Appendices
Appendix A

Fig. A1 LFSR based complex code generator circuit.
Fig. A2 Multichannel encrypter circuit.
### Appendix B

**Table B  A representative Gold code sequence.**

<table>
<thead>
<tr>
<th>Clock</th>
<th>$Q_{s,1}$ (Mother code)</th>
<th>$Q_{s,III}$ (PN sequence)</th>
<th>Gold code ($Q_{s,1} \oplus Q_{s,III}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 $d_0$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0 $d_1$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0 $d_2$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0 $d_3$</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0 $d_4$</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>0</td>
<td>1 $d_{27}$</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>0 $d_{28}$</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>1</td>
<td>1 $d_{29}$</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>1</td>
<td>1 $d_{30}$</td>
</tr>
</tbody>
</table>
Appendix C
Randomness Tests

1.0 Statistical Analysis and Interpretation of Empirical Results
To test the randomness of binary sequences generated by complex code generators the NIST suite, proposed by National Institute of Standards and Technology, is used. The NIST suite includes sixteen tests [1]. These tests fix on a variety of different types of non-randomness that could exist in a sequence. All these tests are frequency (monobit), frequency within a block, runs, longest-run-of-ones in a block, binary matrix rank, discrete Fourier transform (spectral), non-overlapping template matching, overlapping template matching, Maurer’s “Universal statistical”, Lempel-Ziv compression, linear complexity, serial, approximate entropy, cumulative sums, random excursions, random excursions variant. Out of these sixteen tests some are discussed below, which gave successful results when tested for the output binary sequences of the presented schemes in this thesis.

The testing process consists of the following steps [1]:
(1) State the null hypothesis. Assume that the binary sequence is random.
(2) Compute a sequence test statistic. Testing is carried out at the bit level.
(3) Compute the P-value, P-value ∈ [0, 1]
(4) Compare the P-value to \( a \). Fix \( a \), where \( a \in [0.0001, 0.01] \). Success is declared whenever \( P \) value ≥ \( a \); otherwise, failure is declared.

2.0 Frequency (Monobit) Test

2.0.1 Test Purpose
The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test assesses the closeness of the fraction of ones to \( \frac{1}{2} \), that is, the number of ones and zeroes in a sequence. They should be about the same. All subsequent tests depend on the passing of this test; there is no evidence to indicate that the tested sequence is non-random.

2.0.2 Function Call
Frequency \((n)\), where,
\[
n \quad \text{The length of the bit string.}
\]
Additional input used by the function, but supplied by the testing code,
The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; \( \varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \).

2.0.3 Test Statistic and Reference Distribution

\( S_{\text{obs}} \) : The absolute value of the sum of the \( X_i \) (where, \( X_i = 2\varepsilon_i - 1 = \pm 1 \)) in the sequence divided by the square root of the length of the sequence.

The reference distribution for the test statistic is half normal (for large \( n \)). If \( z \), where \( z = \frac{S_{\text{obs}}}{\sqrt{2}} \) is distributed as normal, then \( |z| \) is distributed as half normal. If the sequence is random, then the plus and minus ones will tend to cancel one another out so that the test statistic will be about 0. If there are too many ones or too many zeroes, then the test statistic will tend to be larger than zero.

2.0.4 Test Description

(1) Conversion to \( \pm 1 \): The zeros and ones of the input sequence (\( \varepsilon \)) are converted to values of \(-1\) and \(+1\) and are added together to produce \( S_n = X_1 + X_2 + \ldots + X_n \), where \( X_i = 2\varepsilon_i - 1 \).

(2) Compute the test statistic \( S_{\text{obs}} = \frac{|S_n|}{\sqrt{n}} \).

(3) Compute \( P\)-value = \( \text{erfc} \left( \frac{S_{\text{obs}}}{\sqrt{2}} \right) \), where \( \text{erfc} \) is the complementary error function given by

\[
\text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty \! e^{-u^2} \, du
\]

2.1 Frequency Test within a Block

2.1.1 Test Purpose

The focus of the test is the proportion of ones within M-bit blocks. The purpose of this test is to determine whether the frequency of ones in an M-bit block is approximately \( M/2 \), as would be expected under an assumption of randomness. For block size \( M=1 \), this test degenerates to the frequency (monobit) test.

2.1.2 Function Call

Block frequency \((M, n)\), where,

\( M \) The length of each block.
\( n \) The length of the bit string.

Additional input used by the function, but supplied by the testing code,
The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; \( \varepsilon = \varepsilon_1, \varepsilon_2, ..., \varepsilon_n \).

2.1.3 Test Statistic

\( \chi^2(\text{obs}) \): A measure of how well the observed proportion of ones within a given \( M \)-bit block match the expected proportion (1/2).

The reference distribution for the test statistic is a \( \chi^2 \) distribution.

2.1.4 Test Description

(1) Partition the input sequence into \( N = \left\lfloor \frac{n}{M} \right\rfloor \) non-overlapping blocks. Discard any unused bits.

(2) Determine the proportion \( \pi_i \) of ones in each \( M \)-bit block using the equation

\[
\pi_i = \frac{\sum_{j=i}^{i+M-1} \varepsilon_j}{M}, \text{ for } 1 \leq i \leq N.
\]

(3) Compute the \( \chi^2 \) statistic: \( \chi^2(\text{obs}) = 4M \sum_{i=1}^{N} (\pi_i - 1/2)^2 \).

