Hindawi
Security and Communication Networks
Volume 2020, Article ID 5968584, 12 pages
https://doi.org/10.1155/2020/5968584

Research Article

Rotational-XOR Rectangle Cryptanalysis on Round-Reduced Simon

Bonwook Koo, Younghoon Jung, and Woo-Hwan Kim

The Affiliated Institute of ETRI, Daejeon, Republic of Korea

Correspondence should be addressed to Bonwook Koo; kidkoo@gmail.com

Received 13 November 2018; Revised 24 March 2020; Accepted 12 June 2020; Published 22 July 2020

Academic Editor: Kuo-Hui Yeh

Recently, Ashur and Liu introduced the *Rotational-XOR-difference* approach which is a modification of rotational cryptanalysis, for an ARX cipher Speck (Ashur and Liu, 2016). In this paper, we apply the Rotational-XOR-difference (RXD) approach to a non-ARX cipher Simon and evaluate its security. First, we studied how to calculate the probability of an RXD for bitwise AND operation that the round function of Simon is based on unlike Speck is on modular addition. Next, we prove that two RXD trails can be connected such that it becomes possible to construct a boomerang/rectangle distinguisher similar to the case using differential characteristics. Finally, we construct related-key rectangle distinguishers for round-reduced versions of Simon with block lengths of 32, 48, and 64, and we suggest a five- or six-round key recovery attack. To our knowledge, it is the first attempt to apply the notion of rotational cryptanalysis for a non-ARX cipher. Although our attack does not show the best results for Simon thus far, the attempt here to define and apply a new cryptanalytic characteristic is meaningful, and we expect further improvements and applications to other ciphers to be made in subsequent studies.

1. Introduction

In a cryptosystem for confidentiality, the block cipher is a necessary building block for core functionality. So, because the security of block ciphers affects the applicability of the algorithm and the usability of the cryptosystem which uses the cipher as well, the security of a block cipher should be evaluated comprehensively and precisely. Over the last decade, many researchers have studied various techniques by which to design outstanding lightweight ciphers. One notable result of such research stream is design paradigm is omitting S-box, such as ARX. ARX is a design methodology for secret key primitives which uses only modular Addition, Rotation, and eXclusive OR operations. A number of outperforming lightweight block ciphers, such as Threefish [1], Chaskey Cipher [2], HIGHT [3], Speck [4], LEA [5], and Sparx [6] are designed in this framework. Another design strategy is to use the bitwise AND operation for nonlinear part of an algorithm. Although this approach is somewhat less popular than ARX, outstanding hardware-oriented ciphers such as KATAN/KTANTAN [7], Simon [4], and Simeck [8] utilize this strategy.

Rotational cryptanalysis was initially proposed to attack the block cipher Threefish, which is an internal permutation of the hash function Skein [9]. It was combined with the notion of a rebound attack considering the results of the best attack against Skein. Subsequently, the rotational probability was recalculated [10] considering the failure of the Markov assumption of chained modular additions, and a new calculating algorithm was applied to correct the results on BLAKE2 and to provide valid results on simplified Skein. Nevertheless, it appears to be difficult to apply rotational cryptanalysis to ciphers in which constant XOR is used for the enciphering procedure. This problem has remained unsolved until the following result is presented.

Recently, Ashur and Liu proposed a new type of rotational cryptanalysis that can overcome the aforementioned disadvantage by injecting constants into states [11]. This new approach can be used to evaluate the security of ciphers with constant XOR in their encryption scheme. Therefore, they applied it to the block cipher Speck-32/64 and successfully constructed a seven-round distinguisher. To do this, they introduced the notion of the Rotational-XOR (RX) pair $(x \oplus a_1, (x \lll \gamma) \oplus a_2)$ and the associated rotational-XOR-

differences (RXD) $((a_1, a_2), \gamma)$, where x is a random variable and a_1 and a_2 are constants. In particular, they presented a closed formula for calculating the RX probability occurred upon a modular addition.

In the present paper, we attempt to apply Ashur's constant injecting approach to a non-ARX cipher Simon which is based on the bitwise AND operation. While Ashur and Liu demonstrated how to calculate the RX probability and how to propagate an RX pair through the modular addition, we present a closed formula for calculating the probability and propagation rule of an RX pair through a bitwise AND operation. We also find that the propagation of the RX pair due to the operations used in Simon is similar to those of the ordinary differential characteristics and we show that the probability of boomerang/rectangle characteristics using RXD can be calculated similarly to the boomerang/rectangle characteristic using the ordinary differential characteristics. Therefore, we can construct boomerang/rectangle characteristics using two RXD trails. We refer to this cryptanalysis with such characteristics as *Rotational-XOR boomerang (or rectangle) cryptanalysis*. Our attack works in the related-key model in which the attacker uses ciphertexts encrypted with different but related keys because rotational cryptanalysis is naturally a related-key attack.

Based on our results, we evaluate the security of several instances of Simon in the related-key model. Because our approach is more effective on ciphers with smaller block sizes, we apply it to Simon with a block length of less than or equal to 64. As a result, for some parameters, we could obtain results very close to the best results on Simon thus far. Table 1 shows the results of our attacks compared to the results of other attacks.

Although our results are not the best records for Simon, our approach can be adopted to analyze other existing or future ciphers based on the bitwise AND. Examples include Simeck and KATAN/KTANTAN.

The rest of this paper is organized as follows: in Section 2, we define some of the notations used here and give brief introductions of rotational cryptanalysis, the rotational-XOR-difference, and boomerang/rectangle cryptanalysis. The RX probability and RX characteristics of Simon are described in Section 3. In Section 4, we present the RX rectangle attack on Simon, including the key recovery phase, and calculate the computational and data complexities of the attacks. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Notations. In this paper, we use the following notations:

(i) $w_H(x)$: Hamming weight of bit string x

(ii) $x \boxplus y$: modular addition of bit strings x and y

(iii) $x \vee y$: bitwise OR of bit strings x and y

(iv) $x \wedge y$: bitwise AND of bit strings x and y

(v) $x \ll r$: r bit left shift of a bit string x

(vi) $x^{<r}$, $x \lll r$: r bit left rotation (cyclic shift) of a bit string x

(vii) $x^{>r}$: r bit right rotation (cyclic shift) of a bit string x

(viii) $\overrightarrow{x}$: left rotation (cyclic shift) of a bit string x by a predefined γ, usually $\gamma = 1$

(ix) $\overleftarrow{x}$: right rotation (cyclic shift) of a bit string x by a predefined γ, usually $\gamma = 1$

(x) x^i: i-th bit of a bit string x

(xi) $x \prec y$ means that every bit in y is larger or equal to the corresponding bit in x

2.2. Rotational Cryptanalysis. Since Khovratovich et al. introduced rotational cryptanalysis in 2010 [9], it has been used to evaluate symmetric key cryptographic primitives based on the ARX design framework [10, 18, 19]. Rotational cryptanalysis appears to be suitable for ARX ciphers because the rotational pair is preserved through rotations and XORes between variables and transformed by modular additions with high probability levels, unlike ciphers based on S-boxes.

Rotational cryptanalysis exploits the nonrandom behavior of ciphertext pairs generated from the rotational plaintext pairs $((p_0, p_1, \ldots, p_{m-1}), (\overrightarrow{p_0}, \overrightarrow{p_1}, \ldots, \overrightarrow{p_{m-1}}))$ where $\overrightarrow{p_i} = p_i \lll \gamma$ for some integer γ (γ is typically selected to 1 for a higher probability). The probability that modular addition of two rotational pairs $(x, \overrightarrow{x})$ and $(y, \overrightarrow{y})$ is also a rotational pair is given by

$$P[\overrightarrow{x \boxplus y} = \overrightarrow{x} \boxplus \overrightarrow{y}] = \frac{1}{4}\left(1 + 2^{\gamma - n} + 2^{-\gamma} + 2^{-n}\right), \quad (1)$$

where n is the bit length of both x and y [20]. For a large n, that probability goes to $2^{-1.415}$ when $\gamma = 1$ and to 2^{-2} when $\gamma = (n/2)$.

However, XOR or modular addition with a constant destroys the rotational relationship of a pair when the constant cannot transform into itself by γ-bit rotation. So, the rotational cryptanalysis cannot be widely adopted in relation to the block cipher analysis.

2.3. Rotational-XOR-Difference. In 2016, Ashur and Liu introduced modified rotational cryptanalysis using the rotational-XOR-difference (RXD) to overcome the limitations caused by the constants and applied it to block cipher Speck [11]. They defined an RX pair as $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ and its RXD as $((a_1, a_2), \gamma)$. It is obvious that the RX pair is preserved even if some constant is XORed to the values of the pair. In addition, they proved the following Theorem 1, which shows us how to calculate the transition probability of RX pair through modular addition. We assume that $\gamma = 1$ throughout this paper; hence, we let $\overrightarrow{x}$ denote $x \lll 1$.

