Abstract

A5/1 is a shift register based stream cipher which provides privacy for the GSM system. In this paper, we analyse the loading of the secret key and IV during the initialisation process of A5/1. We demonstrate the existence of weak key-IV pairs in the A5/1 cipher due to this loading process; these weak key-IV pairs may generate one, two or three registers containing all-zero values, which may lead in turn to weak keystream sequences. In the case where two or three registers contain only zeros, we describe a distinguisher which leads to a complete decryption of the affected messages.

keywords: A5/1, initialisation process, loading phase, weak key-IV pairs, ciphertext only attack, stream cipher.

1 Introduction

The privacy of mobile telephone communications is protected by the A5/1 stream cipher. The approximate design of A5/1 was leaked in 1994 and the exact design revealed in 1999, when it was reverse engineered by Briceno (Briceno et al. 1999). A5/1 is a bit-based stream cipher that takes a 64-bit secret key and a 22-bit initialisation vector (IV), the frame number, as inputs into the 64-bit internal state. The internal state of this cipher is contained in three linear feedback shift registers (LFSRs).

Most recent stream cipher proposals require an initialisation process as an essential part of the cipher’s specification. During the initialisation process, a secret key, k, and an IV, v, are loaded into the internal state. This loaded state is then processed to diffuse k and v across the internal state before keystream generation occurs. A good initialisation process should ensure that keystreams formed from the same key but different IVs do not reveal information about the secret key and can not be used to facilitate any attack.

It is sometimes possible to find key-IV pairs that result in one or more registers containing only zeros at the end of the loading and diffusion phases of the initialisation process (Zhang & Wang 2009). We refer to key-IV pairs that generate one or more registers containing all-zeros at the end of the loading phase as weak key-IV pairs.

Keywords: A5/1, initialisation process, loading phase, weak key-IV pairs, ciphertext only attack, stream cipher.

In A5/1, the key and IV are loaded by clocking the registers and combining successive key and IV bits with the usual register feedback; thus, the registers’ feedback mechanism is not autonomous during this process. For some non-zero key-IV combinations, this results in one or more of the registers containing only zeroes at the end of the loading phase.

In this paper, we establish conditions under which such weak key-IV pairs exist for the A5/1 cipher. We show that if a weak key-IV pair results in two or three of the registers containing all-zero values, then the generated keystream is easy to distinguish and the cipher is vulnerable to attack. Under these circumstances, a ciphertext-only attack can be used to reveal the secret key and hence to decrypt the entire conversation.

This paper is organized as follows. Section 2 provides a full description of the A5/1 stream cipher including the non-autonomous feedback mechanism during the loading phase of the initialisation process. Section 3 provides a matrix representation for the operation of an autonomous LFSR and adapts this technique to a non-autonomous LFSR as used in the loading phase of A5/1. This representation is used in Section 4 to determine the conditions under which one, two and three registers contain all-zero values. Section 5 presents a possible ciphertext-only attack procedure to reveal the secret key in the cases where two or three registers contain only zeros. Section 6 concludes the paper.

2 Description of A5/1

A5/1 (Biryukov et al. 2001, Golić 1997) is a bit based stream cipher which uses three binary LFSRs, denoted A, B and C, with lengths of 19, 22 and 23 bits respectively, giving a state size of 64 bits. Each LFSR has a primitive feedback polynomial. Let S denote the internal state of A5/1 and let $S_A$, $S_B$ and $S_C$ denote the internal states for the component registers A, B and C respectively. Let $s_{i,t}$ denote the contents of stage i of register A at time t, for $0 \leq i \leq 18$, $s_{h,t}$ denote the contents of stage i of register B at time t, for $0 \leq i \leq 21$, $s_{b,t}$ denote the contents of stage i of register C at time t, for $0 \leq i \leq 22$ respectively.

A secret key of 64 bits is used for each conversation and a 22-bit frame number is used as the IV for each frame. Let $k_i$ denote the secret key for $0 \leq i \leq 63$ and $v_i$ denote the IV for $0 \leq i \leq 21$. The three registers are regularly clocked during loading of the key and IV (frame number), while a majority clocking mechanism is used for the diffusion phase and for keystream generation. The use of majority clocking implicitly introduces nonlinearity to the keystream sequence. This is the only nonlinear operation performed.
To implement the majority clocking scheme, each register has a clocking tap: stages \( s_{s,t}^8 \), \( s_{b,t}^{10} \) and \( s_{c,t}^{10} \). The contents of these stages determine which registers will be clocked at the next iteration: those registers for which the clock control bits agree with the majority value are clocked. For example, if \( s_{s,t}^8 = 0 \), \( s_{b,t}^{10} = 1 \) and \( s_{c,t}^{10} = 0 \), then the majority value is 0 and registers \( A \) and \( C \) are clocked. Thus, either two or three registers are clocked at each step. Figure 1 shows a pictorial diagram of the A5/1 stream cipher.

![Figure 1: A5/1 Stream cipher.](image)

### 2.1 Initialisation process

The initialisation process is conducted in two phases as follows.

#### 2.1.1 Loading phase

At the beginning, all stages of the three registers are set to zero. Each LFSR is regularly clocked 64 times as each key bit, \( k_i \), is XORed with the register feedback to form the new value of stage \( s_0 \). Following this, each register is regularly clocked 22 times as the IV is loaded in the same manner (Biryukov et al. 2001). At the end of the loading phase, the register contents form the loaded state. Note that the state update function during the loading phase is entirely linear. That is, the contents of each stage in each register comprise a linear combination of key and IV bits.

#### 2.1.2 Diffusion phase

The diffusion phase involves performing 100 iterations of the initialisation state update function using the majority clocking scheme. During this phase, no output is produced. At the end of diffusion phase the cipher is in its initial state and is ready for keystream generation.

### 2.2 Keystream Generation

Keystream generation comprises 228 iterations using the same majority clocking rule used during the diffusion phase. In each iteration, the keystream bit is obtained by XORing the output bit of the three registers. That is, \( s_t = s_{a,t}^8 \oplus s_{b,t}^{10} \oplus s_{c,t}^{10} \).

A conversation between two parties \( A \) and \( B \) is sent as a series of frames, each of 4.6 milliseconds duration. Each frame uses 228 bits of keystream: 114 bits to communicate from \( A \) to \( B \) and another 114 bits to communicate from \( B \) to \( A \). All frames within a conversation are encrypted using the same key, with the frame number being used as the initialisation vector; thus, successive frames use the same key and consecutive IVs for their encryption.

### 2.3 Origin of Weak Key-IV Pairs

During the loading phase, the LFSRs in A5/1 have a non-autonomous feedback mechanism, as the value of the new bit of each LFSR during this phase depends on both the feedback bit and an external value (either the key or IV bit). However, the LFSRs operate or clock independently as the regular clocking mechanism is used during this phase. Because of the non-autonomous feedback, non-zero key-IV pairs may result in an all-zero loaded state for one or more of the registers.

