1 Randomness Tests by the NIST

The presented randomness tests are proposed by the NIST[1] [2]. All tests have a minimal requirement for the length of the sequence to be tested in order for the tests to give significant results. In what follows we give an overview and short description of each of these tests.[2]

2 Frequency (Monobit) Test

This test determines the proportion of the number of ones and zeros in a bit sequence. For a random sequence, one expects that the number of ones (and consequently zeros) in the sequence is close to $\frac{1}{2}$.

Consider the bit sequence $\varepsilon = \varepsilon_1, \ldots, \varepsilon_n$, where $\varepsilon_i = 0$ or $1, \forall i = 1, \ldots, n$ and n is the number of bits in the sequence. The tests transforms the sequence ε into a new sequence X, such that $X_i = 2\varepsilon_i - 1 = \pm 1$. The sum of this sequence is given by $S_n = X_1 + X_2 + \ldots + X_n$. The test statistic for the observed sum s_{obs} is given by

$$s_{obs} = \frac{|S_n|}{\sqrt{n}} \tag{1}$$

The reference distribution for the test statistic is half normal[3] for large n. If the sequence is random, then the plus and minus ones will tend to cancel each other out so that the test statistic is about 0. The P-value is then given by

$$P\text{-value} = \text{erfc}\left(\frac{s_{obs}}{\sqrt{2}}\right) \tag{2}$$

Where erfc(z) is the complementary error function

$$\text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \tag{3}$$

Figure 1 denotes the graphical representation of the complementary error function.

If the number of zeros and ones in the sequence tends to be the same, then S_n will be close to 0 and consequently the P-value will be close to 1.

The tested sequence will be accepted as random if the P-value ≥ 0.01, otherwise it is considered non-random. In order to give significant results the minimum number of bits in the sequence is expected to be 100.

[1] National Institute of Standards and Technology

[2] The presentation of the NIST tests below can be found in appendix of ref.[1], a work completed in the framework of the cryptasc project supported by the Impulse Programme of the Brussels Capital Region for the NICT sector (work package II: "Software for the quality control of random series", VUB 2007-2013).

[3] A half-normal distribution is a normal distribution with mean 0

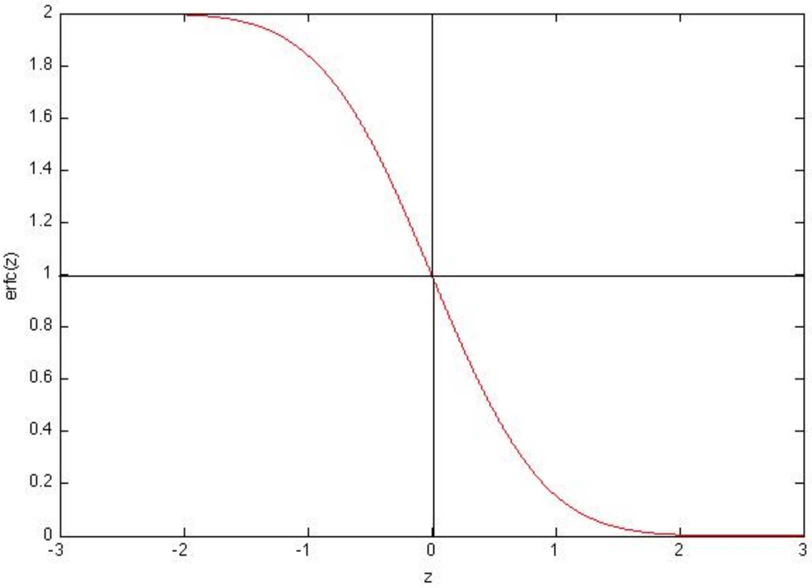

Figure 1: The complementary error function erfc(z)

3 Frequency Test within a Block

This test starts with dividing the sequence of length n in N blocks, each of length M ($N = \lfloor n/M \rfloor$). The purpose of this test is now to test the proportion of ones within each M bit block. The frequency of ones in every M bit block should be $M/2$ as would be expected for a random bit sequence.

