Correlated Sequence Attack on Reduced-Round Simon-32/64 and Simeck-32/64

Raghvendra Rohit and Guang Gong

Department of Electrical and Computer Engineering, University of Waterloo,
Waterloo, Ontario, N2L 3G1, CANADA.
{rsrohit, ggong}@uwaterloo.ca

Abstract. In this paper, we propose a novel cryptanalytic technique called *correlated sequence attack* on block ciphers. Our attack exploits the properties of given key dependent sequences of length t to obtain other keyed sequences of same length with σ $(0 \leq \sigma < t)$ computations of the non-linear function. We call these sequences (σ, t)-*correlated sequences*, and utilize them in a meet-in-the-middle attack for $2t$ rounds. We apply this technique on Simon-32/64 and Simeck-32/64 block ciphers, construct $(1, 8)$-correlated sequences and present the first 25-round attack on both ciphers. Next, we analyze the 8-th element of these sequences by considering the key scheduling algorithms and differential properties, and show that the attack can be improved by two rounds with the same complexities as of the 25-round attack. Overall, our technique is used to attack up to 27 rounds of both Simon-32/64 and Simeck-32/64 with a time complexity less than that of average exhaustive search and data complexity of 3.
Our attack extends the number of previously attacked rounds by 4 and has a success probability 1. This reduces the security margin of both these ciphers to 16%. Up to our knowledge, this is currently the best attack on Simon-32/64 and Simeck-32/64.

Keywords: Correlated sequences, Simon, Simeck, Meet-in-the-middle attack

1 Introduction

Over the past few years, lightweight cryptography has been actively discussed in academia and industry to target the challenges posed by resource constrained environments such as RFID (EPC tags and NFC), IoT devices and sensor networks. As a result, several lightweight block ciphers such as HIGHT [15], PRESENT [7], LED [14], KATAN and KTANTAN [10], PICCOLO [23], SIMON and SPECK [5], SIMECK [30], SKINNY [6], GIFT [3] have been proposed. Recently, National Institute of Standards and Technology (NIST) intiated a call for the standardization of lightweight cryptographic primitives due to the noticeable lack of such standards [1, 20].

Among most of the aforementioned block ciphers, Simon [5] designed by the US National Security Agency (NSA) in 2013, acheive overwhelming performance

in hardware due to its simple non-linear round function which consists of bitwise XORs and ANDs only. Later in CHES 2015, Yang *et al.* [30] proposed Simeck that has a smaller hardware footprint than Simon, by combining the good design components of both Simon and Speck.

Initially, the designers of Simon neither provided design rationale nor security evaluation, and Simeck adopts the similar structure. Accordingly, both ciphers attracted a lot of attention from the cryptographic community. Several papers have analyzed their security and investigated the parameter choices of the round function to get a deeper understanding of design rationale of these ciphers [2, 8, 9, 13, 16, 17, 18, 19, 21, 22, 24, 26, 28, 27, 31]. Currently, the best cryptanalytic results on Simon and Simeck are reduced-round differential/linear and integral attacks. As a result, the average security margin[1] of ten variants of Simon and three variants of Simeck is 29% and 20%, respectively [4, 8, 21, 22]. The smaller versions, namely Simon-32/64 and Simeck-32/64, with blocksize and keysize, 32 and 64-bit, respectively, have security margin of 28%.

In this work, we propose a new attack called *correlated sequence attack* and show that the application of this attack on Simon-32/64 and Simeck-32/64 reduces their security margin to only 16%. Table 1 depicts a summary of the cryptanalytic results on Simon-32/64 and Simeck-32/64. In what follows, we list the contributions of this paper.

- We present a novel attack technique called correlated sequence attack on block ciphers. For a fixed key, we consider t rounds of cipher as a keyed sequence of length t, i.e, $(s_0, s_1, \ldots, s_{t-1})$, where s_i is the state at i-th round. Our attack exploits the properties of given keyed sequences of length t to obtain other keyed sequences of same length with σ $(0 \leq \sigma < t)$ computations of the non-linear function. We call these sequences (σ, t)-correlated sequences. We show how to utilize these sequences in a meet-in-the-middle attack for $2t$ rounds. Unlike other attacks [16, 17, 22, 29], this attack works without the use of dedicated programming tools such as SAT/SMT or Mixed Integer Linear Programming solvers.
- We apply the method of correlated sequences on Simon-32/64 and Simeck-32/64, and provide the theoretical construction of $(1, 8)$-correlated sequences. We show that all keyed sequences can be computed linearly from the keyed sequences whose 6-th element is zero.
- We use $(1, 8)$-correlated sequences for 6 encryption and 6 decryption rounds in a meet-in-the-middle attack [12] and present the first 24 and 25 round key recovery attack on Simon-32/64 and Simeck-32/64.
- By incorporating the properties of correlated sequences, key scheduling algorithms and one round differentials, we show that 8-th element of these sequences can take atmost 2^{15} values. As a result, we improve the key recovery attack by 2 rounds with the same complexities as of the 25-round attack.

[1] $1 - \frac{\#\text{ attacked rounds}}{\#\text{ full rounds}}$

Table 1: Summary of attacks on Simon-32/64 and Simeck-32/64

Attack	Cipher	# attacked rounds / 32	Data	Memory (Bytes)	Time	Success rate
Differential	Simon-32/64 [27]	21	2^{31}	-	$2^{55.25}$	0.51
	Simon-32/64 [21]	22	2^{32}	-	$2^{58.76}$	0.315
	Simeck-32/64 [17]	19	2^{31}	2^{33}	2^{40}	-
	Simeck-32/64 [21]	22	2^{32}	-	$2^{57.9}$	0.417
Linear	Simon-32/64 [8]	23	$2^{31.19}$	-	$2^{61.84}$	0.277
	Simeck-32/64 [22]	23	$2^{31.91}$	-	$2^{61.78}$	0.456
Integral	Simon-32/64 [28]	21	2^{31}	2^{54}	2^{63}	1
	Simon-32/64 [13]	22	2^{31}	$2^{55.8}$	2^{63}	1
	Simon-32/64 [9]	24	2^{32}	$2^{33.64}$	2^{63}	1
	Simeck-32/64 [31]	21	2^{31}	$2^{46.22}$	2^{63}	1
Impossible Differential	Simon-32/64 [11]	20	2^{32}	$2^{45.5}$	$2^{62.8}$	-
	Simeck-32/64 [30]	20	2^{32}	2^{58}	$2^{62.5}$	-
Zero correlation	Simon-32/64 [25]	21	2^{32}	2^{31}	$2^{59.4}$	-
	Simeck-32/64 [32]	21	2^{32}	$2^{47.67}$	$2^{58.78}$	-
Meet-in-the-middle	Simon-32/64 [24]	18	8	2^{52}	$2^{62.57}$	1
Correlated sequence attack Sections 5 and 6	Simon-32/64	**24**	3	2^{50}	$2^{62.87}$	1
		25	3	2^{50}	$2^{62.94}$	1
		26	3	2^{50}	$2^{62.88}$	1
		27	3	2^{50}	$2^{62.94}$	1
	Simeck-32/64	**24**	3	2^{50}	$2^{62.87}$	1
		25	3	2^{50}	$2^{62.94}$	1
		26	3	2^{50}	$2^{62.88}$	1
		27	3	2^{50}	$2^{62.94}$	1

The rest of the paper is organized as follows. In Section 2, we define the notations used throughout the paper and review the specifications of Simon and Simeck. Sections 3 and 4 present the definitions and basic properties of the correlated sequence attack and the construction of such sequences for Simon-32/64 and Simeck-32/64, respectively. In Section 5, we show how we use correlated sequences to mount 25-round key recovery attack on Simon-32/64 and Simeck-32/64. In Section 6, we show that the attack can be improved by 2 rounds leading to 27 round key recovery attack. Finally, the paper is concluded in Section 7.