Theorem 1 (Theorem 1 in [11]). *Let $x, y \in \mathbb{F}_{2^n}$ represent independent uniform random variables. Let $a_1, a_2, b_1, b_2, c_1,$*

Table 1: Comparison of attack results on Simon.

Block/key	Attacks	Rounds	Data	Computation	Reference
32/64	Differential	22/32	2^{32}	$2^{57.9}$	[12]
	Linear hull	23/32	$2^{31.19}$	$2^{61.84}$	[13]
	Impossible	20/32	2^{32}	$2^{45.5}$	[14]
	Zero correlation	21/32	2^{32}	$2^{59.4}$	[15]
	Correlated sequence	27/32	3	$2^{62.94}$	[16]
	Related-key linear	23/32	$2^{46.3}$	$2^{46.65}$	[17]
	Related-key RX rectangle	22/32	$2^{30.5}$	$2^{60.4}$	This paper
48/72	Linear hull	24/36	$2^{47.92}$	$2^{67.89}$	[13]
	Zero correlation	21/36	2^{48}	$2^{61.9}$	[15]
	Related-key RX rectangle	21/36	2^{47}	$2^{69.1}$	This paper
48/96	Differential	24/36	2^{48}	78.99	[12]
	Linear hull	25/36	$2^{47.92}$	$2^{89.89}$	[13]
	Zero correlation	22/36	2^{48}	$2^{80.5}$	[15]
	Related-key linear	28/36	$2^{70.9}$	$2^{71.07}$	[17]
	Related-key RX rectangle	24/36	$2^{46.5}$	$2^{92.3}$	This paper
64/96	Differential	29/42	2^{63}	86.94	[12]
	Linear hull	30/42	$2^{63.53}$	$2^{93.62}$	[13]
	Zero correlation	23/42	2^{64}	$2^{90.4}$	[15]
	Related-key RX rectangle	22/42	2^{62}	$2^{91.8}$	This paper
64/128	Differential	30/44	2^{63}	110.99	[12]
	Linear hull	31/44	$2^{63.53}$	2^{120}	[13]
	Zero correlation	24/44	2^{64}	$2^{116.8}$	[15]
	Related-key linear	34/44	$2^{95.32}$	$2^{95.5}$	[17]
	Related-key RX rectangle	25/44	$2^{61.5}$	$2^{123.0}$	This paper

and c_2 be constants in $\mathbb{F}_{2^n}$ and δ_x, δ_y, and δ_z be the $n-1$ most significant bits of $\overrightarrow{a_1} \oplus a_2$, $\overrightarrow{b_1} \oplus b_2$, and $\overrightarrow{c_1} \oplus c_2$, respectively. Then,

$$P\left[\overrightarrow{(x \oplus a_1) \boxplus (y \oplus b_1)} \oplus \overrightarrow{c_1} = (\overrightarrow{x} \oplus a_2) \boxplus (\overrightarrow{y} \oplus b_2) \oplus c_2\right] = \begin{cases} 2^{-w_H(T)} \cdot 2^{-3}, & \text{if } S \oplus 1 \prec T, \\ 2^{-w_H(T)} \cdot 2^{-1.415}, & \text{else if } S \prec T, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $T = ((\delta_x \oplus \delta_z) \vee (\delta_y \oplus \delta_z)) \ll 1$ and $S = ((\delta_x \oplus \delta_y \oplus \delta_z) \oplus ((\delta_x \oplus \delta_y \oplus \delta_z) \ll 1))$.

It is clear that the rotation of an RX pair is an RX pair and that the XOR of two RX pairs is also an RX pair.

2.4. Boomerang/Rectangle Characteristics. A boomerang attack [21] uses two differential characteristics $\Delta \longrightarrow \Delta^*$ for $E0$ and $\nabla \longrightarrow \nabla^*$ for $E1$, whose probabilities are p and q, respectively, where the target block cipher E is a composition of subciphers $E0$ and $E1$, i.e., $E = E1 \circ E0$. If two plaintexts P and P' such that $P \oplus P' = \Delta$ satisfy

$$E0(P) \oplus E0(P') = \Delta^*, \tag{3}$$

with probability p and both

$$\begin{aligned} E1^{-1}(E(P)) \oplus E1^{-1}(E(P) \oplus \nabla) &= \nabla^*, \\ E1^{-1}(E(P')) \oplus E1^{-1}(E(P') \oplus \nabla) &= \nabla^*, \end{aligned} \tag{4}$$

are satisfied with probability q^2, then, clearly

$$E1^{-1}(E(P) \oplus \nabla) \oplus E1^{-1}(E(P') \oplus \nabla) = \Delta^*. \tag{5}$$

Hence,

$$E^{-1}(E(P) \oplus \nabla) \oplus E^{-1}(E(P') \oplus \nabla) = \Delta, \tag{6}$$

with probability p.

Therefore, if we denote $E^{-1}(E(P) \oplus \nabla)$ and $E^{-1}(E(P') \oplus \nabla)$ as Q and Q', we can distinguish E from the random permutation according to the distribution of $Q \oplus Q'$, where $P \oplus P' = \Delta$ and $p^2 q^2 > 2^{-n}$.

A boomerang attack is an adaptive chosen-ciphertext attack that can be transformed into a known-plaintext attack based on the following rectangle distinguisher [22].

Suppose that we have two pairs of plaintext (P, P') and (Q, Q') such that

$$P \oplus P' = Q \oplus Q' = \Delta. \tag{7}$$

In such a case, we have

$$E0(P) \oplus E0(P') = E0(Q) \oplus E0(Q') = \Delta^*, \tag{8}$$

with probability p^2. Here, if we suppose that $E0(P) \oplus E0(Q) = \nabla^*$ with probability 2^{-n}, then we have

$$E0(P') \oplus E0(Q') = \nabla^*, \tag{9}$$

accordingly

$$E(P) \oplus E(Q) = E(P') \oplus E(Q') = \nabla, \tag{10}$$

with probability q^2.

Thus, we can distinguish E from the random permutation using the distributions of $E(P) \oplus E(Q)$ and $E(P') \oplus E(Q')$, if $P \oplus P' = Q \oplus Q' = \Delta$ and $p^2 q^2 2^{-n} > 2^{-2n}$.

2.5. Description of Simon. Simon [4] is a family of block ciphers which support various bit lengths of blocks and keys. For $w = 16$, 24, 32, 48, and 64, Simon-n/k has a block size of $n = 2w$ and a key size of $k = 2w$, $3w$, or $4w$. Encryption of Simon involves iterations of the round transformations

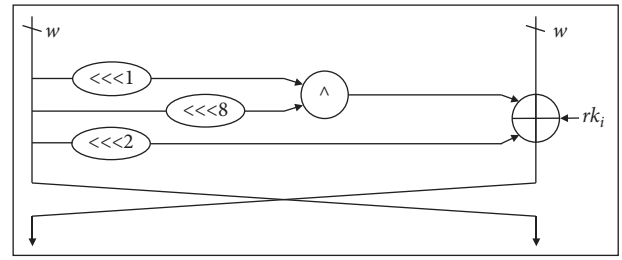

Figure 1: Round transformation of Simon.

shown in Figure 1, where $\wedge$ and $\oplus$ are bitwise AND and XOR, respectively. rk_i for $i = 1, 2, \ldots$ denotes the i-th round keys generated by one of the three key schedules shown in Figure 2 depending on the number of keywords, where c is equal to $2^w - 4$ and $(z_j)_i$ is the i-th bit of z_j, defined as follows.

$$\begin{aligned}
z_0 &= 11111010001001010110000111001101111101000100101011000011100110\ldots,\\
z_1 &= 10001110111110010011000010110101000111011111001001100001011010\ldots,\\
z_2 &= 10101111011100000011010010011000101000010001111110010110110011\ldots,\\
z_3 &= 11011011101011000110010111100000010010001010011100110100001111\ldots,\\
z_4 &= 11010001111001101011011000100000010111000011001010010011101111\ldots.
\end{aligned} \tag{11}$$

More specific descriptions for each instance of Simon can be found in the literature [4].

3. Rotational-XOR-Differences for Simon

Unlike Speck, based on modular addition, Simon uses the bitwise AND for its round function, though this operation does not always preserve RX pairs. Consequently, here it is necessary to calculate the probability that two RX pairs are transformed into another RX pair through the bitwise AND operation.

3.1. Calculating the Probabilities of Rotational-XOR Pairs for the Bitwise AND Operation. Suppose $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ and $(y \oplus b_1, \overrightarrow{y} \oplus b_2)$ are two input RX pairs of a bitwise AND operation. In such a case, the output pair is $((x \oplus a_1) \wedge (y \oplus b_1), (\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2))$. Let $(x \oplus a_1) \wedge (y \oplus b_1) = z \oplus c_1$ and $(\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2) = \tilde{z} \oplus c_2$ for some constants c_1 and c_2. The probability that the output pair becomes an RX pair then becomes

$$P[\overrightarrow{z} = \tilde{z}] = P\left[\overrightarrow{(x \oplus a_1) \wedge (y \oplus b_1) \oplus c_1} = (\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2) \oplus c_2\right]. \tag{12}$$

We can observe when the probability is nonzero and how to calculate the probability by Theorem 2, under the assumption that two inputs of the bitwise AND are independent uniformly random variables.