Consider a bit sequence $\varepsilon = \varepsilon_1, \ldots, \varepsilon_n$. The tests determines the proportion of ones π_i in each M bit block using the equation

$$\pi_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)(M+j)}}{M}, \qquad \text{for } 1 \leq i \leq N \tag{4}$$

Using the computed proportions π_i, the χ^2 statistic is

$$\chi^2(obs) = 4M \sum_{i=1}^{N} (\pi_i - \frac{1}{2})^2 \tag{5}$$

A chi square χ^2 statistic is used to determine if a distribution of observed frequencies differs from the theoretical expected frequencies. The $(\pi_i - \frac{1}{2})^2$ are N independent, normally distributed proportions of random bits with mean 0. The χ^2 distribution has only one parementer N which determines its degrees of freedom, i.e. the number of $\pi_i' s$. It is a special case of the incomplete gamma distribution. Incomplete gamma functions are often met in statistics. For example, the cumulative distribution functions of the gamma distribution, Poisson distribution and chi-square distribution could be written by using the incomplete gamma function. The P-value is therefore computed using the incomplete gamma function.

$$P\text{-value} = igamc(\frac{N}{2}, \frac{\chi^2(obs)}{2}) \tag{6}$$

where *igamc* is the incomplete gamma function for $Q(a, x)$

$$Q(a,x) \equiv 1 - P(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t} t^{a-1} dt \tag{7}$$

Again a P-value ≥ 0.01 will let us conclude that the tested sequence is random. For this test to give accurate results, it is recommended that each sequence to be tested consists of a minimum of 100 bits. Note also that the block size M should be selected such that $M \geq 20$, $M \geq 0.1n$ and $N < 100$.

4 Runs Test

Let us define a *run* as an uninterrupted sequence of identical bits. A run of length k therefore consists of k identical bits bound before and after with a bit of opposite value. For example, the bit sequence 0011101011010111 consists of the runs 00, 111, 0, 1, 0, 11, 0, 1, 0, 111. The purpose of the runs tests is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. More concrete, the test determines whether the oscillation between zeros and ones is too fast or too slow.

The test starts with computing the pre-test proportion π of ones in the input sequence

$$\pi = \frac{\sum_i \varepsilon_i}{n} \tag{8}$$

Secondly, it is determined if the prerequisite frequency test is passed. If the frequency test has not been passed, the P-value of this test is set to 0 and consequently the runs test is not passed. If the frequency test has been passed, the test statistic $V_n(obs)$ is calculated.

$$V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1 \tag{9}$$

where $r(k) = 0$ if $\varepsilon_k = \varepsilon_{k+1}$ and $r(k) = 1$ otherwise Thereafter the P-value is computed

$$P\text{-value} = \text{erfc}\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) \tag{10}$$

It is recommended that the tested sequence consists of a minimum of 100 bits. P-values ≥ 0.01 will enable us to conclude that the bit sequence is random.

5 Test for Longest Run of Ones in a Block

As the name of the test indicates, it focusses on the longest run of ones within M bit blocks. This test determines whether the length of the longest run of ones would be in the same proportion as expected for a random sequence.

The test start by dividing the sequence into bit blocks of length M. The length M of the blocks, together with the number of degrees of freedom K and the number of block N is determined by the length of the sequence according to table 1.

Minimum length n	M	K	N
128	8	3	16
6272	128	5	49
750000	10^4	6	75

Table 1: The blocklength M of each block according to the minimum length n of the tested sequence.

Thereafter, the frequencies ν_i of the longest runs of ones in each block is tabulated in categories in table 2, where each cell contains the number of ones of a given length. There are 3 values of M supported in the code.

ν_i	$M = 8$	$M = 128$	M $= 10^4$
ν_0	≤ 1	≤ 4	≤ 10
ν_1	2	5	11
ν_2	3	6	12
ν_3	≥ 4	7	13
ν_4		8	14
ν_5		≥ 9	15
ν_6			≥ 16

Table 2: The frequencies ν_i of longest runs of ones in each block

Afterwards the χ^2 statistic is computed

$$\chi^2(obs) = \sum_{i=0}^{K} \frac{(\nu_i - N\pi_i)^2}{N\pi_i} \tag{11}$$

where the values π_i are provided in section 3.4 of the NIST document [3]. The values K and N are determined by the value of M in accordance with table 1. Finally the P-value is computed

$$P\text{-value} = \text{igamc}\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) \tag{12}$$

P-values above 0.01 allow us to accept the null hypothesis, i.e. the sequence is accepted to be random.

