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 $\sigma$ ($0 \leq \sigma < t$) computations of the non-linear function. We call these sequences ($\sigma, t$)-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, achieve 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 \(\sigma\) \((0 \leq \sigma < t)\) computations of the non-linear function. We call these sequences \((\sigma, t)\)-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

<table>
<thead>
<tr>
<th>Attack</th>
<th>Cipher</th>
<th># attacked rounds / 32</th>
<th>Data</th>
<th>Memory (Bytes)</th>
<th>Time</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential</td>
<td>Simon-32/64 [27]</td>
<td>21</td>
<td>2^{31}</td>
<td>-</td>
<td>2^{55.25}</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>Simon-32/64 [21]</td>
<td>22</td>
<td>2^{32}</td>
<td>-</td>
<td>2^{55.76}</td>
<td>0.315</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [17]</td>
<td>19</td>
<td>2^{31}</td>
<td>2^{10}</td>
<td>2^{90}</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [21]</td>
<td>22</td>
<td>2^{32}</td>
<td>-</td>
<td>2^{37.9}</td>
<td>0.417</td>
</tr>
<tr>
<td>Linear</td>
<td>Simon-32/64 [8]</td>
<td>23</td>
<td>2^{31.19}</td>
<td>-</td>
<td>2^{51.84}</td>
<td>0.277</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [22]</td>
<td>23</td>
<td>2^{31.91}</td>
<td>-</td>
<td>2^{51.78}</td>
<td>0.456</td>
</tr>
<tr>
<td>Integral</td>
<td>Simon-32/64 [28]</td>
<td>21</td>
<td>2^{31}</td>
<td>2^{34}</td>
<td>2^{83}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Simon-32/64 [13]</td>
<td>22</td>
<td>2^{31}</td>
<td>2^{55.8}</td>
<td>2^{83}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Simon-32/64 [9]</td>
<td>24</td>
<td>2^{32}</td>
<td>2^{33.64}</td>
<td>2^{83}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [31]</td>
<td>21</td>
<td>2^{31}</td>
<td>2^{46.22}</td>
<td>2^{83}</td>
<td>1</td>
</tr>
<tr>
<td>Impossible Differential</td>
<td>Simon-32/64 [11]</td>
<td>20</td>
<td>2^{32}</td>
<td>2^{45.5}</td>
<td>2^{82.8}</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [30]</td>
<td>20</td>
<td>2^{32}</td>
<td>2^{38}</td>
<td>2^{82.5}</td>
<td>-</td>
</tr>
<tr>
<td>Zero correlation</td>
<td>Simon-32/64 [25]</td>
<td>21</td>
<td>2^{32}</td>
<td>2^{31}</td>
<td>2^{59.4}</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64 [32]</td>
<td>21</td>
<td>2^{32}</td>
<td>2^{47.67}</td>
<td>2^{58.78}</td>
<td>-</td>
</tr>
<tr>
<td>Meet-in-the-middle</td>
<td>Simon-32/64 [24]</td>
<td>18</td>
<td>8</td>
<td>2^{32}</td>
<td>2^{52.57}</td>
<td>1</td>
</tr>
<tr>
<td>Correlated sequence attack</td>
<td>Simon-32/64</td>
<td>24</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.87}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.94}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.88}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.94}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Simeck-32/64</td>
<td>24</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.87}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.94}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.88}</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27</td>
<td>3</td>
<td>2^{20}</td>
<td>2^{62.94}</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>bitwise XOR</td>
</tr>
<tr>
<td>&amp;</td>
<td>bitwise AND</td>
</tr>
<tr>
<td>n</td>
<td>wordsize</td>
</tr>
<tr>
<td>K</td>
<td>key space</td>
</tr>
<tr>
<td>$F_2^n$</td>
<td>$n$ dimensional vector space over $F_2$</td>
</tr>
<tr>
<td>$L^i$</td>
<td>left cyclic shift operator, i.e., for $x \in F_2^n$, $L^i(x) = (x_i, x_{i+1}, \ldots, x_{n-1}, x_0, x_1, \ldots, x_{i-1})$</td>
</tr>
<tr>
<td>$C_s$</td>
<td>coset modulo $2^n - 1$, i.e., $C_s = {s, 2s, \ldots, 2^{n-1}s}$ where $n_s$ is the smallest number such that $s \equiv 2^{n_s}s \mod 2^n - 1$, and $s$ is the smallest number in $C_s$ and denotes the coset leader</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
</tr>
<tr>
<td>$A[i]$</td>
<td>$i$-th element of $A$</td>
</tr>
<tr>
<td>$\text{Img}(f)$</td>
<td>Image set of $f$</td>
</tr>
</tbody>
</table>

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 $k_i \in F_2^n$ 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 \( m = 4 \) and the subkeys for \( i \geq 4 \) are calculated as follows.