Theorem 2 (bitwise AND of two random variables). *Let $x, y \in \mathbb{F}_2^w$ represent independent uniformly random variables for some positive integer w, and let a_1, a_2, b_1, b_2, c_1, and c_2 be constants in $\mathbb{F}_2^w$ and $\delta_x = \overrightarrow{a_1} \oplus a_2$, $\delta_y = \overrightarrow{b_1} \oplus b_2$, and $\delta_z = \overrightarrow{c_1} \oplus c_2$. Then,*

$$P\left[\overrightarrow{(x \oplus a_1) \wedge (y \oplus b_1) \oplus c_1} = (\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2) \oplus c_2\right] = \begin{cases} 2^{-w_H(\delta_x \vee \delta_y)}, & \text{if } (\delta_x \vee \delta_y) \wedge \delta_z, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

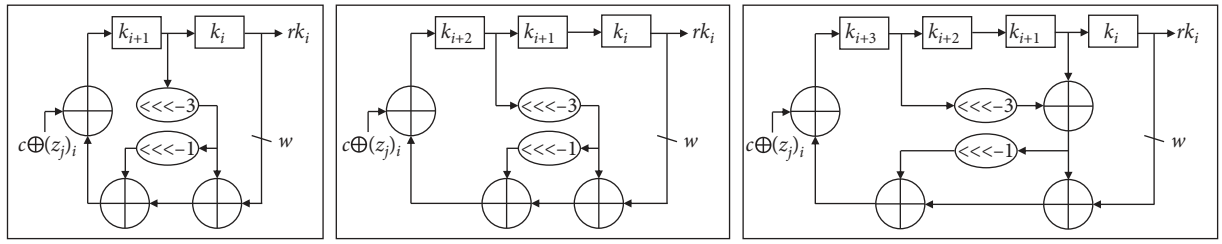

Figure 2: Three key schedules of Simon.

Proof. Let $(x \oplus a_1) \wedge (y \oplus b_1) = z \oplus c_1$ and $(\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2) = \tilde{z} \oplus c_2$. In this case, we will calculate the probability that $\tilde{z} = \overrightarrow{z}$.

Because $\oplus$ and $\wedge$ are bitwise operations, it is clear that

$$\begin{aligned}\overrightarrow{z} &= \overrightarrow{((x \oplus a_1) \wedge (y \oplus b_1)) \oplus c_1} = \left((\overrightarrow{x} \oplus \overrightarrow{a_1}) \wedge \left(\overrightarrow{y} \oplus \overrightarrow{b_1}\right)\right) \oplus \overrightarrow{c_1} \\ &= (\overrightarrow{x} \wedge \overrightarrow{y}) \oplus \left(\overrightarrow{x} \wedge \overrightarrow{b_1}\right) \oplus (\overrightarrow{a_1} \wedge \overrightarrow{y}) \oplus \left(\overrightarrow{a_1} \wedge \overrightarrow{b_1}\right) \oplus \overrightarrow{c_1}, \\ \tilde{z} &= ((\overrightarrow{x} \oplus a_2) \wedge (\overrightarrow{y} \oplus b_2)) \oplus c_2 = (\overrightarrow{x} \wedge \overrightarrow{y}) \oplus (\overrightarrow{x} \wedge b_2) \oplus (a_2 \wedge \overrightarrow{y}) \oplus (a_2 \wedge b_2) \oplus c_2.\end{aligned} \tag{14}$$

Therefore, now we calculate the probability that

$$\begin{aligned}\tilde{z} = \overrightarrow{z} &\Longrightarrow \left(\overrightarrow{x} \wedge \overrightarrow{b_1}\right) \oplus (\overrightarrow{a_1} \wedge \overrightarrow{y}) \oplus \left(\overrightarrow{a_1} \wedge \overrightarrow{b_1}\right) \oplus (\overrightarrow{x} \wedge b_2) \oplus (a_2 \wedge \overrightarrow{y}) \oplus (a_2 \wedge b_2) \\ &= \left(\overrightarrow{x} \wedge \left(\overrightarrow{b_1} \oplus b_2\right)\right) \oplus (\overrightarrow{y} \wedge (\overrightarrow{a_1} \oplus a_2)) \oplus \left(\overrightarrow{a_1} \wedge \overrightarrow{b_1}\right) \oplus (a_2 \wedge b_2) = \overrightarrow{c_1} \oplus c_2.\end{aligned} \tag{15}$$

According to the definitions of δ_x, δ_y, and δ_z, we have the following equations:

$$\begin{aligned}&\left(\overrightarrow{x} \wedge \delta_y\right) \oplus (\overrightarrow{y} \wedge \delta_x) \oplus \left(\overrightarrow{a_1} \wedge \overrightarrow{b_1}\right) \oplus (a_2 \wedge b_2) = \delta_z \\ &\Longrightarrow \left(\overrightarrow{x} \wedge \delta_y\right) \oplus (\overrightarrow{y} \wedge \delta_x) \oplus 0\left(\delta_x \oplus a_2\right) \wedge \left(\delta_y \oplus b_2\right) \oplus (a_2 \wedge b_2) = \delta_z \\ &\Longrightarrow \left(\overrightarrow{x} \wedge \delta_y\right) \oplus (\overrightarrow{y} \wedge \delta_x) \oplus \left(\delta_x \wedge \delta_y\right) \oplus (\delta_x \wedge b_2) \oplus \left(\delta_y \wedge a_2\right) \delta_z.\end{aligned} \tag{16}$$

At this point, we consider equation (16) in bit by bit. For each $i \in \{0, 1, \ldots, w-1\}$, if the i-th bits of δ_x and δ_y are 0, i.e., $\delta_x^i = \delta_y^i = 0$, then the i-th bit of the left-hand side of equation (16) is 0; hence, the stipulations of the i-th bit of equation (16) are met only if $\delta_3^i = 0$ with a probability of 1. Otherwise, if $\delta_x^i = 0$ and $\delta_y^i = 1$, in this case, equation (16) implies $\overrightarrow{x}^i \oplus a_2^i = \delta_z^i$ which is satisfied depending on $\overrightarrow{x}^i$. Because we assume x be a uniform random variable, the probability that the requirements associated with the i-th bit of equation (16) are satisfied is 1/2. Similarly, if $\delta_x^i = 1$ and $\delta_y^i = 0$, the conditions of the i-th bit of equation (16) are met with a probability of 1/2 depending on $\overrightarrow{y}^i$ regardless of the value of δ_z^i. The last case is one in which $\delta_x^i = 1$ and $\delta_y^i = 1$. In this case, (16) implies that $\overrightarrow{x}^i \oplus \overrightarrow{y}^i \oplus 1 \oplus a_2^i \oplus b_2^i = \delta_z^i$ and the conditions of this equation are also satisfied with a probability of 1/2 regardless of the value of δ_z^i because a_2^i and b_2^i are fixed values.

Thus, for some fixed δ_x, δ_y, and δ_z, if there exists i such that $\delta_x^i = \delta_y^i = 0$ and $\delta_z^i = 1$, the probability is then 0. Therefore, the probability is nonzero only if $(\delta_x \mid \delta_y) \wedge \delta_z = \delta_z$. And for each i such that $\delta_x^i \vee \delta_y^i = 1$, the conditions of the i-th bit of (16) are met with a probability of 1/2. Therefore, the probability that the conditions of (16) are met (which we want to calculate) is $2^{-w_H(\delta_x \vee \delta_y)}$.

However, as shown in Figure 1, the two inputs x and y of the bitwise AND operation in Simon are highly dependent on each other. Therefore, we need to calculate the probability more precisely. The following Theorem 3 is analogous to Theorem 3 for covering the Simon case and the case of $j = 7$ is relevant to Simon-$2w/k$. □

Theorem 3. *(bitwise AND of two values from one random variable). Let $x \in \mathbb{F}_2^w$ be an uniformly random variable for a positive integer w and $j \le w$ be a positive integer that does not divide w. Additionally, let a_1, a_2, c_1, and c_2 be constants in $\mathbb{F}_2^w$ and $\delta_x = \overrightarrow{a_1} \oplus a_2$ and $\delta_z = \overrightarrow{c_1} \oplus c_2$. In this case,*

$$P\left[\overrightarrow{(x \oplus a_1) \wedge \left(x^{<j} \oplus a_1^{<j}\right)} \oplus c_1 = (\overrightarrow{x} \oplus a_2) \wedge \left(\overrightarrow{x^{<j}} \oplus a_2^{<j}\right) \oplus c_2\right] = \begin{cases} 2^{-n+1}, & \text{if } \delta_x = 2^n - 1, \\ 2^{-w_H\left(\delta_x \vee \delta_x^{<j}\right)+w_H(T)}, & \text{else if} \left(\delta_x \vee \left(\delta_x^{<j}\right)\right) \wedge \delta_z = \delta_z, \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where $T = \overline{(\delta_x^{<j})} \wedge \delta_x \wedge (\delta_x^{<2j})$.