6 Binary Matrix Rank Test

This test checks correlations or linear dependence among fixed length substrings of the original sequence, i.e. a matrix is created from each substring and statistical analysis is done by looking at the ranks of those matrices.

Initially, the bit sequence is divided in M, Q disjoint blocks $N = \lfloor \frac{n}{MQ} \rfloor$, where M denotes the number of rows in each matrix and Q denotes the number of columns in each matrix. Subsequently, N matrices of dimension $M.Q$ are created.

Thereafter the rank R_l of each matrix is determined, with $l = 1, \ldots, N$ and each matrix is placed in one of three different sets:

> F_M be the number of matrices with full rank, i.e. $R_l = M$.
>
> F_{M-1} be the number of matrices with full rank -1, i.e. $R_l = M - 1$
>
> $N - F_M - F_{M-1}$ all the remaining matrices.

The $\chi^2(obs)$ statistic is then computed as follows

$$\begin{aligned} \chi^2(obs) &= \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} \\ &\quad + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N} \end{aligned} \tag{13}$$

Afterwards, the P-value is computed

$$P\text{-value} = e^{-\chi^2(obs)/2} \tag{14}$$

The minimum number of bits to be tested should be such that $n \geq 38MQ$, i.e. at least 38 matrices are created. In equation (13), the probabilities for M = Q = 32 have been inserted into the code. Other values for M and Q may be chosen, although other probability values would need to be calculated. If the resulting P-value of the test ≥ 0.01, the null hypothesis is accepted.

7 Discrete Fourier Transform Test or Spectral Test

The Fourier test detects short term periodic features in the bit series that would indicate a deviation from the assumption of randomness.

The test starts by converting the bit sequence ε in values of -1 and $+1$, just as in with the frequency test; $X = x_1, \ldots, x_n$, where $x_i = 2\varepsilon_i - 1$. Thereafter, the discrete Fourier transform (DFT) is applied on X to produce $S = DFT(X)$. Now we are left with a sequence of complex variables which represent periodic components of the sequence of bits. Figures 2 and 3 show the graphical DFT analysis of some sampled sequences.

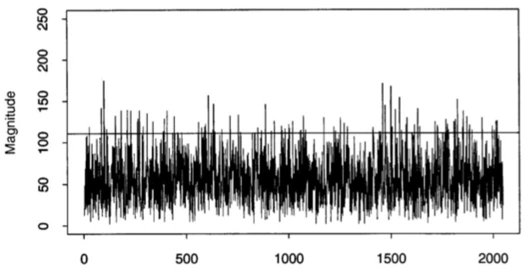

Figure 2: Example where DFT analysis shows no evidence of periodic patterns in a sampled sequence.

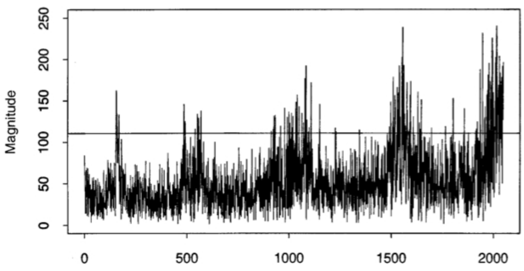

Figure 3: Example where DFT analysis shows clear evidence of periodic patterns in a sampled sequence.

The test continues by calculating the modulus of only half of the DFT converted sequence; $M = modulus(S')$ with S' is the substring of the $n/2$ first elements in S.

Furthermore, the 95% peak threshold value $T = \sqrt{3n}$ is determined. This means that under assumption of randomness 95% of the values obtained should not exceed T. $N_0 = 0.95n/2$, the theoretically expected number of peaks less than T and N_1 the actual observed number of peaks M that are less than T are now worked out.