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

**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 \( 0 \leq i < r \), \( s_{i+1} = \text{rf}(s_i, k_i) \) where \( \text{rf} \) denotes the round function, and is generally a composition of two functions, namely i) a linear function \( \chi \) and ii) a non-linear function \( \rho \).

![Fig. 2: Generic diagram of a block cipher](image)

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 \( s_{i+1} = \text{rf}(s_i, k_i) \) for \( 0 \leq i < t - 1 \).

From Definition 1, it is clear that we need to compute \( \text{rf} \) \( t \) times to obtain \( S_{(k,t)} \). This implies that \( \rho \) is computed \( t \) times in total. Thus, to obtain another sequence \( S_{(k',t)} \) of same length \( t \), the worst case is to compute \( \rho \) 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 \( \rho \) at most \( t \) 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 \( \rho(x) = L(x) \land x + L^2(x) \) 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 \( s'_6 = s_6 + k_4 + k'_4 + \rho(1) \).

We now define the correlated sequences in Definition 2.

**Definition 2** ((\( \sigma, t \))-correlated sequences). Given \( S(k, t) \) and \( 0 \leq \sigma < t \). We say \( S(k, t) \) and \( S(k', t) \) are \((\sigma, t)\)-correlated sequences if \( S(k', t) \) can be obtained from \( S(k, t) \) by computing the non-linear function \( \rho \) exactly \( \sigma \) 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 \( \text{CK}(k) = \{ k' \mid S(k', t) \text{ and } S(k, t) \text{ are } (0, t)\text{-correlated sequences} \} \).

For example, in Table 3, for each row \( S(k, t) \) and \( S(k', t) \) are \((1, 7)\) correlated sequences, \(|\text{CK}((0, 0, 0, 0))| = 15 \) (gray colored rows) and \(|\text{CK}((0, 0, 0, 1))| = 15 \) (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 \( \rho \).

**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 \((\sigma, t_1)\)-correlated sequences and their corresponding \( \text{CK}(\cdot) \) for \( t_1 \) rounds. Next, starting with \( s_r \), we follow the same approach. We then do partial encryption.
Table 3: Keyed sequences