Proof. Similar to the proof of Theorem 3, we now calculate the probability that the following equation holds:

$$\begin{aligned} &\left(\overrightarrow{x} \wedge \delta_x^{<j}\right) \oplus \left(\overrightarrow{x^{<j}} \wedge \delta_x\right) \oplus \left(\delta_x \wedge \delta_x^{<j}\right) \oplus \left(\delta_x \wedge a_2^{<j}\right) \oplus \left(\delta_x^{<j} \wedge a_2\right) = \delta_z, \\ &\Rightarrow \left(\delta_x \wedge \delta_x^{<j}\right) \oplus \left(\delta_x^{<j} \wedge (\overrightarrow{x} \oplus a_2)\right) \oplus \left(\delta_x \wedge (\overrightarrow{x} \oplus a_2)^{<j}\right) = \delta_z. \end{aligned} \tag{18}$$

Here, we consider equation (18) in bit by bit.

For each $i \in \{0, 1, \ldots, w-1\}$, if $(\delta_x)^i = (\delta_x^{<j})^i = 0$, then

$$(\delta_z)^i = 0, \tag{19}$$

with a probability of 1.

Else if $(\delta_x)^i = 0$ and $(\delta_x^{<j})^i = 1$, according to equation (18),

$$(\delta_z)^i = (\overrightarrow{x} \oplus a_2)^i. \tag{20}$$

However, because $(\overrightarrow{x})^i$ would appear again when we define $(\delta_z^{>j})^i$, it is necessary to consider the subcases along with the value of $(\delta_x^{>j})^i$. If $(\delta_x^{>j})^i = 0$, $(\overrightarrow{x})^i$ does not contribute to the definition of $(\delta_z^{>j})^i$. Therefore, $(\overrightarrow{x})^i$ can be regarded as a free random variable (which means it is not used to define other bits of δ_z); therefore, $(\delta_z)^i$ can be 0 or 1 with a probability of 1/2. Otherwise (i.e., $(\delta_x^{>j})^i = 1$), as $(\delta_x)^i = 0$, $(\delta_z^{>j})^i = (\overrightarrow{x} \oplus a_2)^i$ and we have the relationship of $\delta_z^i = (\delta_z^{>j})^i$.

Otherwise, if $(\delta_x)^i = 1$ and $(\delta_x^{<j})^i = 0$, similar to the above case, $(\delta_z)^i$ is defined as a free random variable $(\overrightarrow{x}^{<j})^i$ when $(\delta_x^{<2j})^i = 0$. On the other hand, $(\delta_z^{<j})^i = ((\overrightarrow{x} \oplus a_2)^{<j})^i$ and then we have the relationship of $(\delta_z)^i = (\delta_z^{<j})^i$.

Otherwise, $(\delta_x)^i = 1$ and $(\delta_x^{<j})^i = 1$, according to equation (18),

$$(\delta_z)^i = 1 \oplus \left((\overrightarrow{x} \oplus a_2) \oplus (\overrightarrow{x} \oplus a_2)^{<j}\right)^i. \tag{21}$$

It is necessary to check for subcases for $(\delta_x^{<2j})^i$ and $(\delta_x^{>j})^i$. We already know that $(\delta_x^{<j})^i = 1$ and $(\delta_x)^i = 1$. If $(\delta_x^{<2j})^i \wedge (\delta_x^{>j})^i = 0$, $(\delta_z)^i$ is defined as a free random variable, $(\overrightarrow{x})^i$, $(\overrightarrow{x}^{<j})^i$, or both conditions apply. Hence, $(\delta_z)^i$ is 0 or 1 with a probability of 1/2. Otherwise, $(\delta_x^{<2j})^i = (\delta_x^{>j})^i = 1$; then, $(\delta_z^{>j})^i$ is defined as $(\overrightarrow{x}^{>2j})^i$ and $(\overrightarrow{x}^{>j})^i$, and $(\delta_z^{<j})^i$ is defined according to $(\overrightarrow{x})^i$ and $(\overrightarrow{x}^{<j})^i$. This means that three bits of δ_z are defined as four independent bits of the random variable $\overrightarrow{x}$. Such chain ends with the bit of δ_z, which is independently defined except when $\delta_x = 2^n - 1$ because $j \nmid n$. Thus, every bit in the chain, including $(\delta_z)^i$, has a value of 0 or 1 with a probability of 1/2. If $\delta_x = 2^n - 1$, every single bit of δ_z is defined by two bits of $\overrightarrow{x}$ and they are related to each other. Hence, the probability that δ_z has some value is 2^{-n+1}. Consequently, if $(\delta_x^{>j})^i = 0$, $(\delta_x)^i = 1$, $(\delta_x^{>2j})^i = 1$ for some i, then the freedom of $(\delta_z)^i$ and $(\delta_z^{>j})^i$ is decreased by 1 bit and there are exactly $w_H(\overline{(\delta_x^{<j})} \wedge \delta_x \wedge (\delta_x^{<2j}))$ pairs of bits in δ_x. □

3.2. Searching for the Rotational-XOR-Differences Trail of Simon

3.2.1. How to Define the Rotational-XOR-Differences Trail. Because we let $\gamma = 1$, the RXD of an RX pair $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ can be denoted as $((a_1, a_2), 1)$. However, we use $\delta_x = \overrightarrow{a_1} \oplus a_2$ to calculate the probability of the occurrence of the bitwise AND of the RX pair regardless of the actual values of a_1 and a_2. Thus, we can redefine the RXD of an RX pair $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ as $\delta_x = \overrightarrow{a_1} \oplus a_2$ for the following reason.

Let there be another RX pair $(y \oplus b_1, \overrightarrow{y} \oplus b_2)$ such that $y \oplus b_1 = x \oplus a_1$ and $\overrightarrow{y} \oplus b_2 = \overrightarrow{x} \oplus a_2$ for some random variable $y \neq x$. In this case, we have

$$\delta_y = \overrightarrow{b_1} \oplus b_2 = \overrightarrow{(y \oplus x \oplus a_1)} \oplus \overrightarrow{y} \oplus \overrightarrow{x} \oplus a_2 = \overrightarrow{a_1} \oplus a_2 = \delta_x. \tag{22}$$

This means that the relationship between the constants (i.e., δ's) is sufficient to represent the RX pair and thus is also sufficient to trace the transition of δ's instead of RX pairs to search for an RXD trail. We also refer to this δ value as RXD and we denote an RXD trail from pair $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ to pair $(x \oplus b_1, \overrightarrow{x} \oplus b_2)$ such that $\delta_a = \overrightarrow{a_1} \oplus a_2$ and $\delta_b = \overrightarrow{b_1} \oplus b_2$ as $(\delta_a \longrightarrow \delta_b)$. To find a suitable RXD trail, we need to know how the RXDs are transformed by the operations used in the target cipher.

Because Simon uses only three operations, XOR, rotation, and the bitwise AND, we can discuss these operations. An RXD is transformed by XOR as follows. Let there be two RX pairs $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ and $(y \oplus b_1, \overrightarrow{y} \oplus b_2)$, and $\delta_x = a_1 \oplus a_2$ and $\delta_y = b_1 \oplus b_2$. If a new variable $z = x \oplus y$ is introduced, two RX pairs $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ and $(y \oplus b_1, \overrightarrow{y} \oplus b_2)$ are XORed into an RX pair $(z \oplus c_1, \overrightarrow{z} \oplus c_2)$ such that $c_1 = a_1 \oplus b_1$ and $c_2 = a_2 \oplus b_2$. Because $\delta_x = \overrightarrow{a_1} \oplus a_2$ and $\delta_y = \overrightarrow{b_1} \oplus c_2$, we have

$$\delta_z = \overrightarrow{c_1} \oplus c_2 = \delta_x \oplus \delta_y. \tag{23}$$

If a constant c is XORed into an RXD δ, the RX pair $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ is transformed into $(x \oplus a_1 \oplus c, \overrightarrow{x} \oplus a_2 \oplus c)$. Then, clearly,

$$\delta' = \overrightarrow{a_1 \oplus c} \oplus a_2 \oplus c = \delta \oplus \overrightarrow{c} \oplus c. \quad (24)$$

For the rotation operations, similar to the above case of XOR, if $y = x^{<l}$, then $\delta_y = \delta_x^{<l}$.