In order to keep the calculation of the P-value synoptic, an variable d in introduced

$$d = \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/2}} \tag{15}$$

which allows to write the P-value as

$$P\text{-value} = \text{erfc}\left(\frac{|d|}{\sqrt{2}}\right) \tag{16}$$

It is recommended that the sequences to be tested by this test consists at least of 1000 bits. The user can then accurately conclude that the sequence is considered random by this test if the P-value ≥ 0.01.

8 Non-Overlapping Template Matching Test

It is possible for a random generator to produce too many occurrences of a given non-periodic pattern. In this test as in the overlapping template matching test of section 9, the program looks for a pre-specified target (template) string in an M bit block. If the pattern is not found, the template slides one bit position in the M bit block. If the patter is found, the template slides to the first bit after the found pattern.

The test start by dividing the bit sequence into N independent blocks of length M. A variable W_j ($j = 1, \ldots, N$), the number of times that the template B occurs within the block j, is introduced. The search for matches proceeds by creating an m bit window against the template in each bit block j. If there is no match, the m bit window slides over one bit. If there is match, the window slides over m bits and W_j is incremented by one.

Under assumption of randomness, the theoretical mean μ and variance σ^2 are computed as follows:

$$\mu = \frac{(M - m + 1)}{2^m} \qquad \sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right) \tag{17}$$

Now the $\chi^2(obs)$ statistic is computed

$$\chi^2(obs) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2} \tag{18}$$

and consequently, the P-value is

$$P\text{-value} = \text{igamc}\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \tag{19}$$

The test code has been written to assume a sequence length of $n = 10^6$. It is recommended that the size of the template has length 9 or 10, and that the number of M bit blocks is $N \leq 100$. The length of each M bit block is hard-coded $M = 131072$. If other values than these are desired, be sure that $M > 0.01n$ and $N = \lfloor \frac{n}{M} \rfloor$. All this in order for the P-values to be valid.

9 Overlapping Template Matching Test

Both this test and the non-overlapping template matching test from section 8 use an m bit window to search for a specific template in each M bit block. The difference between this test and the previous test from section 8 is that when the pattern is found, the window slides in this test only one bit before resuming the search.

The test start with partitioning the given bit sequence into N blocks of length M. The search start with creating an m bit window on the sequence, comparing that window against the template B and incrementing a counter when a match is found. The number of occurrences of B in each block is recorded by incrementing an array ν_i, where $i = 0, \ldots, 5$. ν_0 is incremented when there are no occurrences of B in a substring, ν_1 is incremented for one occurrence of B, etc. After the search has finished, consequently, we are left with an array $\nu_0, \ldots, \nu_5$ where each cell has the number of respective occurrences of the template B.

Next, the $\chi^2(obs)$ statistic is calculated

$$\chi^2(obs) = \sum_{i=0}^{5} \frac{(\nu_i - N\pi_i)}{N\pi_i} \tag{20}$$

where $\pi_0 = 0.367879$, $\pi_1 = 0.183940$, $\pi_2 = 0.137955$, $\pi_3 = 0.099634$, $\pi_4 = 0.069935$ and $\pi_5 = 0.140657$ as specified in the document of NIST [3]. Thereafter the P-value is computed

$$P\text{-value} = \text{igamc}\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) \tag{21}$$

The length of the bit sequence should be at least 10^6. The NIST recommends, as for the non-overlapping template matching test, templates of length 9 or 10. As for the other tests, a P-value ≥ 0.01 will allow is to accept the null hypothesis of randomness.

10 Maurer's Universal Statistal Test

This test has as purpose to detect whether or not the given bit sequence can be significantly compressed without loss of information. A significantly compressible sequence will be considered to be not random. More specific, the test looks for the number of bits between matching patterns.

In order to perform this test, the bit sequence ε is partitioned in two segments:

An initialization segment consisting of $Q.L$ bit non overlapping blocks.

A test segment consisting of $K.L$ bit non overlapping blocks.