<table>
<thead>
<tr>
<th>$k_0$ $k_1$ $k_2$ $k_3$ $k_4$</th>
<th>$s_0$ $s_1$ $s_2$ $s_3$ $s_4$ $s_5$</th>
<th>$k'_0$ $k'_1$ $k'_2$ $k'_3$ $k'_4$</th>
<th>$s'_0$ $s'_1$ $s'_2$ $s'_3$ $s'_4$ $s'_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 13</td>
<td>0 0 0 0 0 13</td>
<td>0 0 0 1 14</td>
<td>0 0 0 0 0 1 10</td>
</tr>
<tr>
<td>0 0 1 4 1</td>
<td>0 0 0 0 1 0 0</td>
<td>0 0 1 5 2</td>
<td>0 0 0 0 1 1 7</td>
</tr>
<tr>
<td>0 0 2 8 4</td>
<td>0 0 0 0 2 0 6</td>
<td>0 0 2 9 7</td>
<td>0 0 0 0 2 1 1</td>
</tr>
<tr>
<td>0 0 3 14 14</td>
<td>0 0 0 0 3 0 13</td>
<td>0 0 3 15 13</td>
<td>0 0 0 0 3 1 10</td>
</tr>
<tr>
<td>0 0 4 1 14</td>
<td>0 0 0 0 4 0 10</td>
<td>0 0 4 0 13</td>
<td>0 0 0 0 4 1 13</td>
</tr>
<tr>
<td>0 0 5 5 2</td>
<td>0 0 0 0 5 0 7</td>
<td>0 0 5 4 1</td>
<td>0 0 0 0 5 1 0</td>
</tr>
<tr>
<td>0 0 6 13 11</td>
<td>0 0 0 0 6 0 13</td>
<td>0 0 6 12 8</td>
<td>0 0 0 0 6 1 10</td>
</tr>
<tr>
<td>0 0 7 11 1</td>
<td>0 0 0 0 7 0 6</td>
<td>0 0 7 10 2</td>
<td>0 0 0 0 7 1 1</td>
</tr>
<tr>
<td>0 0 8 2 11</td>
<td>0 0 0 0 8 0 3</td>
<td>0 0 8 3 8</td>
<td>0 0 0 0 8 1 4</td>
</tr>
<tr>
<td>0 0 9 7 4</td>
<td>0 0 0 0 9 0 13</td>
<td>0 0 9 6 7</td>
<td>0 0 0 0 9 1 10</td>
</tr>
<tr>
<td>0 0 10 10 2</td>
<td>0 0 0 0 10 0 8</td>
<td>0 0 10 11 1</td>
<td>0 0 0 0 10 1 15</td>
</tr>
<tr>
<td>0 0 11 13 11</td>
<td>0 0 0 0 11 0 0</td>
<td>0 0 11 12 8</td>
<td>0 0 0 0 11 1 7</td>
</tr>
<tr>
<td>0 0 12 11 1</td>
<td>0 0 0 0 12 0 13</td>
<td>0 0 12 10 2</td>
<td>0 0 0 0 12 1 10</td>
</tr>
<tr>
<td>0 0 13 14 14</td>
<td>0 0 0 0 13 0 3</td>
<td>0 0 13 15 13</td>
<td>0 0 0 0 13 1 4</td>
</tr>
<tr>
<td>0 0 14 7 4</td>
<td>0 0 0 0 14 0 10</td>
<td>0 0 14 6 7</td>
<td>0 0 0 0 14 1 13</td>
</tr>
<tr>
<td>0 0 15 0 13</td>
<td>0 0 0 0 15 0 2</td>
<td>0 0 15 1 14</td>
<td>0 0 0 0 15 1 5</td>
</tr>
</tbody>
</table>

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

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 $\left\lceil \frac{mn}{n} \right\rceil - 1$ 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 \( s = \overline{011 \ldots 1}_n \).

Property 2. Let \( s = \overline{0101 \ldots 01}_n \) and \( a, b \) are not both simultaneously even or odd, then

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

Properties 1 and 2 imply that we need to compute the values of \( f_{(a,b,c)} \) for \( 2^n - 1 \) coset leaders only. However, as \( f_{(a,b,c)} \) is quadratic and the only linear term involved in it is \( L^c(\cdot) \), we have \( f_{(a,b,c)}(x) = L^c(x) + z \) for all \( x \in \mathbb{F}_2^n \) and some constant \( z \in \mathbb{F}_2^2 \). 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.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f_{(8,1,2)}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>( L^2(3) + 2 = 14 )</td>
</tr>
<tr>
<td>6</td>
<td>( L^2(6) + L(2) = 28 )</td>
</tr>
<tr>
<td>12</td>
<td>( L^2(12) + L^2(2) = 56 )</td>
</tr>
<tr>
<td>24</td>
<td>( L^2(24) + L^3(2) = 112 )</td>
</tr>
<tr>
<td>48</td>
<td>( L^2(28) + L^4(2) = 224 )</td>
</tr>
<tr>
<td>96</td>
<td>( L^2(96) + L^3(2) = 193 )</td>
</tr>
<tr>
<td>192</td>
<td>( L^2(192) + L^5(2) = 131 )</td>
</tr>
<tr>
<td>129</td>
<td>( L^2(129) + L^7(2) = 7 )</td>
</tr>
</tbody>
</table>
Table 4: $z$-linear segment sets for $n = 8$ and $(a, b, c) = (8, 1, 2)$

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

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

<table>
<thead>
<tr>
<th>$n$</th>
<th># coset leaders</th>
<th>$N_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$(a, b, c)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(8, 1, 2)$</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>108</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>352</td>
<td>119</td>
</tr>
<tr>
<td>14</td>
<td>1182</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td>4116</td>
<td>909</td>
</tr>
</tbody>
</table>

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

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

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 \( T_{\text{step}} \). 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 \( f_{(a,b,c)}(X + k_2) = k_3 + s_4 \) 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, \( T_{\text{step1}} = 2 + N_z \).