The transition of an RXD by the bitwise AND is as follows. Let $z = x \wedge y$; then, every ω satisfying $(\delta_x \vee \delta_y) \wedge \omega = \omega$ could be δ_z with the probability given in Theorem 2 In the case of Simon, the random variables x and y are dependent on each other such that the ω values that could be δ_z differ lightly from the general case, as shown in Theorem 3

3.2.2. Considerations. We took the following considerations into account during the search for the RXD trails of Simon.

(1) *Round indices.* The indices of start and end round of the characteristic should be specified because a rotational attack is basically in the related-key model and the δ values (RXDs) of the round keys vary according to the round constants z_i's XORed in the key schedule.

(2) *Including Rounds with an RXD Probability of 1.* If RXDs with two input words of encryption and k/w keywords for a round are all zero, we find some output RXD that is maximally $k/(w+2)$ rounds with a probability of 1. Thus, it is effective to search for RXD trails forward and backward beginning with such zero (or with a lower Hamming weight) states to find trails with a high probability.

(3) *Maximizing the Probability of the Next Round.* The probability of an RXD trail of a round is determined by the RXD of the left half of the input. Hence, if we can control the right half of the output of the current round, we can maximize the RX probability of the next round. According to Theorem 3, one input RXD can be transformed into several output RXDs through the bitwise AND, and because their probabilities are identical, we can choose one of them with a condition identical to that of the current round. Let $\delta_{i,L}$ and $\delta_{i,R}$ be the RXDs of the left and the right inputs of i-th round, respectively, and δ_K^i be the δ value of the i-th round key. To maximize the RX probability of the $i+1$-th round, $\delta_{i+1,L}$ should have a lower Hamming weight. Because

$$\delta_{i+1,L} = \delta_{i,R} \oplus \delta_{i,z} \oplus \delta_{i,L}^{<2} \oplus \delta_{i,K}, \quad (25)$$

where $\delta_{i,z}$ is the RXD of the output of the bitwise AND in the i-th round, we can choose $\delta_{i,z}$ for which minimizes the Hamming weight of $\delta_{i+1,L}$. Note that minimizing the Hamming weight of $\delta_{i+1,L}$ does not always guarantee the best RXD trail; however, we searched for RXD trails with such conditions in mind.

3.3. Rotational-XOR-Differences Trails of Simon. Putting the aforementioned considerations together, we searched for RXD trails with a high probability for Simon-32/64, 48/72, 48/96, 64/96, and 64/128. Initially, we let the pairs of both intermediate value and key state of the starting round have the δ values of Hamming weight 0 or 1. We then searched for the RXD trail with the maximum probability for each number of rounds by adding rounds forward and backward, while varying the starting round.

As a result, we can find numerous trails with the maximum probability for various starting round indices. Therefore, we can construct rectangle characteristics using short trails with high probabilities with considering the round indices.

4. Rotational Rectangle Attack on Simon

4.1. Rotational Rectangle Characteristic. In this section, we show that rotational-XOR-differences can be used to construct rectangle characteristics similar to differential characteristics by proving the following Theorem 4.

Theorem 4. *Le x and y be independent random variables and a_1, a_2, b_1, and b_2 be constants in $\mathbb{F}_2^n$ for some positive integer n. In addition, let $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$ and $(y \oplus b_1, \overrightarrow{y} \oplus b_2)$ be RX pairs with $\delta_a = \overrightarrow{a_1} \oplus a_2$ and $\delta_b = \overrightarrow{b_1} \oplus b_2$. If $(x \oplus a_1, \overrightarrow{y} \oplus b_2)$ forms an RX pair with RXD δ_c, then $(y \oplus b_1, \overrightarrow{x} \oplus a_2)$ also forms an RX pair and its RXD is $\delta_d = \delta_a \oplus \delta_b \oplus \delta_c$.*

Proof. Because we assumed that $(x \oplus a_1, \overrightarrow{y} \oplus b_2)$ is an RX pair and that its RXD is δ_c, we can assume that

$$\begin{aligned} z \oplus c_1 &= x \oplus a_1, \\ \overrightarrow{z} \oplus c_2 &= \overrightarrow{y} \oplus b_2, \end{aligned} \quad (26)$$

for a random variable z and for some constants c_1 and c_2 such that $\overrightarrow{c_1} \oplus c_2 = \delta_c$.

We will show that $y \oplus b_1 = z \oplus d_1$, $\overrightarrow{x} \oplus a_2 = \overrightarrow{z} \oplus d_2$ and $\overrightarrow{d_1} \oplus d_2 = \delta_a \oplus \delta_b \oplus \delta_c$.

According to this assumption, we have

$$\begin{aligned} y \oplus \overleftarrow{b_2} &= z \oplus \overleftarrow{c_2}, \\ \overrightarrow{x} \oplus \overrightarrow{a_1} &= \overrightarrow{z} \oplus \overrightarrow{c_1}. \end{aligned} \quad (27)$$

Thus, we have

$$\begin{aligned} y \oplus b_1 &= z \oplus \overleftarrow{c_2} \oplus \overleftarrow{b_2} \oplus b_1, \\ \overrightarrow{x} \oplus a_2 &= \overrightarrow{z} \oplus \overrightarrow{c_1} \oplus \overrightarrow{a_1} \oplus a_2. \end{aligned} \quad (28)$$

Therefore, if we let $d_1 = \overleftarrow{c_2} \oplus \overleftarrow{b_2} \oplus b_1$ and $d_2 = \overrightarrow{c_1} \oplus \overrightarrow{a_1} \oplus a_2$, we then have

$$\delta_d = \overrightarrow{d_1} \oplus d_2 = c_2 \oplus b_2 \oplus \overrightarrow{b_1} \oplus \overrightarrow{c_1} \oplus \overrightarrow{a_1} \oplus a_2 = \delta_a \oplus \delta_b \oplus \delta_c. \quad (29)$$

Accordingly, the proof is complete.

With Theorem 4 in mind, we introduce the rotational rectangle distinguisher as follows. Denote an encryption algorithm E with a key K by E_K. Suppose that E_K is a composition of $E0_K$ and $E1_K$ such that $E_K = E1_K \circ E0_K$. We have RXD trails $(\delta_a \longrightarrow \delta_b)$ satisfied with probability p for $E0$ and $(\delta_c \longrightarrow \delta_d)$ with probability q for $E1$.

Suppose that (P_0, P_1) and (Q_0, Q_1) are plaintext pairs whose δ values are both δ_a. The probability that the pairs of intermediate values $(E0_{K0}(P_0), E0_{K1}(P_1))$ and $(E0_{K2}(Q_0), E0_{K3}(Q_1))$ are both RX pair and their δ values are both δ_b is p^2.

According to Theorem 4, if $(E0_{K0}(P_0), E0_{K3}(Q_1))$ is an RX pair and its RXD is δ_c', then $(E0_{K2}(Q_0), E0_{K1}(P_1))$ is also an RX pair with RXD $\delta_c' \oplus \delta_a \oplus \delta_a = \delta_c'$. If $\delta_c = \delta_c'$, it holds

$$\begin{aligned}&(E1_{K0}(E0_{K0}(P_0)), E1_{K3}(E0_{K3}(Q_1))),\\&(E1_{K2}(E0_{K2}(Q_0)), E1_{K1}(E0_{K1}(P_1))),\end{aligned} \tag{30}$$

are both RX pairs with RXD δ_d with a probability of q^2.

Because the probability that $\delta_c = \delta_c'$ is 2^{-n} for block length n, two RX pairs (P_0, P_1) and (Q_0, Q_1) with δ_a are transformed into two RX pairs:

$$\begin{aligned}&(E1_{K0}(E0_{K0}(P_0)), E1_{K1}(E0_{K1}(P_1))),\\&(E1_{K2}(E0_{K2}(Q_0)), E1_{K3}(E0_{K3}(Q_1))),\end{aligned} \tag{31}$$

according to E_{K0}, E_{K1}, E_{K2}, and E_{K3} with a probability of $p^2 \cdot q^2 \cdot 2^{-n}$.

However, if E is a random permutation, the probability that the resulting four values form two RX pairs both with the expected RXDs is 2^{-2n}. Therefore, we can mount an RX rectangle attack when $2^{-2n} < p^2 \cdot q^2 \cdot 2^{-n}$. □

4.2. Constructing RX Rectangle Distinguishers. We have found many RXD trails for each of the Simon parameters that correspond to the probabilities presented in Table 2. Using these trails, we construct RX rectangle distinguishers by joining two RXD trails with consideration of round indices. As an example of Simon-32/64, we found that there exist eight-round RXD trails which start at eighth and sixteenth rounds. Therefore, we successfully combined them for the rectangle distinguisher with the maximum probability $(2^{-6} \cdot 2^{-6})^2 \cdot 2^{-32} = 2^{-56}$. However, for Simon-48/72, we did not find two eight-round trails that could be combined for a rectangle distinguisher to maximize the probability. Therefore, we use a nine-round trail starting at fifth round and a seven-round trail starting at fourteenth round for the rectangle distinguisher with a probability of $(2^{-4} \cdot 2^{-17})^2 \cdot 2^{-48} = 2^{-90}$. The number of rounds and the probability of the RX rectangle distinguisher for each of the Simon parameters are given in Table 3 and examples of RXD trails are presented in Tables 4 and 5.