2 Preliminaries

In this section, we give a brief description of Simon and Simeck. The notations used throughout the paper are defined in Table 2.

2.1 Specification of Simon and Simeck

Simon-$2n/mn$, where $2n$ and mn denote the blocksize and key length, respectively, is a family of block ciphers proposed by NSA in 2013 [5]. It

Table 2: Notations

Notation	Description
$+$	bitwise XOR
$\&$	bitwise AND
n	wordsize
$\mathcal{K}$	key space
$\mathbb{F}_2$	{0,1}
$\mathbb{F}_2^n$	n dimensional vector space over $\mathbb{F}_2$
L^i	left cyclic shift operator, i.e., for $x \in \mathbb{F}_2^n$, $L^i(x) = (x_i, x_{i+1}, \ldots, x_{n-1}, x_0, x_1, \ldots, x_{i-1})$
C_s	coset modulo $2^n - 1$, i.e., $C_s = \{s, 2s, \ldots, 2^{n_s-1}s\}$ where n_s is the smallest number such that $s \equiv 2^{n_s} s \bmod 2^n - 1$, and s is the smallest number in C_s and denotes the coset leader
$\lvert S \rvert$	cardinality of set S
$A[i]$	i-th element of A
$Img(f)$	Image set of f

adopts a Non-Linear Feedback Shift Register (NLFSR) based structure as depicted in Figure 1. At each round, the state is updated non-linearly using the function $f_{(a,b,c)}(x) = L^a(x)\&L^b(x) + L^c(x)$ where $(a, b, c) = (8, 1, 2)$. For r-round cipher, the $(i + 2)$-th element of NLFSR sequence is given by $s_{i+2} = f_{(8,1,2)}(s_{i+1}) + s_i + k_i$ where $k_i \in \mathbb{F}_2^n$ is the i-th round subkey[2] and $0 \leq i < r$. Finally, the ciphertext is the r-th state of NLFSR, i.e., (s_{r+1}, s_r).

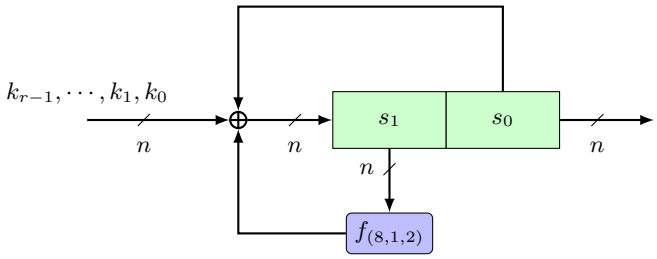

Fig. 1: Simon block cipher

Simeck-$2n/mn$ was proposed in CHES 2015 by Yang *et al.* [30] and adopts a similar structure as Simon. However, it has more efficient and compact hardware implementation because of reuse of the round function in the key scheduling algorithm. The shift parameters of Simeck are given by $(a, b, c) = (5, 0, 1)$.

[2] $k_0, k_1, \ldots, k_{m-1}$ are first m n-bit words of key.

For $n = 16$ and $m = 4$, $r = 32$ and the subkeys for $i \geq 4$ are calculated as follows.

Simon-32/64 key scheduling algorithm. $k_{i+4} = Z_{i-4} + k_i + k_{i+1} + L^{15}(k_{i+1}) + L^{13}(k_{i+3}) + L^{12}(k_{i+3})$.

Simeck-32/64 key scheduling algorithm. $k_{i+4} = Z_{i-4} + f_{(5,0,1)}(k_{i+1}) + k_i$.

The attack presented in this paper is not affected by the constants Z_i and the reader may refer to [5, 30] for more details of their respective key scheduling algorithms. From now onwards, we refer to $f_{(a,b,c)}$ as Simon-like non-linear function unless the parameter set (a, b, c) is explicitly mentioned.

3 Correlated Sequence Attack

In this section, we formally introduce the correlated sequence attack. We first define the correlated sequences of block ciphers. Next, we show how to use such sequences in a meet-in-the-middle (MitM) attack.

Consider an n-bit block cipher with r rounds and mn-bit master key $k = (k_0, k_1, \ldots, k_{m-1})$ as depicted in Figure 2. Let s_i denote the state at i-th round. Then, for $0 \leq i < r$, $s_{i+1} = \mathsf{rf}(s_i, k_i)$ where rf denotes the round function, and is generally a composition of two functions, namely i) a linear function χ and ii) a non-linear function ρ.

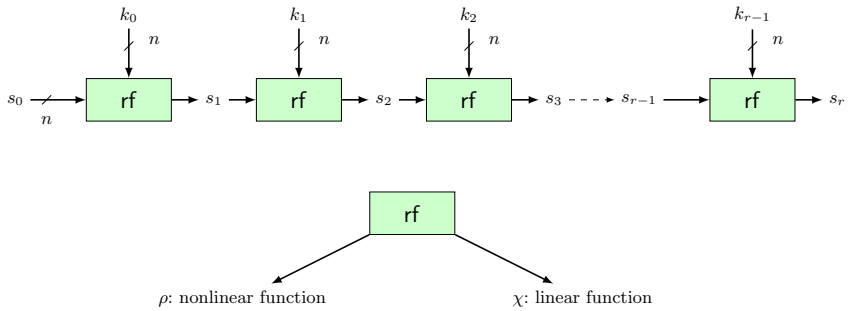

Fig. 2: Generic diagram of a block cipher

3.1 Correlated sequences

Definition 1 (Keyed sequence). *Given k and $1 \leq t < r$. We say $S_{(k,t)} = (s_0, s_1, \ldots, s_{t-1})$ is a keyed sequence of length t, if $s_{i+1} = \mathsf{rf}(s_i, k_i)$ for $0 \leq i < t - 1$.*

From Definition 1, it is clear that we need to compute rf t times to obtain $S_{(k,t)}$. This implies that ρ is computed t times in total. Thus, to obtain another sequence $S_{(k',t)}$ of same length t, the worst case is to compute ρ exactly t times. The idea of correlated sequences is "**Given $S_{(k,t)}$ and $k' \neq k$, obtain the sequence $S_{(k',t)}$ by computing the non-linear function ρ atmost t times.**"

We now present a toy example to illustrate this idea before providing the formal definition.

Example 1. Consider a 4-bit toy Simon-like block cipher with 8-bit blocksize and 16-bit key as depicted in Figure 3. Let the non-linear function is given by $\rho(x) = L(x)\&x + L^2(x)$ where $x \in \mathbb{F}_2^4$. The length seven keyed sequences are given in Table 3. We note the following observations from Table 3.

1. For all k, $s_4 = k_2$, $s_5 = 0$ and $s_6 = k_2 + k_4$.
2. For all k', $s'_4 = k'_2$, $s'_5 = 1$ and $s'_6 = k'_2 + k'_4 + \rho(1)$.
3. For each row, $k'_3 = k_3 + 1$ and $s'_6 = s_6 + k_4 + k'_4 + \rho(1)$.