(4) Compute \( P\)-value = igamc (N/2, \( \chi^2(\text{obs})/2 \)), where \( \text{igamc} \) is the incomplete gamma function for \( Q(a,x) \) as defined below

Gamma Function

\[
\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt
\]

Incomplete Gamma Function

\[
P(a,x) = \frac{\gamma(a,x)}{\Gamma(a)} = \frac{1}{\Gamma(a)} \int_0^x t^{a-1}e^{-t}dt
\]

where \( P(a,0) = 0 \) and \( P(a,\infty) = 1 \)

Incomplete Gamma Function

\[
Q(a,x) = 1 - P(a,x) = \frac{\Gamma(a,x)}{\Gamma(a)} = \frac{1}{\Gamma(a)} \int_x^\infty t^{a-1}e^{-t}dt
\]

Where \( Q(a,0) = 1 \) and \( Q(a,\infty) = 0 \).

2.2 Runs Test

2.2.1 Test Purpose
The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. A run of length \( k \) consists of exactly \( k \) identical bits and is bounded before and after with a bit of the opposite value. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow.

### 2.2.2 Function Call

**Runs** \((n)\), where,

- **\( n \)** The length of the bit string.

Additional inputs for the function, but supplied by the testing code,

- **\( \varepsilon \)** The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; \( \varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \).

### 2.2.3 Test Statistic and Reference Distribution

**\( V_n^{\text{obs}} \):** The total number of runs (i.e., the total number of zero runs + the total number of one-runs) across all \( n \) bits.

The reference distribution for the test statistic is a \( \chi^2 \) distribution.

### 2.2.4 Test Description

Note: The Runs test carries out a Frequency test as a prerequisite.

1. Compute the pre-test proportion \( \pi \) of ones in the input sequence: \( \pi = \frac{\sum_{i=1}^{n} \varepsilon_i}{n} \).
2. Determine if the prerequisite Frequency test is passed: If it can be shown that \( |\pi - \frac{1}{2}| \geq r \), then the Runs test need not be performed i.e., the test should not have been run because of a failure to pass test, the Frequency (monobit) test. If the test is not applicable, then the \( P\text{-value} \) is set to 0.0000. Note that for this test, \( r = \frac{2}{\sqrt{n}} \) has been pre-defined in the test code.
3. Compute the test statistic \( V_n^{\text{obs}} = \sum_{k=1}^{n} r(k) + 1 \), where \( r(k) = 0 \) if \( \varepsilon_k = \varepsilon_{k+1} \), and \( r(k) = 1 \) otherwise.
4. Compute \( P\text{-value} = \text{erfc}\left(\frac{V_n^{\text{obs}} - 2n\pi(1-\pi)}{2\sqrt{2n\pi(1-\pi)}}\right) \).

### 2.3 Test for the Longest Run of Ones in a Block
2.3.1 Test Purpose
The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence. Note that an irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes. Therefore, only a test for ones is necessary.

2.3.2 Function Call
Longest Run Of Ones (n), where,
\[ n \] The length of the bit string.

Additional input for the function supplied by the testing code,
\[ \varepsilon \] The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; \[ \varepsilon = \varepsilon_1, \varepsilon_2, ..., \varepsilon_n \].
\[ M \] The length of each block. The test code has been pre-set to accommodate three values for \( M \): \( M = 8, M = 128 \) and \( M = 10^4 \) in accordance with the following table.

<table>
<thead>
<tr>
<th>Minimum n</th>
<th>( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>8</td>
</tr>
<tr>
<td>6272</td>
<td>128</td>
</tr>
<tr>
<td>750,000</td>
<td>( 10^4 )</td>
</tr>
</tbody>
</table>

\[ N \] The number of blocks; selected in accordance with the value of \( M \).

2.3.3 Test Statistic and Reference Distribution
\( \chi^2(\text{obs}) \): A measure of how well the observed longest run length within M-bit blocks matches the expected longest length within M-bit blocks.

The reference distribution for the test statistic is a \( \chi^2 \) distribution.

2.3.4 Test Description
(1) Divide the sequence into \( M \)-bit blocks.
(2) Tabulate the frequencies \( v_i \) of the longest runs of ones in each block into categories, where each cell contains the number of runs of ones of a given length.

For the values of \( M \) supported by the test code, the \( v_i \) cells will hold the following counts,
(3) Compute $\chi^2(obs) = \sum_{i=0}^{k} \frac{(v_i - N\pi_i)^2}{N\pi_i}$.

The values of $K$ and $N$ are determined by the value of $M$ in accordance with the following table:

<table>
<thead>
<tr>
<th>$M$</th>
<th>$K$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>128</td>
<td>5</td>
<td>49</td>
</tr>
<tr>
<td>$10^4$</td>
<td>6</td>
<td>75</td>
</tr>
</tbody>
</table>

(4) Compute $P$-value $= \text{igamc} \left( \frac{k}{2}, \frac{\chi^2(obs)}{2} \right)$.  

2.4 Binary Matrix Rank Test

2.4.1 Test Purpose

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

2.4.2 Function Call

Rank ($n$), where,

$n$ The length of the bit string.

Additional input used by the function supplied by the testing code,

$\varepsilon$ The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$.

$M$ The number of rows in each matrix. For the test suite, $M$ has been set to 32. If other values of $M$ are used, new approximations need to be computed.
The number of columns in each matrix. For the test suite, \( Q \) has been set to 32. If other values of \( Q \) are used, new approximations need to be computed.