During the diffusion phase and keystream generation process, the LFSRs of A5/1 have an autonomous feedback mechanism, as there is no direct external input to the LFSRs. However, the LFSRs operate dependently during these processes, since each LFSR is clocked using the majority clocking rule and depends on the values of clocking bits of other LFSRs.

As noted, the non-autonomous feedback mechanism during the loading phase implies that one or more of the LFSRs may contain all-zeros at the end of this phase, since a non-zero LFSR state is only guaranteed if the LFSR uses autonomous feedback from a non-zero starting point. On the other hand, the autonomous feedback during the diffusion and keystream generation processes guarantees that any LFSR which contained all-zeros at the end of the loading phase will remain in that state. (Although the individual bits are clocked through the LFSR, the feedback bit is always zero, so the register continues to contain only zeros.)

At the end of the initialisation process, if two or three LFSRs contain all-zero values, then the keystream generated following this specific initialisation will be constant, consisting entirely of zeros or entirely of ones. This flaw results in sending a message as either clear text if the keystream is zeros, or as the complement of the plaintext if the keystream is ones. Clearly this is not a desirable outcome. In the case of one LFSR containing all-zero values, the effective size of the state space is reduced but the keystream appears to remain random (see Section 4.3).

### 3 Analysis of the Loading Phase

As the operation of the LFSRs during the loading phase is linear, it can easily be represented in terms of matrix operations. An analysis of the matrices involved then enables us to identify the conditions under which a weak key-IV pair can occur.

The autonomous operation of each LFSR can be described in terms of a matrix equation (Lidl & Niederreiter 1997) as follows. Suppose that the stages of the LFSR are denoted as \( s^0, s^1, \ldots, s^d \), the update coefficients are \( c_0, c_1, \ldots, c_d \) and that the update function can be represented as

\[
\begin{align*}
  s_{t+1}^0 &= c_0 s_t^0 \oplus c_1 s_t^1 \oplus \ldots \oplus c_d s_t^d \\
  s_{t+1}^j &= s_t^{j-1} & 1 \leq j \leq d
\end{align*}
\]

Putting \( S_t = [s^0, s^1, \ldots, s^d]^\top \), the LFSR update (clocking) operation can be represented by the equivalent matrix equation
the states of the three LFSRs as such as A5/1, which have three LFSRs, by denoting \( \sigma \) matrices as \( N \) transition matrix of the combined system in state transition matrices as \( S \) this equation reduces to

\[
S_{t+1} = TS_t
\]

where \( \sigma \) = [1 0 ... 0]\(^T\) indicates that the new key bit is XORed into the feedback of the LFSR. If we take \( t = \tau \) to indicate the LFSR state before loading commences and iterate this process several times, from \( t = \tau \) to \( t = \tau + l \) (where \( l \) denotes the key length), we have

\[
S_{\tau+1} = TS_{\tau} \oplus \sigma k_0
\]

\[
S_{\tau+2} = T(TS_{\tau} \oplus \sigma k_0) \oplus \sigma k_1 = T^2 S_{\tau} \oplus T\sigma k_0 \oplus \sigma k_1
\]

\[
S_{\tau+l} = T^l S_{\tau} = \tau^{l-1} \sigma k_0 \oplus T^{l-2} \sigma k_1 \oplus \cdots \oplus T \sigma k_{l-2} \quad \oplus \sigma k_{l-1}
\]

\[
= T^l S_{\tau} \oplus NK
\]

where \( N = [T^{l-1} \sigma \ T^{l-2} \sigma \ldots T \sigma \ \sigma] \) and \( K = [k_0 \ k_1 \ldots k_{l-2} \ k_{l-1}]\)\(^T\).

The above analysis can be extended easily to cases such as A5/1, which have three LFSRs, by denoting the states of the three LFSRs as \( A, B \) and \( C \), their state transition matrices as \( T^A, T^B \) and \( T^C \), and their \( \sigma \) matrices as \( \sigma^A, \sigma^B \) and \( \sigma^C \) and defining the state transition matrix of the combined system in matrix block form as

\[
T = \begin{bmatrix} T^A & 0 & 0 \\ 0 & T^B & 0 \\ 0 & 0 & T^C \end{bmatrix}
\]

acting on

\[
\begin{bmatrix} S_A \\ S_B \\ S_C \end{bmatrix}
\]

Likewise, denoting the \( \sigma \) and \( N \) matrices of each LFSR as \( \sigma^A, \sigma^B, \sigma^C \) and \( N^A, N^B, N^C \), the combined \( \sigma \) and \( N \) matrices for the whole system can be defined as

\[
\sigma = \begin{bmatrix} \sigma^A \\ \sigma^B \\ \sigma^C \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} N^A \\ N^B \\ N^C \end{bmatrix}
\]

With these modifications, the equation above, \( S_{\tau+1} = TS_{\tau} \oplus NK \), is also valid for the combined system. Noting that \( S_\tau = [0 \ 0 \ldots 0] \)\(^T\) for the LFSRs of A5/1, this equation reduces to \( S_\tau = NK \).

A similar analysis can also be undertaken for loading the IV bits into the LFSRs. For a 64-bit key and a 22-bit IV, we have

\[
S_{\tau+64} = NK
\]

\[
S_{\tau+86} = T^{22} NK \oplus MV
\]

where \( M = [T^{22} \sigma \ T^{20} \sigma \ T \sigma \ \sigma] \) and \( V = [v_0 \ v_1 \ v_{20} \ v_{21}]\).

Now set \( \tau = -86 \), so that \( S_0 \) represents the loaded state of the system, and consider the behaviour of the loading phase of A5/1. The terms \( T^{22}N \) and \( MV \) are XORED together. So, they can be represented by concatenating \( T^{22}N \) and \( M \) and multiplying by a new vector \( KV \), where \( KV = [k_0 \ k_1 \ldots k_{63} \ v_0 \ v_1 \ldots v_{20} \ v_{21}]\) as follows

\[
S_0 = [T^{22}N][M]KV
\]

We can apply Equation 2 to each LFSR of A5/1. Let \( \sigma_A, \sigma_B \) and \( \sigma_C \) represent the required matrices for LFSR A. Similarly, matrices can be written for LFSRs B and C as follows \( \sigma_B, \sigma_C \), \( M_B \) and \( N_C \), \( M_C \) respectively. The new equations for each LFSR can be written as follows:

\[
S_{A,\tau+86} = T^{22}N_A K \oplus M_A V
\]

\[
S_{B,\tau+86} = T^{22}N_B K \oplus M_B V
\]

\[
S_{C,\tau+86} = T^{22}N_C K \oplus M_C V
\]

Now, we apply Equation 3 to each LFSR of A5/1, the equations are as follows:

\[
S_{A,0} = [T^{22}N_A][M_A]KV
\]

\[
S_{B,0} = [T^{22}N_B][M_B]KV
\]

\[
S_{C,0} = [T^{22}N_C][M_C]KV
\]

4 Findings and Results

This analysis investigates the effect on the security of the A5/1 cipher when the loading phase results in a loaded state with one or more LFSRs containing all-zero values. We call any of these states a weak loaded state. As discussed in Section 2.3, if the contents of any LFSR is all-zeros after the loading phase, it will remain so for the whole of the diffusion and keystream generation phases, since the LFSR has no external input during these phases. Thus, the LFSR contents will not be changed until the cipher is re-initialised to encrypt the next frame in the conversation.