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

\[
s_4 + s_6 = s_4 + f_{(a,b,c)}(s_5) + s_4 + k_4 = s_4 + 0 + s_4 + k_4 \Rightarrow s_6 = s_4 + k_4.
\]

Thus, \( T_{\text{step2}} = 0 \).

**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, \( f_{(a,b,c)}(s_4) = f_{(a,b,c)}(s_4') = k'_3 + s_3 \) (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, \( T_{\text{step3}} = 0 \).

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 \( I_{(k_0, k_1)} = \{ k_3 \mid ((k_0, k_1, k_2, k_3), S_{((k_0, k_1, k_3), k_1)}(0)) \in \mathbb{P}(0, 6, \mathbb{K}_{(k_0, k_1)})) \}^3 \) and \( k = (k_0, k_1, 0, I_{(k_0, k_1)}[0]) \). Let \( 1 \leq k_2 < 2^n \) and \( k' = (k_0, k_1, k_2, I_{(k_0, k_1)}[k_2]) \) 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. \( s_2' = s_3 + k'_5 + I_{(k_0, k_1)}[k_2 + k'_4] \).
5. \( s_7 = s_3 + k_5 + I_{(k_0, k_1)}[k_4] \).
6. \( |\mathbb{C}(k)| = 2^n - 1 \).

**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 \( s_4' = f_{(a,b,c)}(s_3') + s_2' + k_2 = f_{(a,b,c)}(s_3) + s_2 + k_2 \Rightarrow s_4' = X + k_2 \).
3. Consider the following relation \( s_6' + s_6 \).

\[
s_6' + s_6 = f_{(a,b,c)}(s_5') + s_4' + k'_4 + f_{(a,b,c)}(s_5) + s_4 + k_4
\]

\[
= s_4' + k'_4 + s_4 + k_4
\]

\[
= X + k_2 + k'_4 + X + k_4 \Rightarrow s_6' = s_6 + k_4 + k'_4 + k_2.
\]

\[^3\] \( |I_{(k_0, k_1)}| = 2^n \) (see Remark 3). Thus, \( I_{(k_0, k_1)} \) can have multiple values of \( k_3 \).
4. Note that by Step 3 of construction of $P(0, 8, K_{(k_0, k_1)})$, we have $s'_7 = s_3 + k'_5 + k'_3$. Hence, we need to find the index $j$ such that $I_{(k_0, k_1)}[j] = k''_3$. 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 $s'_4$ can be computed linearly.

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

1. $S_{(k, 5)} = S_{(\bar{k}, 5)}$
2. $(\bar{k}, S_{(\bar{k}, 6)}) \in P(i, 6, K_{(k_0, k_1)})$
3. $s_6 = s_6 + k_4 + \bar{k}_4 + f_{(a, b, c)}(i)$
4. $s_7 = s_3 + i + \bar{k}_3 + I_{(k_0, k_1)}[\bar{k}_2 + \bar{k}_4 + f_{(a, b, c)}(i)]$
5. $|CK(k)| = |CK(\bar{k})| = 2^n - 1$

**Proof.** The proof of 1, 2, 3 and 4 is similar to Lemma 1. For part 5, note that for $1 \leq j < 2^n$, $(k_0, k_1, j, I_{(k_0, k_1)}[j]) \in CK(k) \iff (k_0, k_1, j, I_{(k_0, k_1)}[\bar{j}] + i) \in CK(\bar{k})$. This follows because $s_5 + s_5 = k_3 + \bar{k}_3 \implies k_3 + \bar{k}_3 = i$. Thus, $|CK(k)| = |CK(\bar{k})| = 2^n - 1$.