The remaining bits at the end of the sequence that do not form a complete L block are disregarded. Figure 4 shows a schematic representation of the first step in this statistical test. The first Q blocks are used to initialized

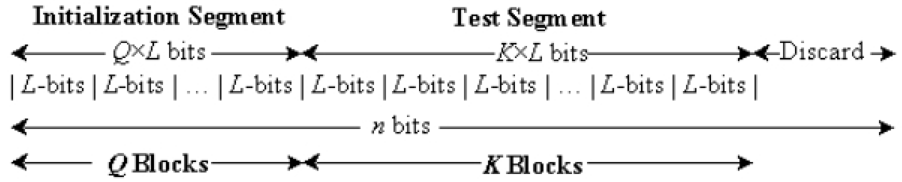
Initialization Segment
Test Segment
Q×L bits
K×L bits
Discard
| L-bits | L-bits | ... | L-bits | L-bits | L-bits | L-bits | ... | L-bits | L-bits |
n bits
Q Blocks
K Blocks

Figure 4: Division of a sequence into a initialization seqment and a test segment

the test, the remaining $K = \lfloor n/L \rfloor - Q$ blocks are test blocks.

The initialization segment is used to create a table T for each possible L bit value, i.e. the L bit value is used as an index into the table. The block number of the last occurrence of each L bit block is noted in the table.

Thereafter each of the K blocks in the test segment are examined and the number of blocks since the last occurrence of the same L bit block is determined. Then, the value in the table with the location of the same L bit block is replaced. The distance between re-occurrences of the same L bit blocks is calculated using an accumulating log_2 sum of all the differences detected in the K blocks.

The test statistic f_n is computed

$$\frac{1}{K} \sum_{i=Q+1}^{Q+K} log_2(i - T_j) \tag{22}$$

where T_j is the table entry corresponding to the decimal representation of the contents of the i^{th} L bit block. Finally the P-value is computed

$$P\text{-value} = \operatorname{erfc}\left(\left|\frac{f_n - expectedValue(L)}{\sqrt{2}\sigma}\right|\right) \tag{23}$$

$expectedValue(L)$ and σ are obtained from a table of precomputed values that can be found in section 2.9.4 of the NIST document [2]. The test requires a bit sequence of $n \geq (Q+K)L$. The L bit blocks should be chosen such that $6 \leq L \leq 16$. The initialization segment should have a length of $Q = 10.2^L$. The test segment requires $K = \lceil \frac{n}{L} \rceil - Q \approx 1000.2^L$. Table 10 shows that depending of the value of n, the values L, Q should be chosen.

n	L	$Q = 10.2^L$
$\geq$ 387.840	6	640
$\geq$ 904.960	7	1280
$\geq$ 2.068.480	8	2560
$\geq$ 4.654.080	9	5120
$\geq$ 1.342.400	10	10240
$\geq$ 22.753.280	11	20480
$\geq$ 49.643.520	12	40960
$\geq$ 107.560.960	13	81920
$\geq$ 231.669.760	14	163840
$\geq$ 496.435.200	15	327680
$\geq$ 1.059.061.760	16	655360

Table 3: The values of L and Q depending on the size of the bit sequence n in Maurer's Test

11 Lempel-Ziv Compression Test

This test continues in the same vein as Maurer's test from section 10. The Lempel-Ziv test determines how far a bit sequence can be compressed. It test the number of cumulative distinct 'words' in the sequence.

The test starts with creating a dictionary of disjoint and distinct words. This is accomplished by creating substrings from consecutive bits of the sequence until a substring is created that can not be found in the previous part of the sequence, i.e. a new substring which has not yet been entered in the dictionary. The resulting bit string is then entered into the dictionary and the algorithm starts again from the first bit after the last created word (no overlapping). Let us call the number of cumulatively distinct words W_{obs}

The only thing left is to compute the P-value

$$P\text{-value} = \frac{1}{2}\text{erfc}\left(\frac{\mu - W_{obs}}{\sqrt{2\sigma^2}}\right) \tag{24}$$

Considering an input sequence of $n = 10^6$, the values μ and σ are set to $\mu = 69586.25$, $\sigma = \sqrt{70.448718}$. There is no known theory is available to determine the exact values of μ and σ, these values were computed under assumption of randomness using SHA-1. SHA stands for Secure Hash Algorithm, a cryptographic hash function. It is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string. SHA-1 is the best established of the existing SHA hash functions, and is employed in several widely used security applications and protocols.