The total size of the secret key and IV for A5/1 is 64 + 22 = 86 bits, which exceeds the 64-bit state size. In addition, the state-update function is linear during the loading phase. From this, it follows that there are 2\(^{22}\) weak key-IV pairs corresponding to each possible weak loaded state. In fact, each pair comprises exactly one of the 2\(^{22}\) possible IVs and a corresponding key.

To investigate the effects of weak loaded states, we consider three scenarios: three LFSRs contain all-zero values, two LFSRs contain all-zero values and one LFSR contains all-zero values, as discussed later. For each scenario, we identify the relationships between key and IV bits that correspond to the weak key-IV combinations.

4.1 Three LFSRs all Zeros

This section focuses on the scenario where all three LFSRs contain all-zero values after completing the loading phase. These three LFSRs will be clocked and produce all zero output bits continuously until the next rekeying. Thus the keystream bits obtained by XORing these three outputs will also be zeros.

From Equation 3, the loaded contents of these three LFSRs are expressed in the three terms \( T^{22}N \odot |M_A|, T^{22}N \odot |M_B| \) and \( T^{22}N \odot |M_C| \). We
start with assuming these three terms \(T_{22}^{22}N_A||M_A\), \(T_{B}^{22}N_B||M_B\), \(T_{C}^{12}N_C||M_C\) are equal to all-zeros and analyse the resulting system of equations using Gaussian Elimination. This process generates the relationship between key and IV bits that result in all three LFSRs containing only zeros. We have 64 equations with 86 variables, and hence 22 variables can be chosen freely.

In particular, we may choose the IV bits \((v_0, v_1, v_2)\) freely and use these to determine the corresponding 64-bit secret key that will result in all three registers containing all-zeros. The 22 free IV bits specify the 64 key bits according to the system of equations presented in Appendix A. By choosing all possible values of the 22 free IV bits, we see that the total number of weak key-IV pairs of this type is \(2^{22}\) and the probability that a randomly chosen key-IV pair satisfies these equations is \(2^{-64}\).

As each of these pairs corresponds to a single IV and an associated key, the probability that a randomly chosen key belongs to a weak pair of this type is \(2^{22}/2^{64} = 2^{-42}\). Since each conversation is encrypted with a single key, this is equivalent to stating that the probability of a single (randomly chosen) conversation being encrypted with one of these keys is also \(2^{-42}\).

Now consider a conversation encrypted with a key which belongs to one of these weak key-IV pairs. If the conversation contains \(N\) frames, there is a further probability of \(N/2^{22}\) that the IV from this weak pair has been used to encrypt one of these frames. Therefore, there is an overall probability of \(N \times 2^{-64}\) that a randomly chosen conversation contains a frame that has been encrypted with a weak key-IV pair of this sort.

Based on this analysis, Table 1 shows an example of a weak key-IV pair that produces a fixed all-zero keystream. The keystream is presented in hexadecimal notation, to save space. As expected, the three LFSRs \(A\), \(B\) and \(C\) contain all-zero values after performing the loading phase.

### 4.2 Two LFSRs all Zeros

This section focuses on the scenario in which two LFSRs contain all-zero values after performing the loading phase. As the clocking stage in each of these two LFSRs will contain a zero, the majority value will be zero. Hence, these two LFSRs will be clocked every time; however, the contents of these two LFSRs will remain all-zeros. The third LFSR will be clocked only until the content of its clocking stage has value “1”. Since the diffusion phase consists of 100 clocking steps before producing any keystream bits, this process will ensure that the third LFSR will be in a fixed state before the keystream generation begins.

The keystream bit \(z\) is obtained by XORing the contents of the final stage of each LFSR: \(s_0^{18}, s_2^{22}\) and \(s_2^{22}\) for LFSRs \(A\), \(B\) and \(C\) respectively, since \(z_t = s_0^{18} \oplus s_2^{22} \oplus s_2^{22}\). The final stage of the non-zero LFSR will be fixed, and could contain either 0 or 1, while the final stages of the other two LFSRs contain zeros. The value in this stage of the fixed register is the value of the key stream bit. Thus the keystream has constant value for the entire frame.

Using the equations developed in Section 3, we can relate the contents of the three LFSRs after the loading phase to the key and IV bits that were loaded into these LFSRs. From Equation 3, the behaviour of these three LFSRs is expressed in the three terms \(T_{A}^{22}N_A||M_A\), \(T_{B}^{22}N_B||M_B\) and \(T_{C}^{22}N_C||M_C\). Our analysis is conducted under the assumption that we are looking for two LFSRs which have zero content in each stage of these two LFSRs after performing the loading phase, as shown in the three cases:

- **Case 1** LFSRs \(A\) and \(B\) have zero content
- **Case 2** LFSRs \(A\) and \(C\) have zero content
- **Case 3** LFSRs \(B\) and \(C\) have zero content

Note in passing that the scenario where all three LFSRs contain all-zero values is the conjunction of these three cases.

As before, we start with the terms \(T_{A}^{22}N_A||M_A\), \(T_{B}^{22}N_B||M_B\), \(T_{C}^{22}N_C||M_C\) and \(T_{B}^{22}N_B||M_B\), \(T_{C}^{22}N_C||M_C\). Setting each term equal to all-zeros and applying Gaussian Elimination will give us the conditions and relationships between key and IV bits that apply to these weak key-IV pairs. For the three cases, we have 41, 42, 45 equations respectively with 86 variables, and hence 45, 44 and 41 variables can be chosen freely.

Conditions and probabilities of the weak key-IV combinations depend on the number of available free bits. Table 2 shows the number of key and IV free bits that must be chosen to form a 64-bit secret key that results in a weak key-IV pair of each type. All of these weak key-IV pairs result in two all-zero LFSRs and produce fixed keystream for the entire frame.