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

<table>
<thead>
<tr>
<th>Approach</th>
<th># computations of $f_{(a, b, c)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive</td>
<td>$2^{32} \times 6$</td>
</tr>
<tr>
<td>Theorem 1 and z-linear segment sets</td>
<td>$(2 + 1818)$</td>
</tr>
<tr>
<td></td>
<td>$(2 + 1596)$</td>
</tr>
</tbody>
</table>

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 \( T_{\text{offline}} \) and \( T_{\text{online}} \), where a subscript (for e.g., \( T_{i}^{\text{online}} \)) 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

1: Input : \( \{(s_0^i, s_1^i), (s_{25}^i, s_{26}^i), (s_0^i, s_1^i), (s_{25}^i, s_{26}^i), (s_2^i, s_4^i), (s_{25}^i, s_{26}^i)\} \)
2: Output : secret key \( k \)
3: function \( \text{CREATE}_{D}(s_{25}^0, s_{26}^0) \) \( \triangleright T_{\text{offline}} \)
4: function \( \text{FIND}_{\text{SECRET\_KEY}}(\text{Input}) \) \( \triangleright T_{\text{online}} \)
5: \( \mathcal{K} = \text{EXTRACT}_{\text{KEY}}(s_0^0, s_1^0, s_{25}^0, s_{26}^0) \)
6: for \( \bar{k} \in \mathcal{K} \) do
7: \( \text{if} \) Encrypt\((s_0^i, s_1^i)\) equals \((s_{25}^i, s_{26}^i)\) \( \text{then} \)
8: \( \mathcal{K}_1.\text{append}(\bar{k}) \)
9: end if
10: end for
11: for \( \bar{k} \in \mathcal{K}_1 \) do
12: \( \text{if} \) Encrypt\((s_0^i, s_1^i)\) equals \((s_{25}^i, s_{26}^i)\) \( \text{then} \)
13: \( \mathcal{K}_2.\text{append}(\bar{k}) \)
14: end if
15: end for
16: return\((\mathcal{K}_2 = \{\bar{k}\})\)
17: 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., \( \text{row}_0, \text{row}_1 \) and \( \text{row}_3 \). By Theorem 1, we observe that for \( 0 \leq i < 2^{16} \), \( s_{19} \) can be computed linearly if \( s_{23}, \mathbb{I}_{(k_{24},k_{23})} \) and the index \( k_{22} + k_{20} + f_{(a,b,c)}(i) \) are known for \( \mathbb{P}(i,6,\mathbb{K}_{(k_{24},k_{23})}) \).

Thus, we assign \( \mathbb{D}_{(k_{24},k_{23})}.\text{row}_0 \leftarrow s_{23} \) and \( \mathbb{D}_{(k_{24},k_{23})}.\text{row}_1 \leftarrow \mathbb{I}_{(k_{24},k_{23})} \). To find the index value for each key, it is enough to store the values of \( s_6 + f_{(a,b,c)}(s_{23}) + s_{24} + \bar{k}_4 \) in \( \mathbb{D}_{(k_{24},k_{23})}.\text{row}_2 \), where \( \bar{k} \ = (k_{24},k_{23},0,\mathbb{I}_{(k_{24},k_{23})}[0] + i) \). This follows because \( (\bar{k},S_{(k_8,\mathcal{K})}) \in \mathbb{P}(i,8,\mathbb{K}_{(k_{24},k_{23})}) \) and \( |\text{CK}(\bar{k})| = |\text{CK}(\bar{k})| = 2^{16} - 1 \) (using Theorem 1).

The function \( \text{CREATE}_{D} \) in Algorithm 2 construct \( 2^{32} \) data structures for all values of \( k_{24} \) and \( k_{23} \), while the function \( \text{COMPUTE}_{s_{19}} \) 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

1: function CREATE_D(s_{25}, s_{26})
2:     for k_{24} = 0 to 2^{16} - 1 do
3:         for k_{23} = 0 to 2^{16} - 1 do
4:             D(k_{24}, k_{23}).row_0 ← s_{23} \quad \triangleright T_{0}^{ offline} 
5:             D(k_{24}, k_{23}).row_1 ← I(k_{24}, k_{23}) \quad \triangleright T_{1}^{ offline} 
6:             TEMP = []
7:         for i = 0 to 2^{16} - 1 do
8:             k = (k_{24}, k_{23}, 0, I(k_{24}, k_{23})[0] + i) \quad \triangleright T_{2}^{ offline} 
9:             TEMP.append(s_{6} + X + \delta + k_{19} + D(k_{24}, k_{23}).row_1[k_{22} + k_{20} + D(k_{24}, k_{23}).row_2[\delta]])
10:        end for
11:     D(k_{24}, k_{23}).row_2 ← TEMP
12: end for
13: 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