2.4.3 Test Statistic and Reference Distribution

\( \chi^2(\text{obs}) \): A measure of how well the observed number of ranks of various orders match the expected number of ranks under an assumption of randomness.

The reference distribution for the test statistic is a \( \chi^2 \) distribution.

2.4.4 Test Description

(1) Sequentially divide the sequence into \( M \cdot Q \)-bit disjoint blocks; there will exist

\[
N = \left\lfloor \frac{n}{M \cdot Q} \right\rfloor
\]

such blocks. Discarded bits will be reported as not being used in the computation within each block. Collect the \( M \cdot Q \) bit segments into \( M \) by \( Q \) matrices. Each row of the matrix is filled with successive \( Q \)-bit blocks of the original sequence \( e \).

(2) Determine the binary rank \( R(\ell) \) of each matrix, where \( \ell = 1, \ldots, N \).

(3) Let \( F_M = \) the number of matrices with \( R(\ell) = M \) (full rank), \( F_{M-1} = \) the number of matrices with \( R(\ell) = M-1 \) (full rank - 1), \( N - F_M - F_{M-1} = \) the number of matrices remaining.

(4) Compute

\[
\chi^2(\text{obs}) = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}
\]

(5) Compute \( P \text{-value} = e^{-\chi^2(\text{obs})/2} \).

2.5 Discrete Fourier Transform (Spectral) Test

2.5.1 Test Purpose

The focus of this test is the peak heights in the discrete Fourier transform of the sequence. The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness. The intention is to detect whether the number of peaks exceeding the 95% threshold is significantly different than 5%.

2.5.2 Function Call

Discrete Fourier Transform \((n)\), where,

\[
n \quad \text{The length of the bit string.}
\]

Additional input used by the function, but supplied by the testing code,
2.5.3 Test Statistic and Reference Distribution

d: The normalized difference between the observed and the expected number of frequency components that are beyond the 95% threshold.

The reference distribution for the test statistic is the normal distribution.

2.5.4 Test Description

(1) The zeros and ones of the input sequence (ε) are converted to values of -1 and +1 to create the sequence \( X = x_1, x_2, \ldots, x_n \), where \( x_i = 2\epsilon_i - 1 \).

(2) Apply a Discrete Fourier transform (DFT) on \( X \) to produce: \( S = DFT(X) \). A sequence of complex variables is produced which represents periodic components of the sequence of bits at different frequencies.

(3) Calculate \( M = \text{modulus}(S') = |S'| \), where \( S' \) is the substring consisting of the first \( n/2 \) elements in \( S \), and the modulus function produces a sequence of peak heights.

(4) Compute \( T = \sqrt{3n} \) = the 95% peak height threshold value. Under an assumption of randomness, 95% of the values obtained from the test should not exceed \( T \).

(5) Compute \( N_0 = 0.95n/2 \). \( N_0 \) is the expected theoretical (95%) number of peaks (under the assumption of randomness) that are less than \( T \).

(6) Compute \( N_1 \) = the actual observed number of peaks in \( M \) that are less than \( T \).

(7) Compute \( d = \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/2}} \).

(8) Compute \( P\)-value = \( \text{erfc} \left( \frac{|d|}{\sqrt{2}} \right) \).

2.6 Non-overlapping Template Matching Test

2.6.1 Test Purpose

The focus of this test is the number of occurrences of pre-specified target strings. The purpose of this test is to detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern.

2.6.2 Function Call

Non overlapping template matching \((m, n)\)

\( m \) The length in bits of each template. The template is the target string.

\( n \) The length of the entire bit string under test.
Additional input used by the function, but supplied by the testing code,

\( \varepsilon \)  The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call;  
\( \varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \).

\( B \)  The \( m \)-bit template to be matched; \( B \) is a string of ones and zeros (of length \( m \)) which is defined in a template library of non-periodic patterns contained within the test code.

\( M \)  The length in bits of the substring of \( \varepsilon \) to be tested.

\( N \)  The number of independent blocks.

### 2.6.3 Test Statistic and Reference Distribution

\( \chi^2(\text{obs}) \): A measure of how well the observed number of template “hits” matches the expected number of template “hits” (under an assumption of randomness).

The reference distribution for the test statistic is the \( \chi^2 \) distribution.

### 2.6.4 Test Description

1. Partition the sequence into \( N \) independent blocks of length \( M \).
2. Let \( W_j (j = 1, \ldots, N) \) be the number of times that \( B \) (the template) occurs within the block \( j \). The search for matches proceeds by creating an \( m \)-bit window on the sequence, comparing the bits within that window against the template. If there is no match, the window slides over one bit, e.g., if \( m = 3 \) and the current window contains bits 3 to 5, then the next window will contain bits 4 to 6. If there is a match, the window slides over \( m \) bits, e.g., if the current (successful) window contains bits 3 to 5, then the next window will contain bits 6 to 8.
3. Under an assumption of randomness, compute the theoretical mean \( \mu \) and variance \( \sigma^2 \):

\[
\mu = \frac{(M - m + 1)}{2^m} \\
\sigma^2 = M \left( \frac{1}{2^m} - \frac{2m - 1}{2^{2m}} \right)
\]

4. Compute \( \chi^2(\text{obs}) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2} \).

5. Compute \( P\text{-value} = \text{igamc} \left( \frac{N}{2}, \frac{\chi^2(\text{obs})}{2} \right) \).
2.7 Overlapping Template Matching Test

2.7.1 Test Purpose
The focus of the overlapping template matching test is the number of occurrences of
prespecified target strings. Both this test and the non-overlapping template matching test
of Section 2.6 use an m-bit window to search for a specific m-bit pattern. As with the test
in Section 2.6, if the pattern is not found, the window slides one bit position. The
difference between this test and the test in Section 2.6 is that when the pattern is found,
the window slides only one bit before resuming the search.