### Table 2: Number of free bits for weak key-IV of each case for two LFSRs all-zero

<table>
<thead>
<tr>
<th>cases</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>key free bits</td>
<td>23</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td>Involved IV bits</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
</tbody>
</table>

**Case 1**: We consider first the case where LFSRs \(A\) and \(B\) both contain all-zeros. In this case, a 64-bit secret key can be determined from 23 free key bits \((k_{b1}, k_{b2}, k_{b3}\) to \(k_{b6})\) and 22 IV bits \((v_0, v_1\) to \(v_{20}, v_{21})\). These 45 bits specify the remaining 41 key bits using the system of equations in Appendix B. Therefore, by choosing all possible values for the 23 key free bits and the 22 IV bits, the total number of weak key-IV pairs for Case 1 is \(2^{45}\) and the probability that a randomly chosen key-IV pair satisfies this system of equations is \(2^{-42}\). It can be shown that each weak pair for Case 1 involves a unique key, so the probability that a randomly chosen key belongs to a weak pair of

```
<table>
<thead>
<tr>
<th>key</th>
<th>loaded</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Table 3: Two examples of weak key-IV (Case 1)

<table>
<thead>
<tr>
<th>key</th>
<th>010101010101000101010101011011010</th>
<th>001101010101111111111111111111110</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV</td>
<td>111000000000000000000000000000000</td>
<td>111000000000000000000000000000000</td>
</tr>
<tr>
<td>loaded</td>
<td>A 000000000000000000000000000000000</td>
<td>000000000000000000000000000000000</td>
</tr>
<tr>
<td>state</td>
<td>B 11111111110010100110101010110011</td>
<td>001010101011010001111111111111110</td>
</tr>
<tr>
<td>initial</td>
<td>A 000000000000000000000000000000000</td>
<td>001010100000011101111111111111110</td>
</tr>
<tr>
<td>state</td>
<td>B 000000000000000000000000000000000</td>
<td>001010111011011100011111111111111</td>
</tr>
<tr>
<td>C 000000000000000000000000000000000</td>
<td>001010110111111111111111111111110</td>
<td></td>
</tr>
<tr>
<td>Keystream 0x000000000000000000000000</td>
<td>000000000000000000000000000000000</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Two examples of weak key-IV (Case 2)

<table>
<thead>
<tr>
<th>key</th>
<th>010101001000101101101101011011010</th>
<th>001101010101111111111111111111110</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV</td>
<td>111000000000000000000000000000000</td>
<td>111000000000000000000000000000000</td>
</tr>
<tr>
<td>loaded</td>
<td>A 000000000000000000000000000000000</td>
<td>000000000000000000000000000000000</td>
</tr>
<tr>
<td>state</td>
<td>B 010101001011111111111111111111110</td>
<td>001010101011010001111111111111110</td>
</tr>
<tr>
<td>initial</td>
<td>A 000000000000000000000000000000000</td>
<td>001010100000011101111111111111110</td>
</tr>
<tr>
<td>state</td>
<td>B 000000000000000000000000000000000</td>
<td>001010111011011100011111111111111</td>
</tr>
<tr>
<td>C 000000000000000000000000000000000</td>
<td>001010110111111111111111111111110</td>
<td></td>
</tr>
<tr>
<td>Keystream 0x000000000000000000000000</td>
<td>000000000000000000000000000000000</td>
<td></td>
</tr>
</tbody>
</table>

Based on this analysis, Table 3 shows two examples of weak key-IV pairs that produce fixed keystream (either zeros or ones). The keystream is presented in hex, the bold bits are the output bits. These 41 bits specify the remaining 45 key bits using the system of equations in Appendix B. Therefore, by choosing all possible values for the 19 key free bits and the 22 IV bits, the total number of weak key-IV for Case 1 is $2^{41}$ and the probability that a randomly chosen key-IV pair satisfies these equations is $2^{-41}$.

Case 2: We now consider the case where LFSRs $A$ and $C$ both contain all-zeros. As in the previous cases, it is possible to obtain a weak key-IV combination for A5/1 from this condition. For Case 2, a 64-bit secret key is formed from 22 key free bits ($k_{21}$ to $k_{33}$) and 22 IV bits ($v_0$, $v_1$ to $v_{21}$). These 44 bits specify the remaining 42 key bits using the system of equations in Appendix B. Therefore, by choosing all possible values for the 22 key free bits and the 22 IV bits, the total number of weak key-IV for Case 2 is $2^{44}$ and the probability that a randomly chosen key-IV pair satisfies this system of equations is $2^{-44}$. Each weak pair for Case 2 again involves a unique key, so the probability that a randomly chosen key belongs to a weak pair of this sort is $2^{-23}$ and the probability that a randomly chosen conversation of $N$ frames contains a frame that has been encrypted with a Case 2 key-IV pair is $N \times 2^{-44}$.

Based on this analysis, Table 4 shows two examples of weak key-IV pairs that produce fixed keystream (either zeros or ones). Again, the keystream is presented in hex, the bold bits are the clocking taps and the underlined bits are the output bits. In these examples, the keystream bits have the same value as the output bit of LFSR $B$.

Case 3: Consider finally the case where LFSRs $B$ and $C$ both contain all-zeros. For this case, a 64-bit secret key is formed from 19 key free bits ($k_{25}$ to $k_{44}$) and 22 IV bits ($v_0$, $v_1$ to $v_{21}$). These 41 bits specify the remaining 45 key bits using the system of equations in Appendix B. Therefore, by choosing all possible values for the 19 key free bits and the 22 IV bits, the total number of weak key-IV for Case 3 is $2^{41}$ and the probability that a randomly chosen key-IV pair satisfies these equations is $2^{-45}$. As for the other cases, each weak pair for Case 3 involves a unique key, so the probability that a randomly chosen key belongs to a weak pair of this sort is $2^{-23}$ and the probability that a randomly chosen conversation of $N$ frames contains a frame that has been encrypted with a Case 3 key-IV pair is $N \times 2^{-45}$.

Based on this analysis, Table 5 shows two examples of weak key-IV pairs that produce fixed keystream (either zeros or ones). Once again, the keystream is presented in hex, the bold bits are the clocking taps and the underlined bits are the output bits. In these examples, the keystream bits have the same value as the output bit of LFSR $A$.

Collating the results above, the number of weak key-IV combinations for Cases 1, 2 and 3 are $2^{45}$, $2^{44}$ and $2^{41}$ respectively. Moreover, apart from the scenario with three all-zero registers, the resulting key-IV pairs are distinct. Therefore, the total number of weak key-IV combinations, when two LFSRs are all-zero, is $2^{45}$.

Likewise, the probability that a randomly chosen key-IV pair satisfies the equations for any of these cases is found to be $2^{-40.36}$ and the probability that a randomly chosen conversation of $N$ frames contains a frame that has been encrypted with any of these key-IV pairs is $N \times 2^{-40.36}$.

4.3 One LFSR all Zeros

It is possible to find that exactly one LFSR contains all-zero values after the loading phase. Whether this LFSR is clocked or not during the keystream generation, the contribution to the keystream bit from this LFSR is zero. In this situation at least one of the other two LFSRs will be clocked. The value of the
keystream bits actually depends only on the content of the last bit of these other two LFSRs.