12 Linear Complexity Test

This tests determines whether the binary sequence is complex enough to be considered random. This is done by testing the length of a linear feedback shiftregister (LFSR)[4] Random sequences are characterized by long LFSRs.

In this test, the bit sequence is divided into N blocks of length M ($n = MN$). The linear complexity L_i of each of the N blocks is determined using the Berlekamp-Massey algorithm [4]. L_i is the shortest LFSR sequence that generates all bits in the block i.

The theoretical mean is calculated (under assumption of randomness)

$$\mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{(M/3 + 2/9)}{2^M} \tag{25}$$

For each substring, the value T_i is comuted as follows

$$T_i = (-1)^M (L_i - \mu) + 2/9 \tag{26}$$

and the T_i values are recorded in $\nu_0 \ldots \nu_6$ according to followin rules:

$T_i \leq -2.5$	Increment ν_0 by one
$-2.5 < T_i \leq -1.5$	Increment ν_1 by one
$-1.5 < T_i \leq -0.5$	Increment ν_2 by one
$-0.5 < T_i \leq 0.5$	Increment ν_3 by one
$0.5 < T_i \leq 1.5$	Increment ν_4 by one
$1.5 < T_i \leq 2.5$	Increment ν_5 by one
$T_i > 2.5$	Increment ν_6 by one

Thereafter, the χ^2 statistic is computed

$$\chi^2(obs) = \sum_{i=0}^{6} \frac{(\nu_i - N\pi_i)^2}{N\pi_i} \tag{27}$$

[4] LFSR is a shift register whose input bit is a linear function of its previous state.The only linear function of single bits is xor, thus it is a shift register whose input bit is driven by the exclusive-or (xor) of some bits of the overall shift register value.

where $\pi_0 = 0.01047$, $\pi_1 = 0.03125$, $\pi_2 = 0.0125$, $\pi_3 = 0.5$, $\pi_4 = 0.25$, $\pi_5 = 0.0625$, $\pi_6 = 0.02078$. Detailed calculations on how these π_i values are determined can be found in [2].

Finally, the P-value of this test is computed

$$P\text{-value} = igamc\left(\frac{6}{2}, \frac{\chi^2(obs)}{2}\right) \tag{28}$$

13 Serial Test

The serial test, determines the number of occurrences of all possible (2^m) overlapping m bit patterns across the entire sequence. It is expected from random sequences to have uniformity, i.e. every m bit pattern has the same chance of appearing as every other one. The purpose of this test is therefore testing whether the number of occurrences of all possible m bit patterns is as would be expected from a random sequence.

The initial sequence ε is extended by appending the first $m-1$ bits of the original sequence to the end of the sequence in order to create a new sequence ε'. Thereafter, the frequency of all possible overlapping $m-1$ bit blocks and all possible overlapping $m-2$ bit blocks. Let

$\nu_{i_l \ldots i_m}$ denote the frequency of the m bit pattern $i_l \ldots i_m$

$\nu_{i_l \ldots i_{m-1}}$ denote the frequency of the $m-1$ bit pattern $i_l \ldots i_{m-1}$

$\nu_{i_l \ldots i_{m-2}}$ denote the frequency of the $m-2$ bit pattern $i_l \ldots i_{m-2}$

After the algorithm has computed the above frequencies, the following functions are caculated respectively for each of the frequencies.

$$\begin{aligned}
\psi_m^2 &= \tfrac{2^m}{n} \textstyle\sum_{i_l \ldots i_m} \left(\nu_{i_l i_m} - \tfrac{n}{2^m}\right)^2 &&= \frac{2^m}{n} \sum_{i_l \ldots i_m} \nu_{i_l \ldots i_m}^2 - n \\
\psi_{m-1}^2 &= \tfrac{2^{m-1}}{n} \textstyle\sum_{i_l \ldots i_{m-1}} \left(\nu_{i_l \ldots i_{m-1}} - \tfrac{n}{2^{m-1}}\right)^2 &&= \frac{2^{m-1}}{n} \sum_{i_l \ldots i_{m-1}} \nu_{i_l \ldots i_{m-1}}^2 - n \\
\psi_{m-2}^2 &= \tfrac{2^{m-2}}{n} \textstyle\sum_{i_l \ldots i_{m-2}} \left(\nu_{i_l i_{m-2}} - \tfrac{n}{2^{m-2}}\right)^2 &&= \frac{2^{m-2}}{n} \sum_{i_l \ldots i_{m-2}} \nu_{i_l \ldots i_{m-2}}^2 - n
\end{aligned}$$

and

$$\begin{aligned}
\nabla \psi_m^2 &= \psi_m^2 - \psi_{m-1}^2 \\
\nabla^2 \psi_m^2 &= \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2
\end{aligned}$$

