d+1} \oplus \Delta_1 R_j^{d+2} \oplus \Delta_1 k_j^d.
\end{aligned}
\tag{30}
$$

As Equation (30) shows, when the values of $\Delta_1 R_j^{d+1}$ and $\Delta_1 R_{j-5}^{d+1}$ are 1 or *, it is necessary to get the values of R_{j-6}^{d+1} and R_{j-1}^{d+1} for the computation of $\Delta_1 R_j^d$. In order to get the values of R_{j-6}^{d+1} and R_{j-1}^{d+1}, we should guess bit of k^{d+1} according to

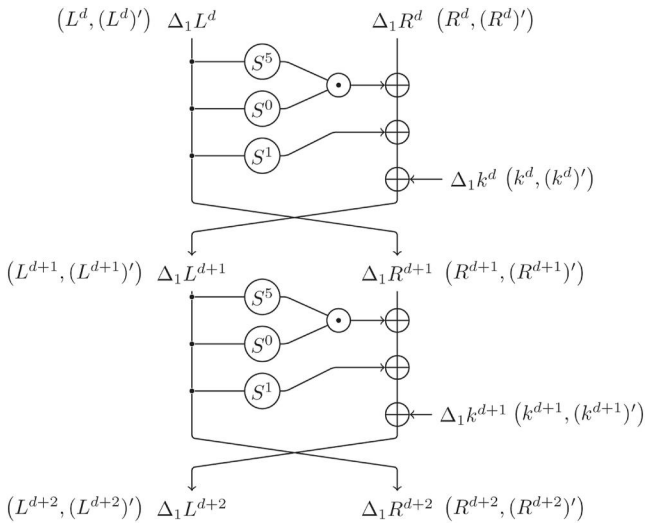

FIGURE C1 Notations in a 2-round key recovery of Simeck

Table C1. Besides, when the value of $\Delta_1 k_j^{d+1}$ is * in the RX-difference, it is necessary to guess the corresponding bits k_{j-1}^{d+1} and $\left(k_j^{d+1}\right)'$; when the value of $\Delta_1 k_j^d$ is * in the RX-difference, it is also necessary to guess the corresponding bits k_{j-1}^d and $\left(k_j^d\right)'$.

The two-round attack can be further extended to partially decrypt more rounds, and the key guessing conditions are similar.

D TESTING THE ATTACK PROCEDURE USING A 10-ROUND RX-DISTINGUISHER

To demonstrate the effectiveness of the attack, we also implement a key recovery attack on 12-round Simeck32 using a 10-round RX-distinguisher

$$D_d : (\texttt{0000}, \texttt{0004}) \xrightarrow{10} (\texttt{0011}, \texttt{0008})$$
$$D_k : (\texttt{0003}, \texttt{0001}, \texttt{0000}, \texttt{0004}) \xrightarrow{6} (\texttt{0001}, \texttt{0008}, \texttt{0001}, \texttt{0000})$$

The probabilities in the encryption part and the key schedule part are 2^{-10} and 2^{-10}, respectively.

The characteristic is extended by two rounds to obtain the difference pattern shown in Table D1. According to Appendix C, the required 12 key bits are $k^{11} = \{2, 4, 7, 8, 10, 14\}$, $(k^{11})' = \{3, 8\}$, $k^{10} = \{4, 15\}$ and $(k^{10})' = \{0, 5\}$ to verify the RX-difference at the beginning of round 10 (Table E1).

TABLE C1 Condition of guessed bits in k^{d+1}

$\Delta_1 R_j^{d+1}$	$\Delta_1 R_{j-5}^{d+1}$	R_{j-1}^{d+1}	R_{j-6}^{d+1}	Guessed bit of k^{d+1}
0	0	-	-	-
0	1/*	Need	-	$j-1$
1/*	0	-	Need	$j-6$
1/*	1/*	Need	Need	$j-1, j-6$

TABLE D1 Truncated RX-differential of bits obtained by extending the path of 10-round Simeck32 $(\texttt{0000}, \texttt{0004}) \xrightarrow{10} (\texttt{0011}, \texttt{0008})$ in the bottom directions

0	$\Delta_1 L^0$	0000 0000 0000 0000
	$\Delta_1 R^0$	0000 0000 0000 0100
	$\Delta_1 k^0$	0000 0000 0000 0100
10 rounds		
10	$\Delta_1 L^{10}$	0000 0000 0001 0001
	$\Delta_1 R^{10}$	0000 0000 0000 1000
	$\Delta_1 k^{10}$	0000 0000 00*0 011*
11	$\Delta_1 L^{11}$	0000 00*0 00** 110*
	$\Delta_1 R^{11}$	0000 0000 0001 0001
	$\Delta_1 k^{11}$	0000 000* 0001 *100
12	$\Delta_1 L^{12}$	0*00 0*** **** ****
	$\Delta_1 R^{12}$	0000 00*0 00** 110*

Using 2^{15} plaintext pairs with input RX-difference (0000,0004), we collect the encrypted pairs after 12 rounds under a pair of keys

$$K = (\texttt{F54A}, \texttt{99C7}, \texttt{2B88}, \texttt{EAB8})$$
$$K' = (\texttt{EA96}, \texttt{338E}, \texttt{5710}, \texttt{D575})$$

which satisfies the RX-distinguisher D_k. By guessing the 12 key bits and setting arbitrary values in the other key bits of k^{11} and k^{10}, we decrypt the last two rounds and count the number of right pairs for the output difference (0011,0008). In 16 experiments, we observe that the number of right pairs is 32.1875, which matches the attack hypothesis: $2^{15} \times 2^{-10} = 2^5$.