Once again, we use the equations developed in Section 3 to relate the contents of the three LFSRs after the loading phase to the key and IV bits that were loaded into these LFSRs. From Equation 3, the contents of the three LFSRs are expressed in the three terms $T^2 N_A |M_A$, $T^2 N_B |M_B$ and $T^2 N_C |M_C$. Our analysis is conducted under the assumption that we are looking for a single LFSR which has all-zeros in its stages after performing the loading phase, as shown in the three cases:

**Case 4** LFSR $A$ contains all-zero values

**Case 5** LFSR $B$ contains all-zero values

**Case 6** LFSR $C$ contains all-zero values

The pairwise conjunctions of these cases are the three cases from the scenario with two all-zero registers (Section 4.2), while the triple conjunction is the scenario where all three LFSRs contain all-zero values.

Equation 3 is analysed using Gaussian Elimination (GE) to find the conditions mentioned above. We start by setting the term $(T^2 N_A |M_A)$ equal to zero to find the relationship between key and IV bits. This process is applied similarly for the other two terms $(T^2 N_B |M_B)$ and $(T^2 N_C |M_C)$. This process generates the relationship between key and IV bits that result in an LFSR containing all-zero values after performing the loading phase. For the three cases (4, 5 and 6), we have 19, 22, 23 equations respectively. For each case, there are 86 variables, and hence 67, 64 and 63 variables can be chosen freely in the respective cases.

The conditions and the probabilities of the weak key-IVs depend on the number of free bits. Table 6 shows the number of key and IV free bits that must be chosen to form a 64-bit secret key that results in a weak key-IV that results in one LFSR having all-zero values for the entire frame.

**Case 4:** For Case 4, a 64-bit secret key is formed from 45 key free bits ($k_{19}$, $k_{20}$ to $k_{63}$) and 22 IV bits ($v_0$, $v_1$ to $v_{21}$). These 67 bits specify the remaining 19 key bits using the relevant system of equations. Therefore, by choosing all possible values for the 45 and 22 key and IV bits respectively, the total number of weak key-IV pairs for Case 4 is $2^{67}$ and the probability that a randomly chosen key-IV pair satisfies these equations is $2^{-19}$. In this case, each of the $2^{64}$ possible keys belongs to exactly 8 Case 4 key-IV pairs. For any given key, the IVs in these pairs are distributed almost regularly through the IV space, so the probability that a randomly chosen conversation of $N$ frames contains a frame that has been encrypted with a Case 4 key-IV pair is $N \times 2^{-19}$ for $N \leq 524043$ and rises to 1 for $N \geq 524339$. (Note: 2$^{19} = 524288$.)

Based on this analysis, Table 7 shows an example of a weak key-IV pair that results in LFSR $A$ containing all-zero values until the next rekeying. The keystream is presented in hex and depends only on LFSRs $B$ and $C$. LFSR $A$ contains all-zero values after performing the loading phase due to the nonautonomous feedback operation during that phase.

**Case 5:** For Case 5, a 64-bit secret key is formed from 42 key free bits ($k_{22}$, $k_{23}$ to $k_{63}$) and 22 IV bits ($v_0$, $v_1$ to $v_{21}$). These 66 key-IV bits specify the remaining 22 key bits using the relevant system of equations. Therefore, by choosing all possible values for the 42 key free bits and the 22 IV free bits, the total number of weak key-IV for Case 5 is $2^{64}$ and the probability that a randomly chosen key-IV pair satisfies these equations is $2^{-22}$. Each of the $2^{64}$ possible keys belongs to exactly one Case 5 key-IV pair and the probability that a randomly chosen conversation of $N$ frames contains a frame that has been encrypted with a Case 5 key-IV pair is $N \times 2^{-22}$. Based on this analysis, Table 8 shows an example of a weak key-IV pair that produces a keystream from only two effective LFSRs $A$ and $C$. LFSR $B$ contains all-zero values after performing the loading phase due to the nonautonomous operation.

<table>
<thead>
<tr>
<th>Case</th>
<th>LFSR</th>
<th>Contains all-zero values</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5: Two examples of weak key-IV (Case 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>IV</strong></td>
</tr>
<tr>
<td><strong>Loaded state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Initial state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Keystream</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6: Number of free bits for weak key-IV of each Case for one LFSR all-zero</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>IV</strong></td>
</tr>
<tr>
<td><strong>Loaded state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Initial state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Keystream</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7: Example of weak key-IV (Case 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>IV</strong></td>
</tr>
<tr>
<td><strong>Loaded state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Initial state</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Keystream</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Case 6: For Case 6, a 64-bit secret key is formed from 41 key free bits \((k_{23}, k_{24} \text{ to } k_{63})\) and 22 IV bits \((v_0, v_1 \text{ to } v_{21})\). These 63 bits specify the remaining 23 key bits using the relevant system of equations. Therefore, by choosing all possible values for the 41 key free bits and the 22 IV free bits, the total number of weak key-IV pairs for Case 6 is \(2^{63}\) and the probability that a randomly chosen key-IV pair satisfies these equations is \(2^{-23}\). Each weak pair for Case 6 involves a unique key, so the probability that a randomly chosen key belongs to a weak pair of this sort is one-half. And the probability that a randomly chosen conversation of \(N\) frames contains a frame that has been encrypted with a Case 6 key-IV pair is \(N \times 2^{-23}\).

Based on this analysis, Table 9 shows an example of a weak key-IV pair that produces a keystream using two effective LFSRs \(A\) and \(B\). LFSR \(C\) contains all-zero values after performing the loading phase until the next rekeying due to the nonautonomous operation during the loading phase.

<table>
<thead>
<tr>
<th>key</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>001110000100011101111111111111111111111111</td>
<td>A 001100110111011111111111111111111111111111</td>
</tr>
<tr>
<td>011110000000000000000000000000000000000000</td>
<td>B 011101101111111101111111111111111111111111</td>
</tr>
<tr>
<td>111110000000000000000000000000000000000000</td>
<td>C 111110000000000000000000000000000000000000</td>
</tr>
</tbody>
</table>

Keystream: 0x05f6260d5acedef21e25c9cd1b78e0e
0a67dce90f26b3e29e9013206

Table 9: Examples of weak key-IV (Case 6)

Collating the results for this scenario, the number of weak key-IV pairs for Cases 4, 5 and 6 are \(2^{67}\), \(2^{64}\) and \(2^{63}\) respectively. Therefore, the total number of weak key-IV pairs that result in one LFSR of A5/1 containing all-zeros is \(2^{67}\), and the probability that a randomly chosen key-IV pair satisfies the equations for any of these cases is \(2^{-18.75}\). (The key-IV pairs from Cases 1, 2 and 3 are included among these, but form a negligible proportion (approximately \(2^{-21.6}\)) of this total.) Each of the \(2^{64}\) possible keys belongs to 8 Case 4 key-IV pairs and one Case 5 pair, while half of them belong to a Case 6 pair as well. Thus, every conversation is encrypted with a key that belongs to either 9 or 10 weak key-IV pairs from Cases 4, 5 or 6, and for \(N < 2^{19}\) the probability that a randomly chosen conversation of \(N\) frames contains a frame that has been encrypted with a weak pair of this sort is approximately \(9.5 \times N \times 2^{-22} = N \times 2^{-18.75}\). (For \(N\) approaching \(2^{19}\), this probability rises to 1 for \(N \geq 524339\).