Now the P-values can be computed as follows

$$\begin{aligned} P - \text{value } 1 &= \text{igamc}\left(2^{m-2}, \nabla\psi_m^2\right) \\ P - \text{value } 2 &= \text{igamc}\left(2^{m-3}, \nabla^2\psi_m^2\right) \end{aligned}$$

If the $\nabla^2\psi_m^2$ or $\nabla\psi_m^2$ are too large, then all the possible the m bit blocks do not appear uniformly across the sequence ε'. The length of the patterns m and the length of the sequece n should be chosen such that $m < \lfloor log_2 n \rfloor - 2$.

14 Approximate Entropy Test

This test compares the frequency of overlapping blocks of two consecutive (m and $m+1$) against the expected restults for a random sequence.

As with the serial test in section 13, this test starts with extending the given input sequence by appending the first $m-1$ bits from the beginning to the end of the sequence in order to create n overlapping m bit sequences. Thereafter, the test continues with a frequency count of the n overlapping blocks. More concrete, if a block containing ε_j to ε_{j+m-1} is examined at time j, then the block containing ε_{j+1} to ε_{j+m} is examined at time $j+1$.

the degree of occurrence of each possible m bit template is calculated

$$C_i^m = \frac{\sharp i}{n} \qquad \text{for each value of i} \tag{29}$$

with $\sharp i$ an m bit pattern. Now a function $\phi^{(m)}$ is calculated

$$\phi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i log \pi_i \tag{30}$$

where $\pi_i = C_j^m$ and $j = log_2 i$.

The algorithm now repeats all the above steps replacing m by $m+1$ and the test statistic is computed

$$\chi^2 = 2n(log2 - (\phi^{(m)} - \phi^{(m+1)})) \tag{31}$$

Small values of $(\phi^{(m)} - \phi^{(m+1)})$ imply strong regularity, large values imply substantial fluctuations or irregularity. The resulting P-value is then

$$P\text{-value} = \text{igamc}(2^{m-1}, \frac{\chi^2}{2}) \tag{32}$$

The NIST recommends the user to choose m and n such that $m < \lfloor log_2 n \rfloor - 2$ in order for the test to give accurate results. The rules for accepting or rejecting the P-value is still the same: all P-values < 0.01 make us reject the null hypothesis of randomness.

15 Cummulative Sums (Cumsum) Test

This test determines whether the maximal excursion of the random walk defined by the cummulative sum of digits in parts of the tested sequence is too large or too small for the expected behavior for random sequences.

The bit sequence to be tested in converted into a sequence of values ±1, just as for the frequency test 2. Thereafter, the partial sum S_k of successive larger subsequences $k = 1, 2, ldotsn$ is calculated, starting with the first element of the sequence if the mode = 0 (forward cummulative sum) and with the last element if the mode = 1 (backward cummulative sum).

Mode = 0 (Forward) Mode = 1 (Backward)

$$\begin{array}{rclcrcl} S_1 & = & X_1 & \quad & S_1 & = & X_n \\ S_2 & = & X_1 + X_2 & & S_2 & = & X_n + X_{n-1} \\ & \vdots & & & & \vdots & \\ S_n & = & X_1 + X_2 + \ldots + X_n & & S_n & = & X_n + X_{n-1} + \ldots + X_1 \end{array}$$

For both cases (forward - and backward cummulative sum), the largest of the absolute value for all partial sums S_k is computed.