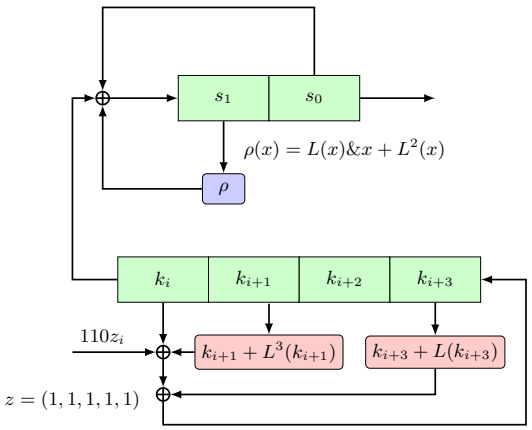

Fig. 3: 4-bit toy Simon-like cipher

We now define the correlated sequences in Definition 2.

Definition 2 ((σ, t)-correlated sequences). *Given $S_{(k,t)}$ and $0 \leq \sigma < t$. We say $S_{(k,t)}$ and $S_{(k',t)}$ are (σ, t)-correlated sequences if $S_{(k',t)}$ can be obtained from $S_{(k,t)}$ by computing the non-linear function ρ exactly σ times.*

Remark 1. $\sigma = 0 \implies S_{(k,t)}$ and $S_{(k',t)}$ are linearly related.

Definition 3 (Correlated keys). *Given $S_{(k,t)}$. We define correlated keys as the set $\mathrm{CK}(k) = \{k' \mid S_{(k,t)}$ and $S_{(k',t)}$ are $(0, t)$-correlated sequences$\}$.*

For example, in Table 3, for each row $S_{(k,t)}$ and $S_{(k',t)}$ are $(1, 7)$ correlated sequences, $|\mathrm{CK}((0, 0, 0, 0))| = 15$ (gray colored rows) and $|\mathrm{CK}((0, 0, 0, 1))| = 15$ (light gray rows). Thus, to obtain all 32 sequences, we only need to compute $S_{((0,0,0,0),5)}$ and $S_{((0,0,0,1),5)}$ which requires only one computation of ρ.

3.2 MitM attack using correlated sequences

Let (s_0, s_r) denote the plaintext and ciphertext pair encrypted with the mn-bit master key k. As depicted in Figure 4, we first use s_0 to construct (σ, t_1)-correlated sequences and their corresponding $\mathrm{CK}(\cdot)$ for t_1 rounds. Next, starting with s_r, we follow the same approach. We then do partial encryption

Table 3: Keyed sequences

k_0	k_1	k_2	k_3	k_4	s_0	s_1	s_2	s_3	s_4	s_5	s_6	k'_0	k'_1	k'_2	k'_3	k'_4	s'_0	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6
0	**0**	**0**	**0**	**13**	**0**	**0**	**0**	**0**	**0**	**0**	**13**	**0**	**0**	**0**	**1**	**14**	**0**	**0**	**0**	**0**	**0**	**1**	**10**
0	0	1	4	1	0	0	0	0	1	0	0	0	0	1	5	2	0	0	0	0	1	1	7
0	0	2	8	4	0	0	0	0	2	0	6	0	0	2	9	7	0	0	0	0	2	1	1
0	0	3	14	14	0	0	0	0	3	0	13	0	0	3	15	13	0	0	0	0	3	1	10
0	0	4	1	14	0	0	0	0	4	0	10	0	0	4	0	13	0	0	0	0	4	1	13
0	0	5	5	2	0	0	0	0	5	0	7	0	0	5	4	1	0	0	0	0	5	1	0
0	0	6	13	11	0	0	0	0	6	0	13	0	0	6	12	8	0	0	0	0	6	1	10
0	0	7	11	1	0	0	0	0	7	0	6	0	0	7	10	2	0	0	0	0	7	1	1
0	0	8	2	11	0	0	0	0	8	0	3	0	0	8	3	8	0	0	0	0	8	1	4
0	0	9	7	4	0	0	0	0	9	0	13	0	0	9	6	7	0	0	0	0	9	1	10
0	0	10	10	2	0	0	0	0	10	0	8	0	0	10	11	1	0	0	0	0	10	1	15
0	0	11	13	11	0	0	0	0	11	0	0	0	0	11	12	8	0	0	0	0	11	1	7
0	0	12	11	1	0	0	0	0	12	0	13	0	0	12	10	2	0	0	0	0	12	1	10
0	0	13	14	14	0	0	0	0	13	0	3	0	0	13	15	13	0	0	0	0	13	1	4
0	0	14	7	4	0	0	0	0	14	0	10	0	0	14	6	7	0	0	0	0	14	1	13
0	0	15	0	13	0	0	0	0	15	0	2	0	0	15	1	14	0	0	0	0	15	1	5

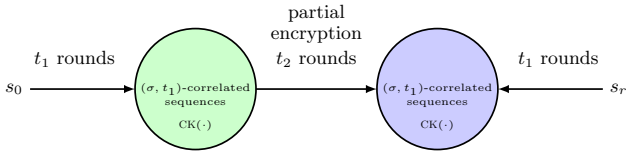

Fig. 4: MitM attack using correlated sequences

for t_2 rounds starting from t_1-th round and match the state values at (t_1+t_2)-th round.

Let T_f (resp. T_b) denote the number of computations of ρ to construct (σ, t_1)-correlated sequences and their corresponding CK$(\cdot)$ in forward (resp. backward) direction. Then, the time complexity in terms of the number of computations of ρ is given by $T_{online} = T_f + T_b + 2^{|\mathcal{K}|} \times \frac{t_2}{r}$. Clearly, $T_{online} < 2^{|\mathcal{K}|}$.

Remark 2. The above attack returns $2^{n(m-1)}$ keys that map s_0 to s_r. The correct key can then be found out by performing an exhaustive search on the remaining known $\lceil \frac{mn}{n} \rceil - 1$ plaintext-ciphertext pairs.

4 Correlated Sequences of Simon-$2n/4n$ and Simeck-$2n/4n$

In this section, we show the construction of correlated sequences of Simon-$2n/4n$ and Simeck-$2n/4n$. We first look at the theoretical properties of non-linear function $f_{(a,b,c)}$. Next, we use these properties to construct $(1, 8)$-correlated sequences. We assume that $a \neq b \neq c$.

4.1 Properties of Simon-like non-linear function

Property 1. Let s be the coset leader corresponding to the coset C_s, then for $0 \leq i < |C_s|$, we have

1. $f_{(a,b,c)}(L^i(s)) = L^i(f_{(a,b,c)}(s))$
2. $f_{(a,b,c)}(s) = L^{a-1}(s) + L^{b-1}(s) + L^{c-1}(s)$, if $s = \underbrace{011\ldots1}_{n}$.

Property 2. Let $s = \underbrace{0101\ldots01}_{n}$ and a, b are not both simultaneously even or odd, then

$$f_{(a,b,c)}(s) = \begin{cases} s & \text{if } c \equiv 0 \bmod 2 \\ L(s) & \text{otherwise} \end{cases}$$

Properties 1 and 2 imply that we need to compute the values of $f_{(a,b,c)}$ for $\frac{2^n-1}{n}$ coset leaders only. However, as $f_{(a,b,c)}$ is quadratic and the only linear term involved in it is $L^c(.)$, we have $f_{(a,b,c)}(x) = L^c(x) + z$ for all $x \in \mathbb{F}_2^n$ and some constant $z \in \mathbb{F}_2^n$. As a result, we can partition the coset leaders based on the values of z. Since, $f_{(a,b,c)}$ is linear on each partition, we call such partition as *z-linear segment set* and formally define it in Definition 4 as follows.