Statistical analysis

In the scenarios where two or three registers contain all-zero values, the keystream is constant and hence immediately distinguishable, but this is not the case for the scenario in which a single register contains all-zero values. On the other hand, the latter scenario potentially occurs in every conversation and its probability of occurrence in a conversation of moderate length is quite high, so it is worth investigating whether the key-stream from this scenario can also be distinguished from the key-stream obtained when none of the registers contains all-zero values. The rest of the section therefore focuses on statistical analyses which might distinguish a keystream from A5/1 that is generated by two LFSRs only (while the third LFSR contains all-zero values) from another keystream generated by three LFSRs (in normal operation).

Gustafson (Gustafson 1996) described statistical randomness tests of symmetric ciphers and the National Institute of Standards and Technology (NIST) (Rukhin et al. 2010) gives 15 tests of the randomness of a given sequence of bits to measure the degree of randomness of that given sequence of bits.

The most powerful tests are examined: Balance, Runs tests and Linear complexity. Balance test is also referred as Frequency (Monobit) test in some references, and Uniformity test in others. Balance test measure the proportion of zeros and ones for an entire sequence. The number of zeros and ones for a given sequence should meet the conditions of the randomness test, with probabilities \(P(0) = P(1) = \frac{1}{2}\).

The Runs test focuses on the total number of runs in a sequence. A run is an unbroken sequence of a particular character, either zeros or ones. Linear Complexity determines the smallest LFSR that can generate the whole keystream over the finite field \(F_2^n\) using Berlekamp-Massey algorithm (Menezes et al. 1996).

Keystream sequences are generated for different scenarios of the targeted operation of the A5/1 cipher as follows:

- None of the LFSRs \(A, B\) or \(C\) contain all-zero values (\(A, B\) and \(C\) all participate to form the keystream bits)
- LFSR \(A\) contains all zeros (\(B\) and \(C\) participate to form the keystream bits)
- LFSR \(B\) contains all zeros (\(A\) and \(C\) participate to form the keystream bits)
- LFSR \(C\) contains all zeros (\(A\) and \(B\) participate to form the keystream bits)

For each scenario, we performed an experiment involving computer simulation. In each trial, we generated \(10^4\) random loaded states using C library seeded random function to generate a 228-bit keystream segment (frame) from each loaded state. The Balance and Runs test are applied for these generated frames. Figure 2 demonstrates the Balance test for each scenario. Table 10 shows the result of the Runs test.

Figures 3 and 4 show the result of linear complexity of A5/1 for frame length 228 bits and 2000 bits respectively.
5 Attack Procedure

Since A5/1 has a 64-bit key and a 64-bit internal state, it is not feasible either to guess the internal state and generate keystream or to guess the whole secret key and generate keystream for any given IV. However, if it is possible to identify that the keystream used to encrypt a specific frame arose from one of a reduced set of initial states with a known format, this enables an attacker to reveal the secret key with reduced complexity. In our case, the keystream generated when two or three LFSRs contain all-zeros is immediately identifiable (since it either consists entirely of zeros or entirely of ones), so the occurrence of these scenarios is likewise distinguishable and an attack can be mounted. Moreover, the redundancy of speech enables us to recognise these frames in a ciphertext-only context, which is the least restrictive scenario for an attacker.

This section therefore presents a procedure for identifying when a keystream sequence is either all zeros or all ones and attacking the cipher by finding the corresponding secret key. This procedure covers all cases where two or three LFSRs contain all-zeros, so it is not necessary to determine beforehand which two (or three) of the registers contain all-zeros. Note that the procedure focuses on finding the secret key, rather than on decrypting an individual frame assuming the IV’s are known. Once the key is obtained, all other frames in the conversation can then be decrypted and the entire message recovered.

Attack Algorithm

The following algorithm is broken into two phases. The first phase is not guaranteed to succeed; however, if the first phase does succeed, the entire algorithm will succeed and the entire message (conversation) will be recovered.

Phase 1 Given an encrypted conversation:

Step 1: Divide the encrypted speech (ciphertext) into separate frames. Each ciphertext frame corresponds to one (known) IV.

Step 2: For each frame, check if either the frame or...
its bitwise complement is intelligible. If so, proceed to Phase 2.

This phase identifies any frames encrypted with a keystream sequence that is either all zeros or all ones. (If any encrypted frame is intelligible and seems to be plaintext, this indicates that the keystream is all zeros, while if the complement of any encrypted frame is intelligible and seems to be plaintext, this indicates that the keystream is all ones.) If none of the frames in the conversation satisfy this test, the attack fails for this conversation.

**Phase 2** Given a frame identified in Phase 1

**Step 1:** For each of Cases 1, 2 and 3 in turn:

- Guess the free key bits in the relevant set of equations from Appendix B.
- Use these guessed free key bits and the known IV to calculate the remaining key bits using the relevant system of equations.
- Set the guessed and calculated key bits as the current trial key.
- For any other frame in the conversation:
  - Use the A5/1 algorithm to generate keystream from the trial key and the IV of that frame.
  - Try to decrypt the encrypted frame using the generated keystream.
    * If the encrypted frame is decrypted successfully, then the secret key has been identified.
    * If not, repeat the process for another guess.

**Step 2:** Once the secret key has been found, use this secret key with the known IVs for each frame to decrypt the entire intercepted ciphertext.

This phase checks all potential weak keys until the correct one is found. Since the scenario with three all-zero registers is a special case of the scenario with two all-zero registers, it will also be covered by this search.

**Discussion and Attack Complexity:** As mentioned previously, the scenarios in which two or three LFSRs contain all-zeros lead to a keystream which is either all zeros or all ones. The corresponding frames of conversation are easily distinguishable, so we refer to these as weak frames. When such a frame is observed, the whole conversation can be decrypted after guessing and checking up to $2^{21.64}$ possible keys to obtain the actual secret key. As discussed in Section 4.2, the probability that a randomly chosen conversation of $N$ frames contains a weak frame of this sort is $N \times 2^{-40.36}$.

Table 1 lists some typical values of $N$ for various lengths of conversation, together with the corresponding probability that a weak frame will be observed in a conversation of that length. If we take 5 minutes as a common length of conversation, we see that roughly one in $2^{24.36} \approx 21.5 \times 10^6$ conversations of this length can be completely decrypted using this attack.