2.7.2 Function Call
Overlapping template matching (m, n)
m The length in bits of the template – in this case, the length of the run of ones.
n The length of the bit string.
Additional input used by the function, but supplied by the testing code,
\( \varepsilon \) The sequence of bits as generated by the RNG or PRNG being tested; this exists
as a global structure at the time of the function call; \( \varepsilon = \varepsilon_1, \varepsilon_2, ..., \varepsilon_n \).
B The m-bit template to be matched.
K The number of degrees of freedom. K has been fixed at 5 in the test code.
M The length in bits of a substring of \( \varepsilon \) to be tested. M has been set to 1032 in the
test code.
N The number of independent blocks of n. N has been set to 968 in the test code.

2.7.3 Test Statistic and Reference Distribution
\( \chi^2(\text{obs}) \): A measure of how well the observed number of template “hits” matches
the expected number of template “hits” (under an assumption of randomness).

The reference distribution for the test statistic is the \( \chi^2 \) distribution.
(1) Partition the sequence into \( N \) independent blocks of length \( M \).
(2) Calculate the number of occurrences of \( B \) in each of the \( N \) blocks. The search for
matches proceeds by creating an \( m \)-bit window on the sequence, comparing the bits
within that window against \( B \) and incrementing a counter when there is a match. The
window slides one bit after each examination, e.g., if \( m = 4 \) and the first window
contains bits 42 to 45, the next window consists of bits 43 to 46. Record the number
of occurrences of \( B \) in each block by incrementing an array \( v_i \) (where \( i = 0, ..., 5 \)), such
that $v_0$ is incremented when there are no occurrences of $B$ in a substring, $v_1$ is incremented for one occurrence of $B$, ... and $v_5$ is incremented for 5 or more occurrences of $B$.

(3) Compute values for $\lambda$ and $\eta$ that will be used to compute the theoretical probabilities $\pi$, corresponding to the classes of $v_i$

$$\lambda = (M - m + 1)/2^n \quad \text{and} \quad \eta = \lambda/2.$$ 

(4) Compute $\chi^2(obs) = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i}$.

(5) Compute $P-value = igamc\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right)$.

2.8 Lempel-Ziv Compression Test

2.8.1 Test Purpose

The focus of this test is the number of cumulatively distinct patterns (words) in the sequence. The purpose of the test is to determine how far the tested sequence can be compressed. The sequence is considered to be non-random if it can be significantly compressed. A random sequence will have a characteristic number of distinct patterns.

2.8.2 Function Call

Lempel-Ziv compression ($n$), where,

- $n$ The length of the bit string.
- $\varepsilon$ The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; $\varepsilon = \varepsilon_1, \varepsilon_2, ..., \varepsilon_n$.

2.8.3 Test Statistic and Reference Distribution

$W_{obs}$: The number of disjoint and cumulatively distinct words in the sequence.

The reference distribution for the test statistic is the normal distribution.

2.8.4 Test Description

(1) Resolve the sequence into consecutive, disjoint and distinct words that will form a “dictionary” of words in the sequence. This is accomplished by creating substrings from consecutive bits of the sequence until a substring is created that has not been found previously in the sequence. The resulting substring is a new word in the dictionary.
<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Bit</th>
<th>New Word?</th>
<th>The Word is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Yes</td>
<td>0 (Bit 1)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Yes</td>
<td>1 (Bit 2)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Yes</td>
<td>01 (Bits 3-4)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>Yes</td>
<td>10 (Bits 5-6)</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>Yes</td>
<td>010 (Bits 7-9)</td>
</tr>
</tbody>
</table>

There are five words in the "dictionary": 0, 1, 01, 10, 010. Hence, $W_{obs} = 5$.

(2) Compute $P-value = \frac{1}{2} \text{erfc} \left( \frac{\mu - W_{obs}}{\sqrt{2\sigma^2}} \right)$.

### 2.9 Linear Complexity Test

#### 2.9.1 Test Purpose

The focus of this test is the length of a linear feedback shift register (LFSR). The purpose of this test is to determine whether or not the sequence is complex enough to be considered random. Random sequences are characterized by longer LFSRs. An LFSR that is too short implies nonrandomness.

#### 2.9.2 Function Call

Linear complexity $(M, n)$, where,

- $M$ The length in bits of a block.
- $n$ The length of the bit string.

Additional input used by the function, but supplied by the testing code,

- $\varepsilon$ The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$.
- $K$ The number of degrees of freedom; $K = 6$ has been hard coded into the test.

#### 2.9.3 Test Statistic and Reference Distribution

$\chi^2 (obs)$: A measure of how well the observed number of occurrences of fixed length LFSRs matches the expected number of occurrences under an assumption of randomness.
The reference distribution for the test statistic is the $\chi^2$ distribution.