4.3. Key Recovery Attack and Complexity. In this section, we present the key recovery attack framework on Simon with block sizes of 32, 48, and 64 using the RX rectangle distinguishers.

We assume the following:

p and q denote the probabilities of RXD trails for $E0$ and $E1$, respectively, and $p^2 \cdot q^2 = 2^{-m}$ for each version of Simon. Therefore, the probabilities of RX rectangle distinguishers are $2^{-(m+n)}$.

We add r_t rounds on top and three rounds at the bottom of the distinguisher for each version of Simon. Thus, the numbers of attacked rounds are $R = r_d + r_t + 3$, where r_d is the number of rounds of distinguishers for each version of Simon. Consequently, we attack round-reduced Simon from the i_s-th round to the $i_f = i_s + R - 1$-th round. The actual round indices of attacked rounds for each version of Simon can be found in Tables 4 and 5.

We use $N = 2^{((m+n+\alpha)/2)+\beta}$ plaintexts for adequate positive values α and β.

δ_i^L, δ_i^R, and δ_i^K denote the RXDs of the left half of an input, the right half of an input, and a round key of i-th round, respectively. $\delta_{init}^L || \delta_{init}^R$ and $\delta_{\text{final}}^L \| \delta_{\text{final}}^R$ are RXDs of an input and an output of the characteristic, respectively.

4.3.1. Generation of Pairs. Because we add r_t rounds on top, it is necessary to explain how to construct the quartets of the plaintexts for each key. We need to generate more than $N_q = 2^{m+n+\alpha}$ quartets to distinguish E from a random permutation when the expected number of right quartets is 2^α. To generate more than N_q quartets, we need two sets of pairs which contains at least $N_p = 2^{((m+n+\alpha)/2)}$ pairs.

We generate the first set of pairs as follows. Let Ω be set of plaintexts. First, we select a random plaintext from Ω and let this value be $(x \oplus a_1)$ for a fixed a_1. And then, we encrypt it for $r_t - 1$ rounds with a guessed subkey of $K0$ and let this value be $(y \oplus b_1)$. Next, we should define the intermediate value of the opposite side of a pair. By rotating $(y \oplus b_1)$ and adding an adequate RXD δ, the value is defined by $\overrightarrow{y} \oplus (\overrightarrow{b_1 \oplus \delta})$. Finally, we could have another plaintext of the pair by decrypting $\overrightarrow{y} \oplus (\overrightarrow{b_1 \oplus \delta})$ for $r_t - 1$ rounds with subkeys of the related $K1$. If the decryption result which is considered as $\overrightarrow{x} \oplus a_2$ is in Ω again, then the two plaintexts and corresponding ciphertexts by $K0$ and $K1$, respectively, are regarded as an RX pair. Similarly, another set of pairs are generated from the Ω and subkeys of $K2$ and $K3$. The numbers of elements in Ω required to obtain N_p pairs will be discussed later in terms of data complexity.

4.3.2. Attack Procedure. The key recovery attack against Simon-n/k proceeds as follows. Let F denote the round function of Simon; i.e., $F(x) = (x \lll 2) \oplus ((x \lll 1) \wedge (x \lll 8))$. Note that we assume that $r_t = 2$ for Simon-48/72 and Simon-64/96; otherwise, $r_t = 3$.

(1) Generate Ω of a sufficient size from the oracles.

(2) Guess $r_t - 1$ $n/2$-bit subkeys $(rk0_{i_s}, \ldots, rk0_{i_s+r_t-2})$ for $K0$, and for each guessed key, do the following:

Calculate the corresponding related subkeys $(rk1_{i_s}, \ldots, rk1_{i_s+r_t-2})$, $(rk2_{i_s}, \ldots, rk2_{i_s+r_t-2})$, and $(rk3_{i_s}, \ldots, rk3_{i_s+r_t-2})$, for $K1$, $K2$, and $K3$, respectively.

For each element x_{i_s} in Ω, do the following:

Encrypt x_{i_s} for $r_t - 1$ rounds with $rk0_{i_s}, \ldots, rk0_{i_s+1}$ to obtain $x_{i_s+r_t-1}$ ($\mathbf{r_t} - 1$ *round encryptions*).

Calculate $x_{i_s+r_t-1}'$ using $x_{i_s+r_t-1}$ and $\delta_{init}^L || \delta_{init}^R$.

Decrypt $x_{i_s+r_t-1}'$ for $r_t - 1$ rounds with $rk1_{i_s}, \ldots, rk1_{i_s+1}$ to obtain x_{i_s}' ($\mathbf{r_t} - 1$ *round decryptions*).

If $x_{i_s}' \in \Omega$, register (x_{i_s}, x_{i_s}') and their corresponding ciphertexts as a RX pair.

For each element y_{i_s} in Ω, do the following:

Table 2: Maximum probability of RXD trails for each number of rounds.

Round	Simon-32/64	Simon-48/72	Simon-48/96	Simon-64/96	Simon-64/128
1 ~ 5	1	1	1	1	1
6	1	2^{-4}	1	2^{-4}	1
7	2^{-3}	2^{-4}	2^{-4}	2^{-4}	2^{-4}
8	2^{-6}	2^{-10}	2^{-6}	2^{-10}	2^{-6}
9	2^{-10}	2^{-17}	2^{-10}	2^{-17}	2^{-10}
10	2^{-14}	2^{-20}	2^{-16}	2^{-24}	2^{-16}
11	2^{-18}	2^{-26}	2^{-22}	2^{-26}	2^{-22}
12	2^{-22}	2^{-32}	2^{-29}	2^{-32}	2^{-29}
13	2^{-29}	2^{-45}	2^{-36}	2^{-38}	2^{-36}
14	2^{-34}	2^{-52}	2^{-43}	2^{-45}	2^{-46}
15	—	—	2^{-51}	2^{-53}	2^{-54}
16	—	—	—	—	2^{-62}
17	—	—	—	—	2^{-74}

Table 3: Combinations of trails for constructing the RX rectangle distinguisher.

Simon-	Rounds ($E0 + E1 = E$)	Probability
32/64	$8 + 8 = 16$	$(2^{-6} \cdot 2^{-6})^2 \cdot 2^{-32} = 2^{-56}$
48/72	$7 + 9 = 16$	$(2^{-4} \cdot 2^{-17})^2 \cdot 2^{-48} = 2^{-90}$
48/96	$9 + 9 = 18$	$(2^{-10} \cdot 2^{-10})^2 \cdot 2^{-48} = 2^{-88}$.
64/96	$8 + 9 = 17$	$(2^{-10} \cdot 2^{-17})^2 \cdot 2^{-64} = 2^{-118}$
64/128	$9 + 10 = 19$	$(2^{-10} \cdot 2^{-16})^2 \cdot 2^{-64} = 2^{-116}$

Table 4: RXD trails for rectangle distinguishers of Simon-32/64, 48/72, and 48/96.

Simon-32/64			Simon-48/72			Simon-48/96		
	RXDes			RXDes			RXDes	
Round	Round States	Round Keys	Round	Round States	Round Keys	Round	Round States	Round Keys
8	0005 8015	8001	5	000005 c50018	f50000	12	000042 c0011b	c00003
9	0000 0005	0005	6	30000c 000005	a00007	13	000010 000042	000006
10	0000 0000	0000	7	000002 30000c	30000c	14	000004 000010	000000
11	0000 0000	0000	8	000001 000002	000006	15	000000 000004	000004
12	0000 0000	0000	9	000000 000001	000001	16	000000 000000	000000
13	0000 0000	0000	10	000000 000000	000000	17	000000 000000	000000
14	0000 0000	0005	11	000000 000000	000000	18	000000 000000	000000
15	0005 0000	f005	12	000000 000000	000004	19	000000 000000	000001
16	f011 0005	—	13	000004 000000	c00005	20	000001 000000	300005
			14	c00015 000004	—	21	300001 000001	—
16	0010 0041	0005	14	000002 30000c	300005	21	c00000 900007	100004
17	0004 0010	0000	15	000001 000002	000006	22	800000 c00000	c00003
18	0000 0004	0004	16	000000 000001	000001	23	000001 800000	800004
19	0000 0000	0000	17	000000 000000	000000	24	000000 000001	000001
20	0000 0000	0000	18	000000 000000	000000	25	000000 000000	000000
21	0000 0000	0000	19	000000 000000	000004	26	000000 000000	000000
22	0000 0000	0001	20	000004 000000	c00006	27	000000 000000	000000
23	0001 0000	3006	21	c00016 000004	—	28	000000 000000	000004
24	3002 0001	—	22	—	—	29	000004 000000	c00005
						30	c00015 000004	—

Encrypt y_{i_s} for $r_t - 1$ rounds with $rk2_{i_s}, \ldots, rk2_{i_s+1}$ to obtain $y_{i_s+r_t-1}$ ($\mathbf{r_t} - 1$ *round encryptions*).