Definition 4 (z-linear segment set). *The z-linear segment set of $f_{(a,b,c)}$ is the set of coset leaders CL_z given by $CL_z = \{s \mid f_{(a,b,c)}(s) + L^c(s) = z\}$.*

Table 4 lists the z-linear segment sets for $n = 8$ and $(a, b, c) = (8, 1, 2)$, while the number of z-linear segments (denoted by N_z) for varying n are presented in Table 5. (Note that since $n = 8$, the shifts $(8, 1, 2)$ is equivalent to $(0, 1, 2)$.)

Example 2. In Table 4, consider $z = 2$ and $3 \in CL_2$. Then, for all $x \in C_3 = \{3, 6, 12, 24, 48, 96, 192, 129\}$, $f_{(8,1,2)}$ is computed as follows.

x	$f_{(8,1,2)}(x)$
3	$L^2(3) + 2 = 14$
6	$L^2(6) + L(2) = 28$
12	$L^2(12) + L^2(2) = 56$
24	$L^2(24) + L^3(2) = 112$
48	$L^2(28) + L^4(2) = 224$
96	$L^2(96) + L^5(2) = 193$
192	$L^2(192) + L^6(2) = 131$
129	$L^2(129) + L^7(2) = 7$

Table 4: z-linear segment sets for $n = 8$ and $(a, b, c) = (8, 1, 2)$

z	CL_z	z	CL_z
0	{0, 1, 5, 9, 17, 21, 37, 85}	2	{3, 11, 19, 43}
6	{7, 23, 39, 87 }	8	{13, 45}
14	{15}	16	{25}
18	{27, 91 }	24	{29}
30	{95, 31}	32	{53}
34	{51}	38	{55}
50	{59}	56	{61}
62	{63}	78	{111}
102	{119}	126	{127}
255	{255}	13	{47}

Table 5: Number of z-linear segment sets for varying n

n	# coset leaders	N_z	
		(a, b, c)	
		$(8, 1, 2)$	$(5, 0, 1)$
8	36	20	17
10	108	42	14
12	352	119	119
14	1182	50	287
16	4116	909	798

4.2 Construction of $(1, 8)$-correlated sequences

Let (s_0, s_1) be any random $2n$-bit value and $\mathbb{K}_{(k_0,k_1)} = \{(k_0, k_1, k_2, k_3) \mid (k_2, k_3) \in \mathbb{F}_2^n \times \mathbb{F}_2^n\}$ be the set of 2^{2n} keys with k_0 and k_1 fixed to some constant value. For $t \geq 6$ and $0 \leq i < 2^n$, define

$$\mathbb{P}(i, t, \mathbb{K}_{(k_0,k_1)}) = \{(k, S_{(k,t)}) \mid k \in \mathbb{K}_{(k_0,k_1)} \text{ and } s_5 = i\}$$

as the set of keys and their corresponding sequences which maps s_5 to i.

We start with the simpler case, i.e., $s_5 = 0$. First, we construct $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$ and then show how to construct $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ using the knowledge of $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$.

4.2.1 Construction of $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$. We divide the construction of $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$ into 3 steps, namely i) Finding $\mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$, ii) Obtaining $\mathbb{P}(0, 7, \mathbb{K}_{(k_0,k_1)})$ from $\mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$, and iii) Obtaining $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$ from

$\mathbb{P}(0, 7, \mathbb{K}_{(k_0,k_1)})$. For each step, we denote the number of computations of $f_{(a,b,c)}$ by T_{step}. We now present the details of each step as follows.

Step 1: Finding $\mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$. We note that $\forall k \in \mathbb{K}_{(k_0,k_1)}$, $S_{(k,4)}$ is a constant sequence and requires only 2 computations of $f_{(a,b,c)}$. Hence, finding the keys for which $s_5 = 0$ is equivalent to solve $f_{(a,b,c)}(X + k_2) = k_3 + s_3$ where $X = f_{(a,b,c)}(s_3) + s_2$. We use z-linear segments (see Definition 4) to solve this equation. As a result, $T_{step_1} = 2 + N_z$.

Remark 3. $|\mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})| = 2^n$, as $s_4 = X + k_2$ can take all 2^n distinct values.

Step 2: Obtaining $\mathbb{P}(0, 7, \mathbb{K}_{(k_0,k_1)})$ from $\mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$. Let $(k, S_{(k,6)}) \in \mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$ and consider the following relation $s_4 + s_6$. We have $s_4 + s_6 = s_4 + f_{(a,b,c)}(s_5) + s_4 + k_4 = s_4 + 0 + s_4 + k_4 \implies s_6 = s_4 + k_4$. Thus, $T_{step_2} = 0$.

Step 3: Obtaining $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$ from $\mathbb{P}(0, 7, \mathbb{K}_{(k_0,k_1)})$. Let $(k, S_{(k,7)})$, $(k', S_{(k',7)}) \in \mathbb{P}(0, 7, \mathbb{K}_{(k_0,k_1)})$ be such that $s'_4 = s_6$. Thus, $f_{(a,b,c)}(s_6) = f_{(a,b,c)}(s'_4) = k'_3 + s_3$ (follows from Step 1). We note that such a pair always exists. This follows directly from Remark 3. We now evaluate s_7 as follows: $s_7 = f_{(a,b,c)}(s_6) + s_5 + k_5 = s_3 + k'_3 + 0 + k_5 = s_3 + k'_3 + k_5$. Hence, $T_{step_3} = 0$.

4.2.2 Computing $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ from $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$. We could use the similar construction shown above to get $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ for $1 \leq i < 2^n$. However, this would require $2^n(2 + N_z)$ computations of $f_{(a,b,c)}$ in total. In Theorem 1, we show how to reduce this number to $(2 + 2N_z)$.

Lemma 1. *Let* $\mathbf{I}_{(k_0,k_1)} = \{k_3 | ((k_0, k_1, k_2, k_3), S_{((k_0,k_1,k_2,k_3),6)}) \in \mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})\}$[3] *and* $k = (k_0, k_1, 0, \mathbf{I}_{(k_0,k_1)}[0])$. *Let* $1 \leq k_2 < 2^n$ *and* $k' = (k_0, k_1, k_2, \mathbf{I}_{(k_0,k_1)}[k_2])$ *be such that* $k \neq k'$. *Then, the following hold.*

1. $S_{(k,4)} = S_{(k',4)}$
2. $s'_4 = X + k_2$, *where* $X = f_{(a,b,c)}(s_3) + s_2$
3. $s'_6 = s_6 + k_4 + k'_4 + k_2$
4. $s'_7 = s_3 + k'_5 + \mathbf{I}_{(k_0,k_1)}[k_2 + k'_4]$
5. $s_7 = s_3 + k_5 + \mathbf{I}_{(k_0,k_1)}[k_4]$
6. $|\mathrm{CK}(k)| = 2^n - 1$

Proof. 1. Since s_0, s_1, s_2 and s_3 are independent of k_3, it follows that $S_{(k,4)} = S_{(k',4)}$.
2. We have $s'_4 = f_{(a,b,c)}(s'_3) + s'_2 + k_2 = f_{(a,b,c)}(s_3) + s_2 + k_2 \implies s'_4 = X + k_2$.
3. Consider the following relation $s'_6 + s_6$.

$$\begin{aligned} s'_6 + s_6 &= f_{(a,b,c)}(s'_5) + s'_4 + k'_4 + f_{(a,b,c)}(s_5) + s_4 + k_4 \\ &= s'_4 + k'_4 + s_4 + k_4 \\ &= X + k_2 + k'_4 + X + k_4 \implies s'_6 = s_6 + k_4 + k'_4 + k_2. \end{aligned}$$

[3] $|\mathbf{I}_{(k_0,k_1)}| = 2^n$ (see Remark 3). Thus, $\mathbf{I}_{(k_0,k_1)}$ can have multiple values of k_3.