E REPORTED RX-CHARACTERISTICS FOR Simeck32/48/64

(Tables E2 and E3)

TABLE E1 A 20-round RX-characteristic for Simeck32/64 and a 27-round RX-characteristic for Simeck48/96

Round	Simeck32/64		Simeck48/96	
	Key RX-difference	Data RX-difference	Key RX-difference	Data RX-difference
0	0004	(0000‖0004)	000004	(000000‖000004)
1	0000	(0000‖0000)	000000	(000000‖000000)
2	0001	(0000‖0000)	000001	(000000‖000000)
3	0002	(0001‖0000)	000002	(000001‖000000)
4	0002	(0000‖0001)	000002	(000000‖000001)
5	0005	(0003‖0000)	000005	(000003‖000000)
6	0001	(0000‖0003)	000001	(000000‖ 000003)
7	0002	(0002‖0000)	000002	(000002‖000000)
8	000a	(0004‖0002)	00000a	(000004‖000002)
9	0002	(0000‖0004)	000002	(000000‖000004)
10	0000	(0006‖0000)	000000	(000006‖000000)
11	0013	(000a‖0006)	000013	(00000a‖000006)
12	000a	(0001‖000a)	00000a	(000001‖00000a)
13	0004	(0002‖0001)	000004	(000002‖000001)
14	0000	(0001‖0002)	000000	(000001‖000002)
15	0001	(0000‖0001)	000003	(000001‖000001)
16	0000	(0000‖0000)	000000	(000000‖000001)
17	0002	(0000‖0000)	000002	(000001‖000000)
18	0006	(0002‖0000)	000002	(000000‖000001)
19	0007	(0000‖0002)	000005	(000003‖000000)
20		(0005‖0000)	000001	(000000‖000003)
21			000000	(000002‖000000)
22			000008	(000006‖000002)
23			000002	(000000‖000006)
24			000004	(000004‖000000)
25			00011d	(000008‖000004)
26			000048	(000001‖000008)
27				(000062‖000001)
Prob.	2^{-34}	2^{-26}	2^{-50}	2^{-44}

TABLE E2 A 34-round RX-characteristic for Simeck64/128

Round	Key RX-difference	Data RX-difference
0	00000004	(00000000‖00000004)
1	00000000	(00000000‖00000000)
2	00000001	(00000000‖00000000)
3	00000002	(00000001‖00000000)
4	00000002	(00000000‖00000001)
5	00000004	(00000003‖00000000)
6	00000001	(00000000‖00000003)
7	00000002	(00000002‖00000000)
8	0000000c	(00000004‖00000002)
9	00000001	(00000002‖00000004)
10	00000000	(00000001‖00000002)
11	00000013	(00000001‖00000001)
12	0000000b	(00000011‖00000001)
13	00000004	(00000009‖00000011)
14	00000001	(00000006‖00000009)
15	00000000	(00000002‖00000006)
16	00000002	(00000000‖00000002)
17	00000002	(00000000‖00000000)
18	00000004	(00000002‖00000000)
19	00000001	(00000000‖00000002)
20	00000002	(00000003‖00000000)
21	00000008	(00000005‖00000003)
22	00000003	(00000000‖00000005)
23	00000000	(00000006‖00000000)
24	0000001f	(00000008‖00000006)
25	00000008	(00000001‖00000008)
26	00000006	(00000002‖00000001)
27	00000000	(00000001‖00000002)
28	00000002	(00000001‖00000001)
29	00000001	(00000000‖00000001)
30	00000000	(00000000‖00000000)
31	00000002	(00000000‖00000000)
32	00000006	(00000002‖00000000)
33	00000007	(00000000‖00000002)
34		(00000005‖00000000)
Prob.	2^{-70}	2^{-56}

TABLE E3 The 23-round RX-characteristic used for the key recovery attack on the 28-round Simeck64/128

Round	Key RX-difference	Data RX-difference
0	00000117	(00000000‖00000113)
1	00000008	(00000004‖00000000)
2	00000004	(00000000‖00000004)
3	00000000	(00000000‖00000000)
4	00000001	(00000000‖00000000)
5	00000002	(00000001‖00000000)
6	00000002	(00000000‖00000001)
7	00000005	(00000003‖00000000)
8	00000001	(00000000‖00000003)
9	00000002	(00000002‖00000000)
10	0000000c	(00000004‖00000002)
11	00000003	(00000002‖00000004)
12	00000000	(00000001‖00000002)
13	00000013	(00000001‖00000001)
14	0000000c	(00000011‖00000001)
15	00000005	(0000000e‖00000011)
16	00000000	(00000004‖0000000e)
17	00000002	(00000002‖00000004)
18	00000002	(00000000‖00000002)
19	00000000	(00000000‖00000000)
20	00000000	(00000000‖00000000)
21	00000002	(00000000‖00000000)
22	00000007	(00000002‖00000000)
23		(00000001‖00000002)
Prob.	2^{-44}	2^{-34}