### 2.9.4 Test Description

1. Partition the $n$-bit sequence into $N$ independent blocks of $M$ bits, where $n = MN$.
2. Using the Berlekamp-Massey algorithms, determine the linear complexity $L_i$ of each block ($i = 1, \ldots, N$). $L_i$ is the length of the shortest linear feedback shift register sequence that generates all bits in the block $i$. Within any $L_i$-bit sequence, some combination of the bits, when added together modulo-2, produces the next bit in the sequence (bit $L_i + 1$).
3. Under an assumption of randomness, calculate the theoretical mean $\mu$:
   \[
   \mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{(M/2 + 2/9)}{2^M}.
   \]
4. For each substring, calculate a value of $T_i$, where $T_i = (-1)^M \cdot (L_i - \mu) + 2/9$.
5. Record the $T_i$ values in $\nu_0, \ldots, \nu_6$ as follows:
   - If $T_i \leq -2.5$, increment $\nu_0$ by one.
   - If $-2.5 < T_i \leq -1.5$, increment $\nu_1$ by one.
   - If $-1.5 < T_i \leq -0.5$, increment $\nu_2$ by one.
   - If $-0.5 < T_i \leq 0.5$, increment $\nu_3$ by one.
   - If $0.5 < T_i \leq 1.5$, increment $\nu_4$ by one.
   - If $1.5 < T_i \leq 2.5$, increment $\nu_5$ by one.
   - If $T_i > 2.5$, increment $\nu_6$ by one.
6. $\chi^2(obs) = \sum_{i=0}^{K} \frac{(\nu_i - N\pi_i)^2}{N\pi_i}$.
7. Compute $P-value = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$.

### 2.10 Cumulative Sums (Cusum) Test

#### 2.10.1 Test Purpose

The focus of this test is the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative...
sum for random sequences. This cumulative sum may be considered as a random walk. For a random sequence, the excursions of the random walk should be near zero. For certain types of non-random sequences, the excursions of this random walk from zero will be large.

2.10.2 Function Call
Cumulative sums \((mode, n)\), where,

\(n\)  The length of the bit string.

Additional input for the function, but supplied by the testing code:

\(e\)  The sequence of bits as generated by the RNG or PRNG being tested; this exists as a global structure at the time of the function call; \(e = e_1, e_2, ..., e_n\).

\(mode\)  A switch for applying the test either forward through the input sequence \((mode = 0)\) or backward through the sequence \((mode = 1)\).

2.10.3 Test Statistic and Reference Distribution
\(z\): The largest excursion from the origin of the cumulative sums in the corresponding \((-1,+1)\) sequence.

The reference distribution for the test statistic is the normal distribution.

2.10.4 Test Description
(1) Form a normalized sequence: The zeros and ones of the input sequence \((e)\) are converted to values \(X_i\) of \(-1\) and \(+1\) using \(X_i = 2e_i - 1\).

(2) Compute partial sums \(S_i\) of successively larger subsequences, each starting with \(X_i\) (if \(mode = 0\)) or \(X_n\) (if \(mode = 1\)).

<table>
<thead>
<tr>
<th>Mode = 0 (forward)</th>
<th>Mode = 1 (backward)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1 = X_1)</td>
<td>(S_1 = X_n)</td>
</tr>
<tr>
<td>(S_2 = X_1 + X_2)</td>
<td>(S_2 = X_n + X_{n-1})</td>
</tr>
<tr>
<td>(S_3 = X_1 + X_2 + X_3)</td>
<td>(S_3 = X_n + X_{n-1} + X_{n-2})</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>(S_k = X_1 + X_2 + X_3 + ... + X_k)</td>
<td>(S_k = X_n + X_{n-1} + X_{n-2} + ... + X_{n-k+1})</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>(S_n = X_1 + X_2 + X_3 + ... + X_n)</td>
<td>(S_n = X_n + X_{n-1} + X_{n-2} + ... + X_{k-1} + ... + X_1)</td>
</tr>
</tbody>
</table>

That is, \(S_k = S_{k-1} + X_k\) for mode 0, and \(S_k = S_{k-1} + X_{n-k+1}\) for mode 1.
(3) Compute the test statistic \( z = \max_{1 \leq k \leq n} |S_k| \), where \( \max_{1 \leq k \leq n} |S_k| \) is the largest of the absolute values of the partial sums \( S_k \).

(4) Compute \( P-value = 1 - \sum_{k=0}^{n-1} \left[ \phi\left( \frac{(4k+1)z}{\sqrt{n}} \right) - \phi\left( \frac{(4k-1)z}{\sqrt{n}} \right) \right] + \sum_{k=0}^{n-3} \left[ \phi\left( \frac{(4k+3)z}{\sqrt{n}} \right) - \phi\left( \frac{(4k+1)z}{\sqrt{n}} \right) \right] \)

Where \( \phi \) is the Standard Normal Cumulative Probability Distribution Function given by

\[ \phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2} \, du \]

3.0 Randomness Test Result

Randomness tests were performed on the output sequences obtained from the following:

- LFSR based complex code generator
- PLA based complex code generator
- Multichannel stream cipher (Technique 1)
  - (i) Keystream sequence
  - (ii) Encrypted message sequence
- Multichannel stream cipher (Technique 2)
  - (i) Keystream sequence
  - (ii) Encrypted message sequence

For every randomness test P-value is computed. Then these P-values are compared to \( \alpha \) in order to decide, whether a test is success or failure, where \( \alpha \in [0.0001, 0.01] \). Success is declared, whenever P value \( \geq \alpha \); otherwise, failure is declared. Randomness tests, which were passed successfully, are given below.