4. Note that by Step 3 of construction of $\mathbb{P}(0, 8, \mathbb{K}_{(k_0,k_1)})$, we have $s'_7 = s_3 + k'_5 + k''_3$. Hence, we need to find the index j such that $\mathbf{I}_{(k_0,k_1)}[j] = k''_3$. This is equivalent to finding j for which $s''_4 = s'_6$. Since, $s'_6 + s'_4 = k'_4 \implies j = k'_2 + k'_4 = k_2 + k'_4$.
5. Follows directly from part 4.
6. The proof is trivial, as for all $2^n - 1$ values of k_2, s'_6 and s'_7 can be computed linearly. □

Theorem 1. *Let* $k = (k_0, k_1, 0, \mathbf{I}_{(k_0,k_1)}[0])$, $(k, S_{(k,6)}) \in \mathbb{P}(0, 6, \mathbb{K}_{(k_0,k_1)})$ *and* $\tilde{k} = (k_0, k_1, 0, \mathbf{I}_{(k_0,k_1)}[0] + i)$ *where* $1 \leq i < 2^n$. *Then, the following hold.*

1. $S_{(k,5)} = S_{(\tilde{k},5)}$
2. $(\tilde{k}, S_{(\tilde{k},6)}) \in \mathbb{P}(i, 6, \mathbb{K}_{(k_0,k_1)})$
3. $\tilde{s_6} = s_6 + k_4 + \tilde{k_4} + f_{(a,b,c)}(i)$
4. $\tilde{s_7} = s_3 + i + \tilde{k_5} + \mathbf{I}_{(k_0,k_1)}[\tilde{k_2} + \tilde{k_4} + f_{(a,b,c)}(i)]$
5. $|\mathrm{CK}(k)| = |\mathrm{CK}(\bar{k})| = 2^n - 1$

Proof. The proof of 1, 2, 3 and 4 is similar to Lemma 1. For part 5, note that for $1 \leq j < 2^n$, $(k_0, k_1, j, \mathbf{I}_{(k_0,k_1)}[j]) \in \mathrm{CK}(k) \iff (k_0, k_1, j, \mathbf{I}_{(k_0,k_1)}[j] + i) \in \mathrm{CK}(\bar{k})$. This follows because $s_5 + \bar{s_5} = k_3 + \bar{k_3} \implies k_3 + \bar{k_3} = i$. Thus, $|\mathrm{CK}(k)| = |\mathrm{CK}(\bar{k})| = 2^n - 1$. □

A brief comparison of different approaches with the number of computations of $f_{(a,b,c)}$ to obtain $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ is provided in Table 6.

Table 6: Comparison of different approaches with the number of computations of $f_{(a,b,c)}$

Approach	# computations of $f_{(a,b,c)}$	
	(a, b, c)	
	$(8, 1, 2)$	$(5, 0, 1)$
Naive	$2^{32} \times 6$	$2^{32} \times 6$
Theorem 1 and z-linear segment sets	$(2 + 1818)$	$(2 + 1596)$

5 Key Recovery Attack on 25 rounds Simon-32/64 and Simeck-32/64

In this section, we show the key recovery attack procedure on 25-round[4] Simon-32/64 and Simeck-32/64. We utilize $(1, 8)$-correlated sequences as described in Section 4 for 6 encryption and 6 decryption rounds in a MitM attack. As a result, we do partial encryption for 12 rounds, starting from round 6 and match the left half of state, i.e., s_{19} at 19-th round with the stored value.

In Algorithm 1, we present a generic procedure for recovering the secret key. It takes input as 3 known plaintext-ciphertext pairs encrypted either by

[4] (s_0, s_1) is the plaintext and (s_{25}, s_{26}) is the ciphertext after 25 rounds.

Simon-32/64 or Simeck-32/64 and returns the secret key. The attack procedure is divided into two phases, namely i) *Offline phase* and ii) *Online phase*. The time complexities of both phases are given by $T^{offline}$ and T^{online}, where a subscript (for e.g., T_i^{online}) denotes the time complexity of i-th step of the corresponding phase. In what follows, we present the details of both phases.

Algorithm 1 Generic secret key finding algorithm

Input : $\{(s_0^0, s_1^0), (s_{25}^0, s_{26}^0), (s_0^1, s_1^1), (s_{25}^1, s_{26}^1), (s_0^2, s_1^2), (s_{25}^2, s_{26}^2)\}$
Output : secret key k
CREATE_$\mathbb{D}(s_{25}^0, s_{26}^0)$ ▷ $T^{offline}$
function FIND_SECRET_KEY(*Input*) ▷ T^{online}
    $\mathbb{K}$ = EXTRACT_KEYS$(s_0^0, s_1^0, s_{25}^0, s_{26}^0)$
    for $\bar{k} \in \mathbb{K}$ **do**
        if Encrypt(s_0^1, s_1^1) equals (s_{25}^1, s_{26}^1) **then**
            $\mathbb{K}_1$.append$(\bar{k})$
        end if
    end for
    for $\bar{k} \in \mathbb{K}_1$ **do**
        if Encrypt(s_0^2, s_1^2) equals (s_{25}^2, s_{26}^2) **then**
            $\mathbb{K}_2$.append$(\bar{k})$
        end if
    end for
    return$(\mathbb{K}_2 = \{k\})$
end function

5.1 Offline phase

In this phase, we construct 2^{32} data structures that are used in the online phase to compute the value of s_{19} for all 2^{64} keys without doing any nonlinear operation. For a fixed k_{24} and k_{23}, we denote such a structure by $\mathbb{D}_{(k_{24},k_{23})}$ where each structure has 3 rows, i.e., row_0, row_1 and row_3. By Theorem 1, we observe that for $0 \leq i < 2^{16}$, s_{19} can be computed linearly if s_{23}, $\mathbf{I}_{(k_{24},k_{23})}$ and the index $k_{22} + k_{20} + f_{(a,b,c)}(i)$ are known for $\mathbb{P}(i, 6, \mathbb{K}_{(k_{24},k_{23})})$[5]. Thus, we assign $\mathbb{D}_{(k_{24},k_{23})}.row_0 \leftarrow s_{23}$ and $\mathbb{D}_{(k_{24},k_{23})}.row_1 \leftarrow \mathbf{I}_{(k_{24},k_{23})}$. To find the index value for each key, it is enough to store the values of $\tilde{s_6} + f_{(a,b,c)}(s_{23}) + s_{24} + \tilde{k_4}$ in $\mathbb{D}_{(k_{24},k_{23})}.row_2$, where $\tilde{k} = (k_{24}, k_{23}, 0, \mathbf{I}_{(k_{24},k_{23})}[0] + i)$. This follows because $(\tilde{k}, S_{(\tilde{k},8)}) \in \mathbb{P}(i, 8, \mathbb{K}_{(k_{24},k_{23})})$ and $|\mathrm{CK}(k)| = |\mathrm{CK}(\bar{k})| = 2^{16} - 1$ (using Theorem 1).