1: function COMPUTE_s_{19}(k_{24}, k_{23}, k_{22}, k_{21}, k_{20}, k_{19})
2:     if k_{22} = 0 then
3:         \delta = k_{21}
4:     else
5:         \delta = k_{21} + D(k_{24}, k_{23}).row_1[k_{22}]
6:     end if
7:     s_{19} = D(k_{24}, k_{23}).row_0 + \delta + k_{19} + D(k_{24}, k_{23}).row_1[k_{22} + k_{20} + D(k_{24}, k_{23}).row_2[\delta]]
8:     return(s_{19})
9: end function

5.2 Online phase

In this phase, we first find the set of keys that maps (s_{0}, s_{1}) to (s_{25}, s_{26}). 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 (K) of 2^{48} keys\footnote{We are matching 16-bit state and the key size is 64-bit.}. The correct secret key can then be obtained by doing brute force search on \text{K} using another two known plaintext-ciphertext pairs.
Complexity of online phase. The time complexity is calculated as follows:

\[ T_{\text{online}}^0 = \frac{2}{25} \] 
\[ T_{\text{online}}^1 = \frac{12}{25} \] 
\[ T_{\text{online}}^2 = \frac{N_z}{25} \] 
\[ T_{\text{online}}^3 = (2^{32} - 2^{16}) \times \frac{12}{25} \]

The time complexity of complete attack is dominated by \( T_{\text{online}} \approx 2^{62.94} \).

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 \( \mathbb{P}(i, 8, K_{(k_0, k_1)}) \), we incorporate the properties of key scheduling algorithms (KSA) and one round differentials and show that \( \mathbb{P}(i, 9, K_{(k_0, k_1)}) \) can be computed from \( \mathbb{P}(i, 8, K_{(k_0, k_1)}) \) 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 \( \mathbb{P}(i, 8, K_{(k_0, k_1)}) \) and key scheduling algorithms. We present the main result of this section in Lemma 2.

Property 3 (Simon KSA and \( \mathbb{P}(i, 8, K_{(k_0, k_1)}) \)). Let \( n = 16 \), \( F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \) be such that \( F(x) = f_{(8,1,2)}(x + \Delta_y) + x + L^{n-1}(x) + L^{n-0}(y) + L^{n-8}(y) \), where \( y = I_{(k_0, k_1)}[x] \) and \( \Delta_y = L^{n-3}(y) + L^{n-4}(y) \). Then, \( |\text{Img}(F(x))| \leq 2^{n-1} \).

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 \((s_{25}, s_{26})\)

1: function EXTRACT KEYS\((s_0, s_1, s_{25}, s_{26})\)
2: \(K = []\)
3: for \(k_0 = 0\) to \(2^{16} - 1\) do
4:   for \(k_1 = 0\) to \(2^{16} - 1\) do \(\triangleright T_{0}^{online}\)
5:     Obtain \(I_{(k_0, k_1)}\) and \(s_3\)
6:     \(k = (k_0, k_1, 0, I_{(k_0, k_1)}[0])\)
7:     \(s_6 = s_4 + k_4\)
8:     \(s_7 = s_3 + I_{(k_0, k_1)}[k_4]\)
9:   for \(j = 0\) to \(2^{16} - 1\) do \(\triangleright \text{Lemma 1}\)
10:     \(k' = (k_0, k_1, j, I_{(k_0, k_1)}[j])\)
11:     \(s_6' = s_6 + k_4'\)
12:     \(s_7' = s_3 + k_4' + I_{(k_0, k_1)}[j + k_4]\)
13:     Encrypt \((s_7', s_6')\) for 12 rounds and get \(s_{19}'\)
14:     if \(s_{19}' = \text{compute}_{s_{19}}(k_{24}, k_23, k_22, k_21, k_20, k_{19})\) then
15:         \(K.append(k')\)
16:     end if
17: end for
18: for \(z\) in \(z\)-linear segment sets do
19:   for \(x \in CL_z\) and \(x \neq 0\) do
20:     for \(i = 0\) to \(|C_z| - 1\) do \(\triangleright T^{online}\)
21:       \(T = L^i(C_z[i]) + L^j(z)\)
22:     for \(j = 0\) to \(2^{16} - 1\) do \(\triangleright \text{Theorem 1}\)
23:       \(k = (k_0, k_1, j, I_{(k_0, k_1)}[j] + C_z[i])\)
24:       \(s_6 = s_6 + k_4 + k_4 + T\)
25:       \(s_7 = s_3 + C_z[i] + k_5 + I_{(k_0, k_1)}[j + k_4 + T]\)
26:       Encrypt \((s_7, s_6)\) for 12 rounds and get \(s_{19}\)
27:     if \(s_{19} = \text{compute}_{s_{19}}(k_{24}, k_23, k_22, k_21, k_20, k_{19})\) then
28:         \(K.append(k)\)
29:     end if
30: end for
31: end for
32: end for
33: end for
34: end for
35: end for
36: return(\(K\))
37: end function