3.0.1 LFSR based Complex Code Generator

Frequency Test

----------------------------------------
COMPUTATIONAL INFORMATION:
----------------------------------------
(a) The nth partial sum = -23
(b) \( S_{n/n} = -0.004599 \)
SUCCESS  \( P\_\text{value} = 0.745002 \)

**Block Frequency Test**

\[ \chi^2 = 39.187500 \]

(b) Number of substrings = 39

(c) Block length = 128

\( \text{(d) Note: 9 bits were discarded.} \)

SUCCESS  \( P\_\text{value} = 0.461466 \)

**Runs Test**

\[ n = 0.497700 \]

(b) \( V_r(\text{obs}) \) (Total number of runs) = 2505

(c) \( V_r(\text{obs}) - 2 n \pi (1-\pi) \)

\[ \frac{0.091051}{2 \sqrt{2n} \pi (1-\pi)} \]

SUCCESS  \( P\_\text{value} = 0.897544 \)

**Longest Runs of Ones Test**

\[ N = 0 \]

(b) \( M \) (Substring Length) = 10000

(c) \( \chi^2 \) = -1. # IND00

\[ \text{FREQUENCY} \]

112
<10 11 12 13 14 15 >=16 P-value Assignment
 0 0 0 0 0 0 0 0 0 0 1.000000 SUCCESS

Rank Test

COMPUTATIONAL INFORMATION:

(a) Probability P_{32} = 0.288788
(b) P_{31} = 0.577576
(c) P_{30} = 0.133636
(d) Frequency F_{32} = 1
(e) F_{31} = 3
(f) F_{30} = 0
(g) Number of matrices= 4
(h) \chi^2 = 0.761277
(i) NOTE: 905 BITS WERE DISCARDED.

SUCCESS P_value = 0.683425

Nonperiodic Templates Test

COMPUTATIONAL INFORMATION

LAMBDA = 0.601563 M = 625 N = 8 m = 10 n = 5001

FREQUENCY

<table>
<thead>
<tr>
<th>Template</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
<th>W4</th>
<th>W5</th>
<th>W6</th>
<th>W7</th>
<th>W8</th>
<th>\chi^2</th>
<th>P_value</th>
<th>Assignment</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001011111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.832872</td>
<td>0.775281</td>
<td>SUCCESS</td>
<td>44</td>
</tr>
<tr>
<td>0001100101</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3.815597</td>
<td>0.873366</td>
<td>SUCCESS</td>
<td>45</td>
</tr>
<tr>
<td>0001100111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.815597</td>
<td>0.873366</td>
<td>SUCCESS</td>
<td>46</td>
</tr>
<tr>
<td>0001101001</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.815597</td>
<td>0.873366</td>
<td>SUCCESS</td>
<td>47</td>
</tr>
<tr>
<td>0001110101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4.154689</td>
<td>0.842904</td>
<td>SUCCESS</td>
<td>48</td>
</tr>
<tr>
<td>0001110111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>9.475821</td>
<td>0.303758</td>
<td>SUCCESS</td>
<td>49</td>
</tr>
<tr>
<td>0001110111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2.798322</td>
<td>0.946369</td>
<td>SUCCESS</td>
<td>50</td>
</tr>
</tbody>
</table>

113
Overlapping Template of All Ones Test

COMPUTATIONAL INFORMATION:

(a) $n$ (sequence_length) = 5001
(b) $m$ (block length of 1s) = 10
(c) $M$ (length of substring) = 1032
(d) $N$ (number of substrings) = 4
(e) $\lambda = \frac{(M-m+1)}{2^m} = 0.999023$
(f) $\eta = 0.499512$

FREQUENCY

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>&gt;=5</th>
<th>$\chi^2$</th>
<th>P-value</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.591666</td>
<td>0.762631</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

Cumulative Sums (FORWARD) Test

COMPUTATIONAL INFORMATION:

(a) The maximum partial sum = 35

SUCCESS P_value = 0.991729

Cumulative Sums (REVERSE) Test

COMPUTATIONAL INFORMATION:

(a) The maximum partial sum = 58

SUCCESS P_value = 0.796580

3.0.2 PLA based Complex Code Generator

In order to ascertain the randomness of the sequence generated with the help of the PLA based complex generator circuit, statistical randomness tests (such as runs, rank, Lempel-
Ziv compression, frequency, cumulative sums and block frequency) were also applied to the generated sequence, which were passed successfully.

3.0.3 Multichannel Stream Cipher (Technique 1)
Randomness tests were performed on the keystreams and encrypted data generated in multichannel stream cipher (Technique 1) where Gold codes and LFSRs were used for the generation of the keystream (section 5.5.1).

(i) Keystream Sequence

**Longest Runs of Ones Test**

<table>
<thead>
<tr>
<th>&lt;=10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>&gt;=16</th>
<th>P-value</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.000000</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

**Rank Test**

<table>
<thead>
<tr>
<th>&lt;=10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>&gt;=16</th>
<th>P-value</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.000000</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

COMPUTATIONAL INFORMATION:

(a) N (number of substrings) = 0
(b) M (substring length) = 10000
(c) $\chi^2$ = -1. # INDO0

---

COMPUTATIONAL INFORMATION:

(a) Probability $P_{32} = 0.288788$
(b) $P_{31} = 0.577576$
(c) $P_{30} = 0.133636$
(d) Frequency $F_{32} = 0$
(e) $F_{31} = 0$
(f) $F_{30} = 1$
(g) Number of matrices = 1
(h) $\chi^1 = 6.483030$
(i) NOTE: 706 BITS WERE DISCARDED.
SUCCESS \( P_{\text{value}} = 0.039105 \)