The function CREATE_$\mathbb{D}$ in Algorithm 2 construct 2^{32} data strutures for all values of k_{24} and k_{23}, while the function COMPUTE_s_{19} evaluates the value of s_{19} for any key using the stored structures.

Complexities of offline phase. The memory required to store a single structure is $(1 + 2^{16} + 2^{16}) \times 16$ bit. Thus, the total memory is given by $2^{32} \times (1 + 2^{16} + 2^{16}) \times 16 \approx 2^{50}$ bytes. The time complexity in terms of the number of computations of $f_{(a,b,c)}$ is given by

[5] We apply correlated sequences in decryption side.

Algorithm 2 Constructing data structures for 6 decryption rounds

function CREATE_$\mathbb{D}(s_{25}, s_{26})$
  for $k_{24} = 0$ to $2^{16} - 1$ **do**
    for $k_{23} = 0$ to $2^{16} - 1$ **do**
      $\mathbb{D}_{(k_{24},k_{23})}.row_0 \leftarrow s_{23}$ ▷ $T_0^{offline}$
      $\mathbb{D}_{(k_{24},k_{23})}.row_1 \leftarrow \mathbf{I}_{(k_{24},k_{23})}$ ▷ $T_1^{offline}$
      TEMP = []
      for $i = 0$ to $2^{16} - 1$ **do** ▷ $T_2^{offline}$
        $\tilde{k} = (k_{24}, k_{23}, 0, \mathbf{I}_{(k_{24},k_{23})}[0] + i)$
        TEMP.append$(\tilde{s_6} + X + \tilde{k_4})$ ▷ $X = f_{(a,b,c)}(s_{23}) + s_{24}$
      end for
      $\mathbb{D}_{(k_{24},k_{23})}.row_2 \leftarrow$TEMP
    end for
  end for
end function

$T^{offline} = 2^{32}(T_0^{offline} + T_1^{offline} + T_2^{offline}) = 2^{32}(\frac{2}{25} + \frac{N_z}{25} + \frac{N_z}{25})$. From Table 5, we have $N_z = 909$ (resp. 798) for Simon-32/64 (resp. Simeck-32/64). Hence, the respective $T^{offline}$ are $2^{38.19}$ and 2^{38}.

Algorithm 3 Obtaining s_{19} from stored data structures

function COMPUTE_$s_{19}(k_{24}, k_{23}, k_{22}, k_{21}, k_{20}, k_{19})$
  if $k_{22} = 0$ **then**
    $\delta = k_{21}$
  else
    $\delta = k_{21} + \mathbb{D}_{(k_{24},k_{23})}.row_1[k_{22}]$
  end if
  $s_{19} = \mathbb{D}_{(k_{24},k_{23})}.row_0 + \delta + k_{19} + \mathbb{D}_{(k_{24},k_{23})}.row1[k_{22} + k_{20} + \mathbb{D}_{(k_{24},k_{23})}.row_2[\delta]]$
  return(s_{19})
end function

5.2 Online phase

In this phase, we first find the set of keys that maps (s_0^0, s_1^0) to (s_{25}^0, s_{26}^0). The algorithmic details of this step is shown in Algorithm 4. The function EXTRACT_KEYS in Algorithm 4 uses Lemma 1 and Theorem 1 in steps 9-12 and 22-25, respectively, and returns the set ($\mathbb{K}$) of 2^{48} keys[6]. The correct secret key can then be obtained by doing brute force search on $\mathbb{K}$ using another two known plaintext-ciphertext pairs.

[6] We are matching 16-bit state and the key size is 64-bit.

Complexity of online phase. The time complexity is calculated as follows:

$$
\begin{aligned}
T_0^{online} &= \underbrace{\frac{2}{25}}_{\text{2-round encryption}} + \underbrace{\frac{N_z}{25}}_{\#\text{ computations of } f_{(a,b,c)} \text{ to get } \mathbb{P}(0,6,\mathbb{K}_{(k_0,k_1)})} \\
T_1^{online} &= \underbrace{2^{16} \times \frac{12}{25}}_{\text{12-round encryption}} \\
T_2^{online} &= \underbrace{\frac{N_z}{25}}_{\#\text{ computations of } f_{(a,b,c)} \text{ to get } \mathbb{P}(i,8,\mathbb{K}_{(k_0,k_1)})} \\
T_3^{online} &= \underbrace{(2^{32} - 2^{16}) \times \frac{12}{25}}_{\text{12-round encryption}} \\
T^{online} &= 2^{32}(T_0^{online} + T_1^{online} + T_2^{online} + T_3^{online}) + \underbrace{2^{48} + 2^{16}}_{\text{brute force}} \\
&\approx 2^{32}(\frac{2 + 2N_z}{25} + 2^{32} \times \frac{12}{25}) + 2^{48} + 2^{16} \\
&\approx 2^{64} \times \frac{12}{25} \approx 2^{62.94}.
\end{aligned}
$$

The time complexity of complete attack is dominated by $T^{online} \approx 2^{62.94}$.

Remark 4. For the 24-round attack, the data and memory complexities are the same. However, the time complexity is $2^{64} \times \frac{11}{24} \approx 2^{62.87}$.

6 Improving Key Recovery Attack by 2 Rounds

In this section, we show how to improve the key recovery attack presented in previous section by 2 rounds with the same complexities as of the 25-round attack. For a fixed partition $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$, we incorporate the properties of key scheduling algorithms (KSA) and one round differentials and show that $\mathbb{P}(i, 9, \mathbb{K}_{(k_0,k_1)})$ can be computed from $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ by computing $f_{(a,b,c)}$ at most 2^{15} times. As a result, both forward and middle rounds can be extended by one round each, i.e., partial encryption starts from round 7 and matching is done at 20-th round. The results of the following two properties can be obtained directly by the definition of $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$ and key scheduling algorithms. We present the main result of this section in Lemma 2.

Property 3 (Simon KSA and $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$). Let $n = 16$, $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be such that $F(x) = f_{(8,1,2)}(x + \Delta_y) + x + L^{n-1}(x) + L^{n-6}(y) + L^{n-8}(y)$, where $y = \mathbf{I}_{(k_0,k_1)}[x]$ and $\Delta_y = L^{n-3}(y) + L^{n-4}(y)$. Then, $|Img(F(x))| \leq 2^{n-1}$.

Property 4 (Simeck KSA). Let $n \geq 4$, $k_{i+4} = f_{(5,0,1)}(k_{i+1}) + k_i$ and $i \geq 4$. Then for a fixed (k_0, k_1) pair, k_4 is constant for all $2^n \times 2^n$ values of k_2 and k_3.