We have also examined the scenario in which only one register contains all-zero values. The keystream generated under this scenario is not immediately distinguishable, so we applied some common statistical randomness tests to see whether these would enable this scenario to be distinguished. Although we have been unable to find a test that does so, this scenario is of great interest as it potentially occurs in every conversation and its probability of occurrence in a conversation of moderate length is quite high, rising to 1 for $N$ just over $2^{19}$ (equivalent to 40 minutes of conversation). Moreover, although the approximation of $N \times 2^{-18.75}$ for this probability does not hold exactly as $N$ approaches $2^{19}$, it is clear that the probability becomes alarmingly high (greater than 1 in 7) for conversations as short as 5 minutes in duration. On the other hand, an attack based on this scenario would require guessing and checking of up to $2^{45.25}$ possible keys in order to find the correct one for that particular conversation.

**Table 11:** Number of frames for various lengths of conversation

<table>
<thead>
<tr>
<th>No of frames</th>
<th>Conversation time</th>
<th>Probability of weak frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{14}$</td>
<td>1min 16sec</td>
<td>$2^{-26.36}$</td>
</tr>
<tr>
<td>$2^{16}$</td>
<td>5min 2sec</td>
<td>$2^{-24.36}$</td>
</tr>
<tr>
<td>$2^{18}$</td>
<td>20min 6sec</td>
<td>$2^{-22.36}$</td>
</tr>
<tr>
<td>$2^{20}$</td>
<td>1h 21min</td>
<td>$2^{-20.36}$</td>
</tr>
<tr>
<td>$2^{22}$</td>
<td>5h 22min</td>
<td>$2^{-18.36}$</td>
</tr>
</tbody>
</table>

6 Conclusion

The non-autonomous feedback mechanism which applies during the loading phase of A5/1 may result in weak key-IV pairs, where one or more of the LFSRs contain all zero values at the end of the loading phase. If this condition occurs, it is then maintained during the diffusion and keystream generation phases, potentially leading to a weak keystream.

This paper has considered three scenarios related to weak key-IV pairs in A5/1: all three LFSRs contain all-zero values, exactly two LFSRs may contain all-zero values or a single LFSR may contain all-zero values. For a conversation containing $N$ frames, the probabilities that the conversation includes a frame satisfying these respective scenarios are $N \times 2^{-64}$, $N \times 2^{-40.36}$ and approximately $N \times 2^{-18.75}$.

When either of the first two scenarios occurs, the corresponding frame is immediately distinguishable, as it contains either plaintext only or complemented plaintext only. Once such a frame has been identified, the secret key can easily be obtained and the entire conversation can then be decrypted.

We have presented a ciphertext-only attack which exploits this weakness; the probability of a successful attack on a five minute conversation is approximately one in 20 million and the complexity of a successful attack is approximately $2^{23.64}$. (Due to the distinguishing nature of this attack, no calculations are required in cases where the targeted scenario does not occur. In particular, no pre-computation is required.)

The third scenario (when a single LFSR contains all-zeros) could potentially lead to a similar attack, provided the resulting keystream can be distinguished. However, finding a distinguisher for this keystream remains an open problem for the moment. If such a distinguisher could be found, conversations as short as 5 minutes in duration would be at high risk of attack, although the associated complexity would be higher.

It should be noted that in the current specification of A5/1, there is no pre-testing of the key-IV pair to
identify whether the pair is weak or not, nor is the loaded state checked to determine whether it satisfies any of the above scenarios, but the keystream is used in any case. The weak loaded states arise due to the non-autonomous feedback used during the loading phase, so a completely different loading mechanism would be required if the occurrence of such states is to be avoided. Alternatively, the loaded state could be checked each time the algorithm is re-initialised and a suitable adjustment could be applied if it was found to be a weak state.

References


A Three LFSRs contain all-zeros values

The conditions for the system of equations that result in all three LFSRs containing all-zero values can be described as follows:

\[ k_0 = v_0 \oplus v_1 \oplus v_2 \oplus v_6 \oplus v_8 \oplus v_{11} \oplus v_{18} \]
\[ k_1 = v_1 \oplus v_2 \oplus v_6 \oplus v_7 \oplus v_9 \oplus v_{12} \oplus v_{19} \]
\[ k_2 = v_2 \oplus v_3 \oplus v_7 \oplus v_8 \oplus v_{10} \oplus v_{13} \oplus v_{20} \]
\[ k_3 = v_3 \oplus v_4 \oplus v_8 \oplus v_9 \oplus v_{11} \oplus v_{14} \oplus v_{21} \]
\[ k_4 = v_4 \oplus v_5 \oplus v_9 \oplus v_{10} \oplus v_{12} \oplus v_{15} \]
\[ k_5 = v_5 \oplus v_6 \oplus v_{10} \oplus v_{11} \oplus v_{14} \]
\[ k_6 = v_6 \oplus v_7 \oplus v_{11} \oplus v_{12} \oplus v_{15} \oplus v_{17} \]
\[ k_7 = v_7 \oplus v_8 \oplus v_{12} \oplus v_{13} \oplus v_{15} \oplus v_{18} \]
\[ k_8 = v_8 \oplus v_9 \oplus v_{13} \oplus v_{14} \oplus v_{16} \oplus v_{18} \oplus v_{19} \]
\[ k_9 = v_9 \oplus v_{10} \oplus v_{15} \oplus v_{16} \oplus v_{17} \oplus v_{19} \oplus v_{20} \]
\[ k_{10} = v_1 \oplus v_2 \oplus v_3 \oplus v_7 \oplus v_{10} \oplus v_{12} \oplus v_{14} \oplus v_{15} \oplus v_{17} \oplus v_{19} \oplus v_{20} \]
\[ k_{11} = v_2 \oplus v_3 \oplus v_7 \oplus v_{10} \oplus v_{11} \oplus v_{13} \oplus v_{14} \oplus v_{16} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{12} = v_3 \oplus v_4 \oplus v_7 \oplus v_{10} \oplus v_{11} \oplus v_{13} \oplus v_{15} \oplus v_{17} \oplus v_{18} \oplus v_{20} \]
\[ k_{13} = v_4 \oplus v_5 \oplus v_7 \oplus v_{10} \oplus v_{11} \oplus v_{13} \oplus v_{15} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{14} = v_5 \oplus v_6 \oplus v_7 \oplus v_{10} \oplus v_{12} \oplus v_{15} \oplus v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{15} = v_6 \oplus v_7 \oplus v_8 \oplus v_{10} \oplus v_{12} \oplus v_{13} \oplus v_{16} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{16} = v_7 \oplus v_8 \oplus v_{10} \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{15} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{17} = v_8 \oplus v_9 \oplus v_{10} \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{15} \oplus v_{20} \oplus v_{21} \]
\[ k_{18} = v_9 \oplus v_{10} \oplus v_{11} \oplus v_{12} \oplus v_{13} \oplus v_{15} \oplus v_{16} \oplus v_{20} \]
\[ k_{19} = v_{10} \oplus v_{11} \oplus v_{12} \oplus v_{13} \oplus v_{15} \oplus v_{18} \oplus v_{19} \]
\[ k_{20} = v_{11} \oplus v_{12} \oplus v_{13} \oplus v_{15} \oplus v_{16} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{21} = v_{12} \oplus v_{13} \oplus v_{15} \oplus v_{16} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{22} = v_{13} \oplus v_{14} \oplus v_{16} \oplus v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{23} = v_{14} \oplus v_{15} \oplus v_{16} \oplus v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{24} = v_{15} \oplus v_{16} \oplus v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{25} = v_{16} \oplus v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{26} = v_{17} \oplus v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{27} = v_{18} \oplus v_{19} \oplus v_{20} \]