**FFT TEST**

-----------------------------------------------------------------

**COMPUTATIONAL INFORMATION:**

-----------------------------------------------------------------

(a) Percentile = 93.757225  
(b) \( N_1 = 811.000000 \)  
(c) \( N_0 = 821.750000 \)  
(d) \( d = -1.677079 \)

-----------------------------------------------------------------

SUCCESS \( P_{\text{value}} = 0.093527 \)

**Nonperiodic Templates Test**

-----------------------------------------------------------------

**COMPUTATIONAL INFORMATION**

-----------------------------------------------------------------

\[ \lambda = 0.406250 \quad M = 216 \quad N = 8 \quad m = 9 \quad n = 1730 \]

-----------------------------------------------------------------

**FREQUENCY**

<table>
<thead>
<tr>
<th>Template</th>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( W_3 )</th>
<th>( W_4 )</th>
<th>( W_5 )</th>
<th>( W_6 )</th>
<th>( W_7 )</th>
<th>( W_8 )</th>
<th>( \chi^2 )</th>
<th>( P_{\text{value}} )</th>
<th>Assignment Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>0011010101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00110110101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00110111101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00111010101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00111011101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00111101011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>00111110101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.237112</td>
<td>0.918608</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

**Overlapping Template of All Ones Test**

-----------------------------------------------------------------

**COMPUTATIONAL INFORMATION:**

-----------------------------------------------------------------

(a) \( n \) (sequence_length) = 1730  
(b) \( m \) (block length of 1s) = 9

116
(c) M (length of substring) = 1032
(d) N (number of substrings) = 1
(e) $\lambda = [(M-m+1)/2^m] = 2.000000$
(f) $\eta = 1.000000$

------------------------------------
F R E Q U E N C Y

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>&gt;=5</th>
<th>$\chi^2$</th>
<th>P-value</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6.109513</td>
<td>0.295708</td>
<td>SUCCESS</td>
<td></td>
</tr>
</tbody>
</table>

**Lempel-Ziv Compression Test**

----------------------------------------
COMPUTATIONAL INFORMATION:
----------------------------------------
(a) W (number of words) = 221
(b) Bits Discarded = 1730

SUCCESS P_value = 1.000000

(ii) Encrypted Message Sequence

**Frequency Test**

----------------------------------------
COMPUTATIONAL INFORMATION:
----------------------------------------
(a) The nth partial sum = 28
(b) $S_n/n = 0.016185$

SUCCESS P_value = 0.500829

**Block Frequency Test**

----------------------------------------
COMPUTATIONAL INFORMATION:
----------------------------------------
(a) $\chi^2 = 9.953757$
(b) Number of substrings = 10
(c) Block length = 173
SUCCESS

P_value = 0.444560

Longest Runs of Ones Test

COMPUTATIONAL INFORMATION:

(a) N (number of substrings) = 0
(b) M (substring length) = 10000
(c) \( \chi^2 \) = -1. # IND00

FREQUENCY

<=10  11  12  13  14  15 >=16  P-value Assignment
  0  0  0  0  0  0  0  1.000000  SUCCESS

Rank Test

COMPUTATIONAL INFORMATION:

(a) Probability \( P_{32} = 0.288788 \)
(b) \( P_{31} = 0.577576 \)
(c) \( P_{30} = 0.133636 \)
(d) Frequency \( F_{32} = 0 \)
(e) \( F_{31} = 0 \)
(f) \( F_{30} = 1 \)
(g) Number of matrices = 1
(h) \( \chi^2 \) = 6.483030
(i) NOTE: 706 BITS WERE DISCARD ED.

SUCCESS

P_value = 0.039105

Nonperiodic Templates Test

COMPUTATIONAL INFORMATION

118
\[ \lambda = 0.406250 \quad M = 216 \quad N = 8 \quad m = 9 \quad n = 1730 \]

**FREQUENCY**

Template \( W_1 \ W_2 \ W_3 \ W_4 \ W_5 \ W_6 \ W_7 \ W_8 \ \chi^2 \) P-value Assignment Index

| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 44 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 45 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 46 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 47 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 48 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 49 |
| 001100110 | 0 0 0 0 0 0 0 0 | 3.237112 0.918608 SUCCESS 50 |

**Overlapping Template of All Ones Test**

**COMPUTATIONAL INFORMATION:**

(a) \( n (\text{sequence length}) = 1730 \)

(b) \( m (\text{block length of 1s}) = 9 \)

(c) \( M (\text{length of substring}) = 1032 \)

(d) \( N (\text{number of substrings}) = 1 \)

(e) \( \lambda = \frac{(M-m+1)}{2^m} = 2.000000 \)

(f) \( \eta = 1.000000 \)

**FREQUENCY**

| 0 1 2 3 4 >=5 \\( \chi^2 \) P-value Assignment |
| 0 0 0 0 0 | 1 6.109513 0.295708 SUCCESS |

**Lempel-Ziv Compression Test**

**COMPUTATIONAL INFORMATION:**
(a) \( W \) (number of words) = 224
(b) Bits Discarded = 1730

SUCCESS \( P \text{\_value} = 1.000000 \)

**Cumulative Sums (FORWARD) Test**

**COMPUTATIONAL INFORMATION:**

(a) The maximum partial sum = 51

SUCCESS \( P \text{\_value} = 0.439808 \)

**Cumulative Sums (REVERSE) Test**

**COMPUTATIONAL INFORMATION:**

(a) The maximum partial sum = 61

SUCCESS \( P \text{\_value} = 0.284957 \)

3.0.4 Multichannel Stream Cipher (Technique 2)

Randomness tests were performed on the keystreams and encrypted data in multichannel stream cipher, where modified codes were used for the generation of the keystream.