Algorithm 4 Extracting keys that maps (s_0, s_1) to (s_{25}, s_{26})

function EXTRACT_KEYS(s_0, s_1, s_{25}, s_{26})
  $\mathbb{K} = []$
  for $k_0 = 0$ to $2^{16} - 1$ **do**
    for $k_1 = 0$ to $2^{16} - 1$ **do**
      Obtain $\mathbf{I}_{(k_0,k_1)}$ and s_3 ▷ T_0^{online}
      $k = (k_0, k_1, 0, \mathbf{I}_{(k_0,k_1)}[0])$
      $s_6 = s_4 + k_4$
      $s_7 = s_3 + \mathbf{I}_{(k_0,k_1)}[k_4]$
      for $j = 0$ to $2^{16} - 1$ **do** ▷ Lemma 1
        $k' = (k_0, k_1, j, \mathbf{I}_{(k_0,k_1)}[j])$ ▷ $j = 0 \implies k = k', s'_6 = s_6, s'_7 = s_7$
        $s'_6 = s_6 + k'_4$
        $s'_7 = s_3 + k'_5 + \mathbf{I}_{(k_0,k_1)}[j + k'_4]$
        Encrypt (s'_7, s'_6) for 12 rounds and get s'_{19} ▷ T_1^{online}
        if s'_{19}==compute_$s_{19}(k'_{24}, k'_{23}, k'_{22}, k'_{21}, k'_{20}, k'_{19})$ **then**
          $\mathbb{K}$.append(k')
        end if
      end for
      for z in z-linear segmet sets **do**
        for $x \in CL_z$ and $x \neq 0$ **do**
          for $i = 0$ to $|C_x|-1$ **do**
            $T = L^c(C_x[i]) + L^i(z)$ ▷ T_2^{online}
            for $j = 0$ to $2^{16} - 1$ **do** ▷ Theorem 1
              $\tilde{k} = (k_0, k_1, j, \mathbf{I}_{(k_{24},k_{23})}[j] + C_x[i])$
              $\tilde{s_6} = s_6 + k_4 + \tilde{k_4} + T$
              $\tilde{s_7} = s_3 + C_x[i] + \tilde{k_5} + \mathbf{I}_{(k_0,k_1)}[j + \tilde{k_4} + T]$
              Encrypt $(\tilde{s_7}, \tilde{s_6})$ for 12 rounds and get $\tilde{s}_{19}$ ▷ T_3^{online}
              if $\tilde{s}_{19}$==compute_$s_{19}(\tilde{k}_{24}, \tilde{k}_{23}, \tilde{k}_{22}, \tilde{k}_{21}, \tilde{k}_{20}, \tilde{k}_{19})$ **then**
                $\mathbb{K}$.append($\tilde{k}$)
              end if
            end for
          end for
        end for
      end for
    end for
  end for
  return($\mathbb{K}$)
end function

Property 5 (Differential [16]). Let $n \geq 4$, $\Delta \in \mathbb{F}_2^n$ be fixed. Then,

$$|Img(f_{(a,b,c)}(x) + f_{(a,b,c)}(x + \Delta))| \leq 2^{n-1}.$$

Lemma 2. *Given $n = 16$ and $(a, b, c) = (8, 1, 2)/(5, 0, 1)$. Then, $\forall (k, S_{(k,8)}) \in \mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$, s_7 can take atmost 2^{n-1} values.*

Proof. Consider the value of s_7 in the following cases:

– Case 1 : $(a, b, c) = (8, 1, 2)$

$$\begin{aligned} s_7 &= f_{(8,1,2)}(s_6) + s_5 + k_5 = f_{(8,1,2)}(s_4 + k_4 + f_{(8,1,2)}(i)) + i + k_5 \\ &= f_{(8,1,2)}(X + k_2 + k_4 + f_{(8,1,2)}(i)) + i + k_5, X = f_{(8,1,2)}(s_3) + s_2 \\ &= f_{(8,1,2)}(C_0 + k_2 + (L^{13}(k_3) + L^{12}(k_3)) \\ &\quad + C_1 + k_2 + L^{15}(k_2) + L^{10}(k_3) + L^{8}(k_3) \text{ (Simon KSA)} \end{aligned}$$

Here C_0 and C_1 are constants and given by:

$$\begin{aligned} C_0 &= X + f_{(8,1,2)}(i) + k_0 + k_1 + L^{15}(k_1) + Z_0 \\ C_1 &= i + Z_1 + L^{13}(Z_0) + L^{12}(Z_0) + L^{13}(k_0) + L^{12}(k_0) \\ &= +k_1 + L^{13}(k_1) + L^{11}(k_1) \end{aligned}$$

By Property 3, s_7 can take atmost 2^{n-1} values.

– Case 2 : $(a, b, c) = (5, 0, 1)$

$$\begin{aligned} s_7 &= f_{(5,0,1)}(s_6) + s_5 + k_5 = f_{(5,0,1)}(s_4 + k_4 + f_{(5,0,1)}(i)) + i + k_5 \\ &= f_{(5,0,1)}(X + k_2 + k_4 + f_{(5,0,1)}(i)) + i + k_5, X = f_{(5,0,1)}(s_3) + s_2 \\ &= f_{(5,0,1)}(\Delta + k_2) + C_1 + f_{(5,0,1)}(k_2) \text{ (Property 4)} \end{aligned}$$

Similar to previous case, Δ and C_1 are constants and given by:

$$\begin{aligned} \Delta &= X + f_{(5,0,1)}(i) + k_0 + f_{(5,0,1)}(k_1) + Z_0 \\ C_1 &= i + Z_1 + k_1 \end{aligned}$$

The proof then follows from Property 5. □

From Lemma 2, we note that for each partition $\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)})$, s_7 can take atmost 2^{15} values. In Algorithm 5, we compute the indices for which s_7 values are same. We only evaluate $f_{(a,b,c)}(s_7)$ for the distinct indices only. Hence, 2^{15} computations of $f_{(a,b,c)}$ are needed to obtain $\mathbb{P}(i, 9, \mathbb{K}_{(k_0,k_1)})$. We incorporate Algorithm 5 in the online phase of the attack. The partial encryption then starts from (s_8, s_7) and matching is done at 20-th round.

Algorithm 5 Equal indices algorithm

```
1: function GET_EQUAL_INDEX(ℙ(i, 8, 𝕂_(k0,k1)), I_(k0,k1))
2:     INDICES = []
3:     T = f_(a,b,c)(i)
4:     for j = 0 to 2^16 − 1 do
5:         k = (k0, k1, j, I_(k0,k1)[j])
6:         INDICES.append(I_(k0,k1)[j + k4 + T] + k5)
7:     end for
8:     Return(INDICES)
9: end function
```

Attack complexities. The data and memory complexities are same as 25-round attack. The time complexity is given by:

$$\begin{aligned} T^{online} &= 2^{32}(T_0^{online} + T_1^{online} + T_2^{online} + T_3^{online}) \\ &\quad + \underbrace{2^{48} + 2^{16}}_{\text{brute force}} \\ &\approx 2^{32}(\frac{2+2N_z}{27} + 2^{31} \times \frac{1}{27} + 2^{32} \times \frac{13}{27}) + 2^{48} + 2^{16} \\ &\approx 2^{64} \times \frac{13}{27} \approx 2^{62.94}. \end{aligned}$$

Remark 5. The complexities of 26-round attack are calculated accordingly.

7 Concluding Remarks

In this work, we have proposed correlated sequence attack and demonstrated its application on two lightweight block ciphers Simon-32/64 and Simeck-32/64. As a result, we presented the first 24, 25, 26, 27-round attack on these ciphers with data and memory complexities of 3 and 2^{50} bytes, respectively. The time complexities are $2^{62.87}$(resp. $2^{62.94}$) for 24, 26 (resp. 25, 27)-round attacks.