\[ k_{28} = v_{19} \oplus v_{20} \]
\[ k_{29} = v_{20} \]
\[ k_{30} = v_{20} \]
\[ k_{31} = v_{20} \]
\[ k_{32} = v_{20} \]
\[ k_{33} = v_{20} \]
\[ k_{34} = v_{20} \]
\[ k_{35} = v_{20} \]
\[ k_{36} = v_{20} \]
\[ k_{37} = v_{20} \]
\[ k_{38} = v_{20} \]
For two LFSRs contain all-zeros, there are 3 possible values of equations must be satisfied:

B Two LFSRs contain all-zeros values

For two LFSRs contain all-zeros, there are 3 possible cases of LFSRs that contain all-zeros, as mentioned previously; Case 1, 2 and 3 for LFSRs (A, B), (A, C) or (B, C) respectively. The relationship between key and IV bits that result in two LFSRs containing all-zeros values can be expressed as follows:

Case 1:

To obtain two LFSRs A and B containing all-zero values at the end of loading phase, the following system of equations must be satisfied:
Case 2:
The conditions for the system of equations that are required to result in LFSRs $A$ and $C$ containing all-zero values after completing loading phase can be expressed as follows:

$$
k_0 = k_{42} \oplus k_{44} \oplus k_{47} \oplus k_{49} \oplus k_{50} \oplus k_{51} \oplus k_{53} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{58} \oplus k_{60} \oplus k_{61} \oplus v_4 \oplus v_5 \oplus v_7 \oplus v_9 \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{16} \oplus v_{18} \oplus v_{20} \oplus v_{21}$$

$$k_1 = k_{43} \oplus k_{45} \oplus k_{48} \oplus k_{50} \oplus k_{51} \oplus k_{52} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{58} \oplus k_{60} \oplus k_{61} \oplus v_2 \oplus v_3 \oplus v_4 \oplus v_6 \oplus v_7 \oplus v_9 \oplus v_{11} \oplus v_{12}$$

$$k_2 = k_{44} \oplus k_{46} \oplus k_{49} \oplus k_{50} \oplus k_{51} \oplus k_{52} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_3 = k_{45} \oplus k_{47} \oplus k_{50} \oplus k_{51} \oplus k_{52} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_4 \oplus v_9 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_4 = k_{46} \oplus k_{48} \oplus k_{51} \oplus k_{52} \oplus k_{53} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus v_3 \oplus v_4 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_5 = k_{47} \oplus k_{49} \oplus k_{52} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{58} \oplus k_{59} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_6 = k_{48} \oplus k_{50} \oplus k_{53} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{59} \oplus k_{60} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_7 = k_{49} \oplus k_{51} \oplus k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_7 \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{15}$$

$$k_8 = k_{52} \oplus k_{54} \oplus k_{57} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_7 \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{15} \oplus v_{17}$$

$$k_9 = k_{55} \oplus k_{57} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_7 \oplus v_{11} \oplus v_{12} \oplus v_{14} \oplus v_{15} \oplus v_{17}$$

$$k_{10} = k_{44} \oplus k_{46} \oplus k_{49} \oplus k_{51} \oplus k_{53} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{11} = k_{45} \oplus k_{47} \oplus k_{50} \oplus k_{52} \oplus k_{54} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{12} = k_{46} \oplus k_{48} \oplus k_{51} \oplus k_{53} \oplus k_{55} \oplus k_{56} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{13} = k_{47} \oplus k_{49} \oplus k_{52} \oplus k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{14} = k_{48} \oplus k_{50} \oplus k_{53} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{15} = k_{49} \oplus k_{51} \oplus k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{16} = k_{50} \oplus k_{52} \oplus k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{17} = k_{51} \oplus k_{53} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{18} = k_{52} \oplus k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{19} = k_{53} \oplus k_{55} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$

$$k_{20} = k_{54} \oplus k_{56} \oplus k_{57} \oplus k_{58} \oplus k_{59} \oplus k_{60} \oplus k_{61} \oplus v_3 \oplus v_5 \oplus v_{10} \oplus v_{11} \oplus v_{12}$$
Case 3:
The system of equations that is required to result in LFSRs $B$ and $C$ containing all-zero values can be expressed as follows:

$$k_0 = k_{45} \oplus k_{46} \oplus k_{53} \oplus k_{63} \oplus v_5 \oplus v_{13} \oplus v_{14} \oplus v_{16} \oplus v_{17} \oplus v_{18}$$

$$k_1 = k_{46} \oplus k_{47} \oplus k_{53} \oplus k_{58} \oplus k_{62} \oplus v_3 \oplus v_{10} \oplus v_{14} \oplus v_{15} \oplus v_{17} \oplus v_{18}$$

$$k_2 = k_{47} \oplus k_{50} \oplus k_{53} \oplus k_{60} \oplus v_3 \oplus v_{15} \oplus v_{17} \oplus v_{19} \oplus v_{20}$$

$$k_3 = k_{48} \oplus k_{53} \oplus k_{57} \oplus k_{60} \oplus v_3 \oplus v_{15} \oplus v_{17} \oplus v_{19} \oplus v_{20}$$

$$k_4 = k_{49} \oplus k_{52} \oplus k_{58} \oplus k_{61} \oplus v_3 \oplus v_{19} \oplus v_{21}$$

$$k_5 = k_{50} \oplus k_{53} \oplus k_{59} \oplus k_{62} \oplus v_2 \oplus v_{19} \oplus v_{20}$$

$$k_6 = k_{51} \oplus k_{54} \oplus k_{57} \oplus k_{60} \oplus v_3 \oplus v_{19} \oplus v_{20}$$

$$k_7 = k_{52} \oplus k_{55} \oplus k_{58} \oplus k_6 \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_8 = k_{45} \oplus k_{58} \oplus k_{53} \oplus k_{57} \oplus k_6 \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_9 = k_{52} \oplus k_{55} \oplus k_{58} \oplus k_6 \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{10} = k_{57} \oplus k_{50} \oplus k_{53} \oplus k_{59} \oplus k_6 \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{11} = k_{58} \oplus k_{51} \oplus k_{54} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{12} = k_{59} \oplus k_{52} \oplus k_{55} \oplus k_6 \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{13} = k_{60} \oplus k_{53} \oplus k_{57} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{14} = k_{51} \oplus k_{54} \oplus k_{57} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{15} = k_{52} \oplus k_{55} \oplus k_{58} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{16} = k_{53} \oplus k_{56} \oplus k_{58} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{17} = k_{54} \oplus k_{57} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{18} = k_{55} \oplus k_{58} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{19} = k_{56} \oplus k_{59} \oplus k_{62} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{20} = k_{57} \oplus k_{60} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{21} = k_{58} \oplus k_{61} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{22} = k_{59} \oplus k_{62} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{23} = k_{60} \oplus k_{63} \oplus v_4 \oplus v_{19} \oplus v_{20}$$

$$k_{24} = k_{46} \oplus k_{49} \oplus k_{52} \oplus k_{58} \oplus k_{62} \oplus v_3 \oplus v_{10} \oplus v_{14} \oplus v_{15} \oplus v_{17} \oplus v_{19}$$

$$k_{25} = k_{47} \oplus k_{50} \oplus k_{53} \oplus k_{59} \oplus k_{62} \oplus v_3 \oplus v_{10} \oplus v_{14} \oplus v_{15} \oplus v_{17} \oplus v_{19}$$

$$k_{26} = k_{48} \oplus k_{51} \oplus k_{54} \oplus k_{57} \oplus k_{60} \oplus v_3 \oplus v_{10} \oplus v_{14} \oplus v_{15} \oplus v_{17} \oplus v_{19}$$