(i) **Keystream Sequence**

**Longest Runs of Ones Test**

**COMPUTATIONAL INFORMATION:**

(a) \( N \) (number of substrings) = 0
(b) \( M \) (substring length) = 10000
(c) \( \chi' \) = -1. # IND00

**FREQUENCY**

---

120
Nonperiodic Templates Test

<table>
<thead>
<tr>
<th>Template</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
<th>W_5</th>
<th>W_6</th>
<th>W_7</th>
<th>W_8</th>
<th>χ^2</th>
<th>P_value</th>
<th>Assignment</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>001101011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>44</td>
</tr>
<tr>
<td>00110110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>45</td>
</tr>
<tr>
<td>001101111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>46</td>
</tr>
<tr>
<td>001110101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>47</td>
</tr>
<tr>
<td>001110111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>48</td>
</tr>
<tr>
<td>001111011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>49</td>
</tr>
<tr>
<td>001111101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>50</td>
</tr>
</tbody>
</table>

Lempel-Ziv Compression Test

<table>
<thead>
<tr>
<th>Template</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
<th>W_5</th>
<th>W_6</th>
<th>W_7</th>
<th>W_8</th>
<th>χ^2</th>
<th>P_value</th>
<th>Assignment</th>
<th>Index</th>
</tr>
</thead>
</table>

SUCCESS P_value = 1.000000

(ii) Encrypted Message Sequence

Frequency Test

<table>
<thead>
<tr>
<th>Template</th>
<th>W_1</th>
<th>W_2</th>
<th>W_3</th>
<th>W_4</th>
<th>W_5</th>
<th>W_6</th>
<th>W_7</th>
<th>W_8</th>
<th>χ^2</th>
<th>P_value</th>
<th>Assignment</th>
<th>Index</th>
</tr>
</thead>
</table>

SUCCESS P_value = 1.000000

(a) W (number of words) = 580

(b) Bits Discarded = 8680
SUCCESS  \( P_{\text{value}} = 0.547790 \)

**Longest Runs of Ones Test**

COMPUTATIONAL INFORMATION:

(a) \( N \) (number of substrings) = 0
(b) \( M \) (substring length) = 10000
(c) \( \chi^2 \) = -1. # IND00

**FREQUENCY**

<table>
<thead>
<tr>
<th>&lt;=10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>&gt;=16</th>
<th>P-value</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.000000</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

**Nonperiodic Templates Test**

COMPUTATIONAL INFORMATION

\[ \lambda = 2.103516 \quad M = 1085 \quad N = 8 \quad m = 9 \quad n = 8680 \]

**FREQUENCY**

<table>
<thead>
<tr>
<th>Template</th>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( W_3 )</th>
<th>( W_4 )</th>
<th>( W_5 )</th>
<th>( W_6 )</th>
<th>( W_7 )</th>
<th>( W_8 )</th>
<th>( \chi^2 )</th>
<th>P-value</th>
<th>Assignment</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>00110101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>44</td>
</tr>
<tr>
<td>00110110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>45</td>
</tr>
<tr>
<td>00110111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>46</td>
</tr>
<tr>
<td>00111010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>47</td>
</tr>
<tr>
<td>00111011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>48</td>
</tr>
<tr>
<td>00111101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>49</td>
</tr>
<tr>
<td>00111110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.277721</td>
<td>0.027344</td>
<td>SUCCESS</td>
<td>50</td>
</tr>
</tbody>
</table>

**Lempel-Ziv Compression Test**
COMPUTATIONAL INFORMATION:

(a) W (number of words) = 796
(b) Bits Discarded = 8680

SUCCESS  \( P_{\text{value}} = 1.000000 \)

Cumulative Sums (FORWARD) Test

COMPUTATIONAL INFORMATION:

(a) The maximum partial sum = 139

SUCCESS  \( P_{\text{value}} = 0.271408 \)

Cumulative Sums (REVERSE) Test

COMPUTATIONAL INFORMATION:

(a) The maximum partial sum = 95

SUCCESS  \( P_{\text{value}} = 0.611321 \)

Note: In case of nonperiodic templates test \( P \)-value is computed for each possible template. Most of them give successful results. Few results are given here out of large number of template results.

Some tests were not applicable because of the limitations of the computing machine and hence the insufficient number of cycles. These tests are random excursions test, random excursions variant test, approximate entropy test and Maurer’s “universal statistical” test.
Reference

Appendix D

List of Publications of the Author

- **Farah Maqsood**, Arshad Ahmad and Wasim Ahmad
  “Scrambler for Increased Data Security and Low Bit Error Propagation,”

- **Farah Maqsood**, Omar Farooq, Wasim Ahmad and Shabbir Ahmad
  “LFSR based Complex Code Generator for Stream Cipher,”

- **Farah Maqsood**, Wasim Ahmad and Omar Farooq,
  “Random Selection of Feedback Tapings for LFSR based Complex Code Generator for Stream Cipher,”

Two Papers are communicated for the Publication

- **Farah Maqsood**, Omar Farooq and Wasim Ahmad, “LFSR based Line Encryption for Multichannel Secure Communication”.

- **Farah Maqsood**, Omar Farooq, and Wasim Ahmad, “LFSR and PLAs based Complex Code Generator for Stream Cipher”.

125