Calculate $y'_{i_s+r_t-1}$ using $y_{i_s+r_t-1}$ and $\delta^L_{init} || \delta^R_{init}$.

Decrypt $y_{i_s+r_t-1}$ for $r_t - 1$ rounds with $rk3_{i_s}, \ldots, rk3_{i_s+1}$ to obtain y'_{i_s} ($\mathbf{r_t} - 1$ *round decryptions*).

If $y'_{i_s} \in \Omega$, register (y_{i_s}, y'_{i_s}) and their corresponding ciphertexts as a RX pair.

Using two sets of pairs, construct a set of quartets $(x_{i_s}, x'_{i_s}, y_{i_s}, y'_{i_s})$, and for each quartet, do the following:

For ciphertext pairs (x_{i_f+1}, y'_{i_f+1}) and (y_{i_f+1}, x'_{i_f+1}), calculate the δ values of $(F(x_{i_f+1}), F(y'_{i_f+1}))$ and $(F(y_{i_f+1}), F(x'_{i_f+1}))$. Using these values and $\delta^K_{i_f}$, calculate $\delta^L_{i_f-1}$. Then, first

check that δ^L_{final} is in the set of candidates for $\delta^R_{i_f-1}$ calculated by $\delta^L_{i_f-1}$, $\delta^L_{i_f}$, and $\delta^K_{i_f-1}$. Second, check that δ^R_{final} is in the set of candidates calculated for δ^R_{final} by $\delta^L_{i_f-1}$, $\delta^K_{i_f-2}$, and δ^L_{final} (*filtering ratio* 2^{-2t_0}, *three round decryptions*).

Guess the $n/2$-bit key $rk0_{i_f}$ and calculate the related subkeys $rk1_{i_f}$, $rk2_{i_f}$, and $rk3_{i_f}$, and for each guessed key, do the following:

For the remaining quartets, decrypt one round for $x_{i_f}, x_{i_f}', y_{i_f}, y_{i_f}'$. Using these values and $\delta^K_{i_f-1}$, calculate $\delta^L_{i_f-2}$ and check that $\delta^L_{i_f-2} = \delta^L_{\text{final}}$. And check that δ^R_{final} is in the set of candidates calculated for δ^R_{final} by $\delta^L_{i_f-1}$, $\delta^K_{i_f-2}$, and δ^L_{final} (*filtering ratio* 2^{-2t_1}, *three round decryptions*).

Guess the $n/2$-bit key $rk0_{i_f-1}$ and calculate the related subkeys $rk1_{i_f-1}$, $rk2_{i_f-1}$, and $rk3_{i_f-1}$, and for each guessed keys, do the following:

For the remaining quartets, decrypt one more round for $x_{i_f-1}, x_{i_f-1}', y_{i_f-1}, y_{i_f-1}'$. Using $\delta^K_{i_f-2}$, calculate $\delta^R_{i_f-2}$ and check $\delta^R_{i_f-2} = \delta^R_{final}$ (*filtering ratio* 2^{-2t_2}, *two round decryptions*).

Increase the counter of the current guessed key by the number of remaining quartets.

(3) Sort the guessed keys by the number of remaining quartets and exhaustively search the remaining key bits using highly ranked guessed keys.

The discard ratios for each filtering step are denoted by t_0, t_1, and t_2. These would be determined by the exact RXDs of the characteristic and should satisfy $t_0 + t_1 + t_2 < n$.

4.3.3. Data Complexity. The data complexity of this attack is estimated by the required number of elements in Ω. Using elements in Ω, we generate two sets of $N_p = 2^{((m+n+\alpha)/2)}$ pairs so that we have $N_q = 2^{m+n+\alpha}$ quartets. Let $l = ((m+n+\alpha)/2)$. We define pairs by choosing a text in Ω, encrypting it for the $r_t - 1$ round with a guessed key, adding some differences, and decrypting for $r_t - 1$ rounds with the related key. Therefore, we should assume that the processes after choosing a text are random permutations from $\{0,1\}^n$ to $\{0,1\}^n$, for counting the required number of elements in Ω. The question is that if we have set Ω of random texts in $\{0,1\}^n$ with $2^{l+\beta}$ elements and a random permutation f of $\{0,1\}^n$, what is the condition of β such that we have more than 2^l pairs $(x, f(x))$ where $f(x) \in \Omega$. Because we assume that f is a random permutation,

$$\Pr[f(x) \in \Omega] = \frac{2^{l+\beta}}{2^n} = 2^{l+\beta-n}, \tag{32}$$

for an $x \in \Omega$. Therefore, the expected number of pairs that we could have is $2^{l+\beta} \times 2^{l+\beta-n} = 2^{2l+2\beta-n}$. Given that we would like to have more than 2^l pairs, β should satisfy

$$2^{2l+2\beta-n} > 2^l. \tag{33}$$

So, we could have a lower bound of the β as follows:

$$\frac{n-l}{2} < \beta. \tag{34}$$

As we have assumed that $l = ((m+n+\alpha)/2)$, a straightforward computation gives the condition:

$$\frac{(n-m-\alpha)}{4} < \beta. \tag{35}$$

Thus, if we choose the minimum β, we can have the required number of pairs on average. Table 6 shows the data complexities of these attacks for each version of Simon.

4.3.4. Computational Complexity. At this stage, we calculate the computational complexities of this attack. At the beginning of the attack, we perform four rounds of encryption on N texts to define the pairs for each guessed $n/2$-bit key. We then filter out quartets with two rounds of decryption without key guessing. Next, we consume one round of encryption for each filtering step with $n/2$-bit key guessing. Finally, we should exhaustively search the remaining key bits. Therefore, we can estimate the computational complexity of this attack as follows while taking the above factors into account:

$$2^{((r_t-1)n/2)} \cdot \left(\frac{4N}{R} + \frac{3N_p}{R} + 2^{n/2} \cdot \left(2^{-2t_0} \cdot \frac{3N_p}{R} + 2^{n/2} \cdot \left(2^{-2t_0-2t_1} \cdot \frac{2N_p}{R}\right)\right)\right). \tag{36}$$

We have assumed that $N = 2^{((m+n+\alpha)/2)+\beta}$ and $N_p = 2^{((m+n+\alpha)/2)}$. Therefore, if we apply these assumptions to equation (36), then we have the following formula for computational complexity:

$$\frac{1}{R}\left(2^{(nr_t+m+\alpha+2\beta+4)/2} + 3 \cdot 2^{(nr_t+m+\alpha)/2} + 3 \cdot 2^{n(r_t+1)+m+\alpha-4t_0/2} + 2 \cdot 2^{(n(r_t+2)+m+\alpha-4t_0-4t_1)/2}\right). \tag{37}$$

Table 6 shows the computational complexities of these attacks as calculated using equation (37) along with the data complexities when $\alpha = 2$ and β has the minimum value. The filtering ratio t_0, which is most crucial with regard to the computational complexity among the ratios, is affected by how many types of RXDs of the outputs that could be produced by the round function where δ^L_{final} and a random δ value are the respective inputs. According to our

Table 5: RXD trails for rectangle distinguishers of Simon-64/96 and 64/128.

Simon-64/96			Simon-64/128		
	RXDes			RXDes	
Round	Round States	Round Keys	Round	Round States	Round Keys
15	30000000 30000005	f0000004	7	00000042 00000118	c0000000
16	00000001 30000000	30000005	8	00000010 00000042	00000006
17	00000001 00000001	00000005	9	00000004 00000010	00000000
18	00000000 00000001	00000001	10	00000000 00000004	00000004
19	00000000 00000000	00000000	11	00000000 00000000	00000000
20	00000000 00000000	00000000	12	00000000 00000000	00000000
21	00000000 00000000	00000004	13	00000000 00000000	00000000
22	00000004 00000000	c0000005	14	00000000 00000000	00000002
23	c0000015 00000004	—	15	00000002 00000000	60000005
			16	6000000d 00000002	—
23	00000006 c5000014	f5000000	16	40000014 60000012	60000002
24	3000000c 00000006	a0000004	17	00000041 40000014	40000002
25	00000002 3000000c	30000005	18	00000010 00000041	00000005
26	00000001 00000002	00000006	19	00000004 00000010	00000000
27	00000000 00000001	00000001	20	00000000 00000004	00000004
28	00000000 00000000	00000000	21	00000000 00000000	00000000
29	00000000 00000000	00000000	22	00000000 00000000	00000000
30	00000000 00000000	00000004	23	00000000 00000000	00000000
31	00000004 00000000	c0000005	24	00000000 00000000	00000002
32	c0000015 00000004	—	25	00000002 00000000	60000005
			26	6000000d 00000002	—

Table 6: Data and computational complexities for each version of Simon when β has the minimum value ($\alpha = 2$).