We observe that correlated sequences play a crucial role in the security evaluation, as improving the length of correlated sequences by 1 extends the number of attacked rounds by 2. Furthermore, we expect that the presented attack on Simon-32/64 and Simeck-32/64 can be improved by uplifting the coset leaders properties from the round function to partitions $\mathbb{P}(i, 9, \mathbb{K}_{(k_0,k_1)})$. It should be noted that our cryptanalytic technique has similar applications to other variants of Simon and Simeck, which we plan to investigate in our future work.

References

[1] NIST lightweight cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/Draft-LWC-Submission-Requirements-April2018.pdf. Accessed 9 May 2018.

[2] Abed, F., List, E., Lucks, S., and Wenzel, J. Differential cryptanalysis of round-reduced simon and speck. In: Cid C., Rechberger C. (eds.), FSE 2014. LNCS, vol. 8540, pp. 525–545. Springer, Heidelberg (2014)

[3] Banik, S., Pandey, S. K., Peyrin, T., Sasaki, Y., Sim, S. M., and Todo, Y. Gift: a small present. In: Fischer W., Homma N. (eds), CHES 2017. LNCS, vol. 10529, pp. 321–345. Springer, Cham (2017)

[4] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. Notes on the design and analysis of simon and speck. Cryptology ePrint Archive, Report 2017/560, 2017. https://eprint.iacr.org/2017/560.pdf.

[5] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. The simon and speck families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org/2013/404.

[6] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., and Sim, S. M. The skinny family of block ciphers and its low-latency variant MANTIS. In: M. Robshaw and J. Katz (eds.), CRYPTO 2016. LNCS, vol. 9815, pp. 123–153. Springer, Heidelberg (2016)

[7] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. In: P. Paillier and I. Verbauwhede (eds.), CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)

[8] Chen, H., and Wang, X. Improved linear hull attack on round-reduced simon with dynamic key-guessing techniques. In: Peyrin T. (eds), FSE 2016. LNCS, vol. 9783, pp. 428–449. Springer, Heidelberg (2016)

[9] Chu, Z., Chen, H., Wang, X., Dong, X., and Li, L. Improved integral attacks on simon32 and simon48 with dynamic key-guessing techniques. Security and Communication Networks 2018. https://doi.org/10.1155/2018/5160237.

[10] De Cannière, C., Dunkelman, O., and Knežević, M. KATAN and KTANTAN — A family of small and efficient hardware-oriented block ciphers. In: Clavier C., Gaj K. (eds.), CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)

[11] Derbez, P., and Fouque, P.-A. Automatic search of meet-in-the-middle and impossible differential attacks. n: Robshaw M., Katz J. (eds), Crypto 2016. LNCS, vol. 9815, pp. 157–184. Springer, Heidelberg (2016)

[12] Diffie, W., and Hellman, M. E. Special feature exhaustive cryptanalysis of the nbs data encryption standard. Computer 10(6), pp. 74–84. (1977)

[13] Fu, K., Sun, L., and Wang, M. New integral attacks on simon. IET Information Security 11(5), pp. 277–286. (2016)

[14] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. The led block cipher. In: B. Preneel and T. Takagi (eds.), CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)

[15] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., Lee, C., Chang, D., Lee, J., Jeong, K., et al. HIGHT: A new block cipher suitable for low-resource device. In: Goubin L., Matsui M (eds.), CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)

[16] Kölbl, S., Leander, G., and Tiessen, T. Observations on the simon block cipher family. In: R. Gennaro and M. Robshaw (eds.), CRYPTO 2015. LNCS, vol. 9215, pp. 258–269. Springer, Heidelberg (2015)

[17] Kölbl, S., and Roy, A. A brief comparison of simon and simeck. In: Bogdanov A. (eds), LightSec 2016. LNCS, vol. 10098, pp. 69–88. Springer, Cham (2016)

[18] Kondo, K., Sasaki, Y., and Iwata, T. On the design rationale of simon block cipher: integral attacks and impossible differential attacks against simon variants. In: Manulis M., Sadeghi AR., Schneider S. (eds), ACNS 2016. LNCS, vol. 9696, pp. 518–536. Springer, Cham (2016)

[19] Liu, Z., Li, Y., and Wang, M. Optimal differential trails in simon-like ciphers. IACR Transactions on Symmetric Cryptology 2017 1, pp. 358–379. (2017) http://dx.doi.org/10.13154/tosc.v2017.i1.358-379

[20] McKay, K., Bassham, L., Sönmez Turan, M., and Mouha, N.: Report on lightweight cryptography (NISTIR8114). (2017)

[21] Qiao, K., Hu, L., and Sun, S. Differential analysis on simeck and simon with dynamic key-guessing techniques. In: Camp O., Furnell S., Mori P. (eds), ICISSP 2016. LNCS, vol. 691, pp. 64–85. Springer, Cham (2016)

[22] Qin, L., Chen, H., and Wang, X. Linear hull attack on round-reduced simeck with dynamic key-guessing techniques. In: Liu J., Steinfeld R. (eds) Information Security and Privacy, ACISP 2016. LNCS, vol. 9723, pp. 409–424. Springer, Cham (2016)

[23] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T. Piccolo: an ultra-lightweight blockcipher. In: Preneel B., Takagi T. (eds), CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011)
[24] Song, L., Hu, L., Ma, B., and Shi, D. Match box meet-in-the-middle attacks on the simon family of block ciphers. In: Eisenbarth T., Öztürk E. (eds), LightSec 2014. LNCS, vol. 8898, pp. 140–151. Springer, Cham (2014)
[25] Sun, L., Fu, K., and Wang, M. Improved zero-correlation cryptanalysis on simon. In: Lin D., Wang X., Yung M. (eds), Inscrypt 2015. LNCS, vol. 9589, pp. 125–143. Springer, Cham (2014)
[26] Todo, Y., and Morii, M. Bit-based division property and application to simon family. In: Peyrin T. (eds), FSE 2016. LNCS, vol. 9783, pp. 357-377. Springer, Heidelberg (2016)
[27] Wang, N., Wang, X., Jia, K., and Zhao, J. Differential attacks on reduced simon versions with dynamic key-guessing techniques. Cryptology ePrint Archive, Report 2014/448, 2014. https://eprint.iacr.org/2014/448.pdf.
[28] Wang, Q., Liu, Z., Varici, K., Sasaki, Y., Rijmen, V., and Todo, Y. Cryptanalysis of reduced-round simon32 and simon48. In: Meier W., Mukhopadhyay D. (eds), Indocrypt 2014. LNCS, vol. 8885, pp. 143–160. Springer, Cham (2014)
[29] Xiang, Z., Zhang, W., Bao, Z., and Lin, D. Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: J. H. Cheon and T. Takagi (Eds.), ASIACRYPT 2016. LNCS, vol. 10031, pp. 648–678. Springer, Heidelberg (2016)
[30] Yang, G., Zhu, B., Suder, V., Aagaard, M. D., and Gong, G. The simeck family of lightweight block ciphers. In: T. Güneysu and H. Handschuh (Eds.), CHES 2015. LNCS, vol. 9293, pp. 307–329. Springer, Heidelberg (2015)
[31] Zhang, K., Guan, J., Hu, B., and Lin, D. Integral cryptanalysis on simeck. ICIST 2016. IEEE, pp. 216–222. (2016)
[32] Zhang, K., Guan, J., Hu, B., and Lin, D. Security evaluation on simeck against zero-correlation linear cryptanalysis. IET Information Security 12(1), pp. 87–93. (2018)