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

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

Lemma 2. Given \(n = 16\) and \((a, b, c) = (8, 1, 2)/(5, 0, 1)\). Then, \(\forall (k, S_{(k, 8)}) \in \mathbb{F}(i, 8, K_{(k, 8)})\), \(s_7\) can take at most \(2^{n-1}\) values.

Proof. Consider the value of \(s_7\) in the following cases:
Case 1: \((a, b, c) = (8, 1, 2)\)

\[
s_7 = f_{(8,1,2)}(s_6) + s_5 + k_5 = f_{(8,1,2)}(s_4 + k_4 + f_{(8,1,2)}(i)) + i + k_5
\]

\[
= f_{(8,1,2)}(X + k_2 + k_4 + f_{(8,1,2)}(i)) + i + k_5, X = f_{(8,1,2)}(s_3) + s_2
\]

\[
= f_{(8,1,2)}(C_0 + k_2 + (L_{13}(k_3) + L_{12}(k_3)) + C_1 + k_2 + L_{15}(k_2) + L_{10}(k_3) + L_8(k_3) \text{ (Simon KSA)}
\]

Here \(C_0\) and \(C_1\) are constants and given by:

\[
C_0 = X + f_{(8,1,2)}(i) + k_0 + k_1 + L_{15}(k_1) + Z_0
\]

\[
C_1 = i + Z_1 + L_{13}(Z_0) + L_{12}(Z_0) + L_{13}(k_0) + L_{12}(k_0) = +k_1 + L_{13}(k_1) + L_{11}(k_1)
\]

By Property 3, \(s_7\) can take at most \(2^{n-1}\) values.

Case 2: \((a, b, c) = (5, 0, 1)\)

\[
s_7 = f_{(5,0,1)}(s_6) + s_5 + k_5 = f_{(5,0,1)}(s_4 + k_4 + f_{(5,0,1)}(i)) + i + k_5
\]

\[
= f_{(5,0,1)}(X + k_2 + k_4 + f_{(5,0,1)}(i)) + i + k_5, X = f_{(5,0,1)}(s_3) + s_2
\]

\[
= f_{(5,0,1)}(\Delta + k_2) + C_1 + f_{(5,0,1)}(k_2) \text{ (Property 4)}
\]

Similar to previous case, \(\Delta\) and \(C_1\) are constants and given by:

\[
\Delta = X + f_{(5,0,1)}(i) + k_0 + f_{(5,0,1)}(k_1) + Z_0
\]

\[
C_1 = i + Z_1 + k_1
\]

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 at most \(2^{15}\) 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(\(\mathbb{P}(i, 8, \mathbb{K}_{(k_0,k_1)}), \mathbb{I}_{(k_0,k_1)}\))
2: INDICES = []
3: \(T = f_{(a,b,c)}(i)\)
4: for \(j = 0\) to \(2^{16} - 1\) do
5: \(k = (k_0, k_1, j, \mathbb{I}_{(k_0,k_1)}[j])\)
6: INDICES.append(\(\mathbb{I}_{(k_0,k_1)}[j + k_4 + T] + k_5\))
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:

\[
T_{\text{online}} = 2^{32}(T_0^{\text{online}} + T_1^{\text{online}} + T_2^{\text{online}} + T_3^{\text{online}}) + 2^{48} + 2^{16} \\text{brute force}
\]

\[
\approx 2^{32}(\frac{2 + 2N_z}{27} + 2^{31} \times \frac{1}{27} + 2^{32} \times \frac{13}{27}) + 2^{48} + 2^{16}
\]

\[
\approx 2^{64} \times \frac{13}{27} \approx 2^{62.94}.
\]

**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 $P(i, 9, K_{(k_0, k_1)})$. 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