Simon-	32/64	48/72	48/96	64/96	64/128
n	32	48	48	64	64
i_s	5	3	9	13	4
i_f	26	23	32	34	28
r_t	3	2	3	2	3
m	24	42	40	54	52
R	22	21	24	22	25
Data complexity ($\lvert\Omega\rvert$)	$2^{30.5}$	2^{47}	$2^{46.5}$	2^{62}	$2^{61.5}$
Comp. complexity	$2^{60.4}$	$2^{69.1}$	$2^{92.3}$	$2^{91.8}$	$2^{123.0}$

investigation, $t_0 > (n/3)$ on average; thus, we assume that $t_0 = t_1 = (n/3)$.

4.3.5. Signal-to-Noise Ratio and Success Probabilities. Similar to differential cryptanalysis, rotational cryptanalysis uses randomly selected dataset so the attack works with probability less than or equal to one. Thus, we should calculate the success probability of each attack to make sure the possibility of the attack. By an earlier literature [23], the success probability of differential cryptanalysis could be calculated using signal-to-noise (S/N) ratio. We adopt that methodology for calculating the success probability of our attacks. We use the following equation for estimating success probabilities:

$$P_S = \Phi\left(\frac{\sqrt{\mu S_N} - \Phi^{-1}(1 - 2^{-a})}{\sqrt{S_N + 1}}\right), \tag{38}$$

where Φ is the cumulative distribution function of the standard normal distribution, μ denotes the number of right quartets, and we set the advantage a to 8. The S/N ratio S_N is calculated as follows:

$$S_N = \frac{2^{k_0} P_{char}}{\alpha_0 \beta_0}. \tag{39}$$

In the above equation, k_0 denotes the bit length of the target subkey, which is assumed to be equal to the bit length of the secret key. P_{char} denotes probability of characteristic, which is $2^{-(m+n)}$. α_0 is the average number of subkeys suggested by one analysed quartet. Since this attack generates $N_q = 2^{m+n+\alpha}$ quartets, $\alpha_0 = (2^{k_0}/N_q)$. β_0 is the ratio of filtering before key guessing but β_0 is fixed to 1 for all attacks because there is no filtering before key guessing. Therefore, the S/N ratio S_N is 2^α, and thus, the success probability P_S is 0.73 when $\alpha = 2$.

5. Conclusion

In this paper, we study how to apply cryptanalysis based on the rotational-XOR-difference approach to the block cipher Simon. We present a closed formula that is used to calculate the transition probability of an RXD trail according to the bitwise AND operation. Moreover, we demonstrate that we could construct the rectangle characteristic using RXD trails in a manner similar to how ordinary differential trails are used. Consequently, we could define a new RX rectangle attack and mount it onto some instances of the Simon family. Although our results are not the best for Simon to date, it is the first result for rotational cryptanalysis applied

to a non-ARX cipher and it would be a worthwhile endeavor to attempt to improve our approach or to apply to other ciphers based on bitwise AND.

6. RXD Trails for Rectangle Distinguishers

Tables 4 and 5 show actual RXD trails for which establish the rectangle distinguishers for each version of Simon, presented in Table 3.

Data Availability

The RXD trails used to support the findings of this study are included within Tables 4 and 5. More trails are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by an Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (Grant no. 2017-0-00267).

References

[1] N. Ferguson, S. Lucks, B. Schneier et al., *The Skein Hash Function Family. Submission to NIST (Round 3)*, Wiley, Hoboken, NJ, US, 2010.

[2] N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, "Chaskey: an efficient MAC algorithm for 32-bit microcontrollers," in *Selected Areas in Cryptography–SAC 2014*, pp. 306–323, Springer, Berlin, Germany, 2014.

[3] D. Hong, J. Sung, S. Hong et al., "HIGHT: a new block cipher suitable for low-resource device," in *Lecture Notes in Computer Science*, L. Goubin and M. Matsui, Eds., vol. 4249, pp. 46–59, Springer, Berlin, Germany, 2006.

[4] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The Simon and Speck families of lightweight block ciphers," in *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference*, San Francisco, CA, USA, June 2015.

[5] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: a 128-bit block cipher for fast encryption on common processors," in *Information Security Applications*, Y. Kim, H. Lee, and A. Perrig, Eds., vol. 8267, pp. 3–27, Springer, Berlin, Germany, 2014.

[6] D. Dinu, L. Perrin, A. Udovenko, V. Velichkov, J. Großschädl, and A. Biryukov, "Design strategies for ARX with provable bounds: Sparx and LAX," in *Advances in Cryptology–ASIACRYPT 2016*, J. H. Cheon and T. Takagi, Eds., vol. 10031, pp. 484–513, Springer, Berlin, Germany, 2016.

[7] C. De Cannière, O. Dunkelman, and M. KneževiC, "KATAN and KTANTAN-a family of small and efficient hardware-oriented block ciphers," in *Lecture Notes in Computer Science*, C. Clavier and K. Gaj, Eds., vol. 5747, pp. 272–288, Springer, Berlin, Germany, 2009.

[8] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, "The simeck family of lightweight block ciphers," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 307–29, Springer, Berlin, Germany, 2015.

[9] D. Khovratovich and I. Nikolić, "Rotational cryptanalysis of ARX," in *Fast Software Encryption*, S. Hong and T. Iwata, Eds., vol. 6147, pp. 333–346, Springer, Berlin, Germany, 2010.

[10] D. Khovratovich, I. Nikolić, J. Pieprzyk, P. Sokołowski, and R. Steinfeld, "Rotational cryptanalysis of ARX revisited," in *Fast Software Encryption*, G. Leander, Ed., vol. 9054, pp. 519–536, Springer, Berlin, Germany, 2015.

[11] T. Ashur and Y. Liu, "Rotational cryptanalysis in the presence of constants," *IACR Transactions on Symmetric Cryptology*, vol. 2016, no. 1, pp. 57–70, 2016.

[12] K. Qiao, L. Hu, and S. Sun, "Differential analysis on simeck and SIMON with dynamic key-guessing techniques," in *Information Systems Security and Privacy. ICISSP 2016*, O. Camp, S. Furnell, and P. Mori, Eds., Springer, Berlin, Germany, 2017.

[13] H. Chen and X. Wang, "Improved linear hull attack on round-reduced Simon with dynamic key-guessing techniques," in *Fast Software Encryption*, T. Peyrin, Ed., vol. 9783, Berlin, Germany, Springer, 2016.

[14] P. Derbez and P. A. Fouque, "Automatic search of meet-in-the-middle and impossible differential attacks," in *Advances in Cryptology-CRYPTO 2016. CRYPTO 2016*, M. Robshaw and J. Katz, Eds., vol. 9815, Berlin, Germany, Springer, 2016.

[15] L. Sun, K. Fu, and M. Wang, "Improved zero-correlation cryptanalysis on SIMON," in *Information Security and Cryptology. Inscrypt 2015*, D. Lin, X. Wang, and M. Yung, Eds., Springer, Berlin, Germany, 2016.

[16] R. Rohit and G. Gong, "Correlated sequence attack on reduced-round simon-32/64 and simeck-32/64," *IACR Cryptology ePrint Archive*, vol. 2018, p. 699, 2018.

[17] J.-K. Lee, B. Koo, and W.-H. Kim, "A general framework for the related-key linear attack against block ciphers with linear key schedules," in *Selected Area in Cryptography. SAC 2019. Lecture Notes in Computer Science*, K. G. Paterson and D. Stebila, Eds., vol. 11959, pp. 1–31, Springer, Berlin, Germany, 2020.

[18] D. Khovratovich, I. Nikolić, and C. Rechberger, "Rotational rebound attacks on reduced Skein," in *Advances in Cryptology-ASIACRYPT 2010*, M. Abe, Ed., vol. 6477, pp. 1–19, Springer, Berlin, Germany, 2010.

[19] P. Morawiecki, J. Pieprzyk, and M. Srebrny, "Rotational cryptanalysis of round-reduced Keccak," in *Fast Software Encryption*, S. Moriai, Ed., Springer, Berlin, Germany, pp. 241–262, 2014.

[20] M. Daum, *Cryptanalysis of hash functions of the MD4-family*, Ph.D. thesis, Ruhr-Universitat Bochum, Bochum, Germany, 2005.

[21] D. Wagner, "The boomerang attack," in *Fast Software Encryption. FSE 1999*, L. Knudsen, Ed., pp. 156–170, Springer, Berlin, Germany, 1999.

[22] E. Biham, O. Dunkelman, and N. Keller, "The rectangle attack-rectangling the serpent," in *Lecture Notes in Computer Science*, B. Pfitzmann, Ed., vol. 2045, pp. 340–357, Springer, Berlin, Germany, 2001.

[23] A. A. Selçuk, "On probability of success in linear and differential cryptanalysis," *Journal of Cryptology*, vol. 21, pp. 131–47, 2008.