$$z = max_{1 \leq k \leq n} |S_k| \tag{33}$$

The P-value is computed as follows

$$\begin{aligned} P\text{-value} &= 1 - \sum_{k=(\frac{-n}{z}+1)/4}^{(\frac{n}{z}-1)/4} \left(\Phi\left(\frac{(4k+1)z}{\sqrt{z}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{z}}\right) \right) \\ &+ \sum_{k=(\frac{-n}{z}-3)/4}^{(\frac{n}{z}-1)/4} \left(\Phi\left(\frac{(4k+3)z}{\sqrt{z}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{z}}\right) \right) \end{aligned}$$

Where Φ is the standard normal distribution function $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2} du.$

The NIST recommends that each sequence to be tested consists at least of 100 bits. From P-values < 0.01 should be concluded that the sequence is not random.

16 Random Excursions Test

The random excursions tests focusses on the number of cycles having exactly K visits in a cummulative sum. A cycle of a random walk is a sequence of that begins and returns to the origin, where the origin a value 0 in a cumulative sum.

A set of cummulative sum subsequences $S = \{S_1, \ldots, S_n\}$ is achieved in the same way as computing the forward cummulative sum from the Cumsum test in **??**. Thereafter, a new sequence S' is formed by attaching zeros before and after the set S. Figure 5 shows an example of a graphical representation of the sequence S'. Now, we want to calculate the number of cycles in the sequence S'. Let J be that number. J is obtained as follows: Count all the zero crossings in the sequence S' where a zero crossing is a zero in the sequence occurring after the starting zero. Considering the example represented in figure 5, we obtain $J = 3$, i.e. The sequence has 3 cycles.

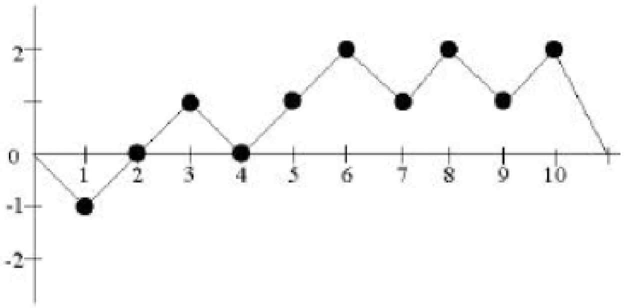

Figure 5: An example of the set $S' = \{\mathbf{0}, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, \mathbf{0}\}$. The zeros in bold are the ones that were added to S in order to create S'.

For each cycle and for each non-zero state value x with the values $-4 \leq -1$ and $1 \leq 4$, the frequency of each x within each cycle is computed. For example, in the cycle $(0, 1, 2, 1, 0)$ x has two occurences of $x = 1$ and one occurence of $x = 2$.

For each of the eight possible states of x, compute $\nu_k(x)$, which is the total number of cycles in which state x occurs exactly k times among all cycles, for $k = 0, 1, \ldots 5$. All frequencies ≥ 5 are also stored in $\nu_5(x)$. (Note that $\sum_{k=0}^{5} \nu_k(k) = J$).

Finally, for each of the eight states of x, the test statistic $\chi^2(obs)$ is computed

$$\chi^2 obs = \sum_{k=0}^{5} \frac{(\nu_k(x) - J\pi_k(x))^2}{J\pi_k(x)} \tag{34}$$

with $\pi_k(x)$ the probability that the state x occurs k times in a random distribution. Table values for π_k can be found in the document of the NIST

[3]. Now, for each state of x, a P-value is computed as follows

$$P\text{-value} = \text{igamc}\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) \tag{35}$$

In order to produce significant results, it is recommended that each sequence to be tested with the random excursions test consists of at least 10^6 bits. Since we obtain eight P-values from this test, we can only accept the sequence as random if all eight P-values are ≥ 0.01. If one or more of the P-values is < 0.01, further sequences should be examined to give conclusive results about the random generator.

17 Random Excursions Variant Test

This test focusses on the number of times that a particular state is visited in a cumulative sum random walk. It detects deviations from the expected number of visits to various states in the random walk. The largest part of this test follows the same computation as for the random excursions test from section 16, nevertheless we will briefly recapitulate these operations.

The input sequence ε is re-computed to form a normalized sequence X in which the zeros and ones are converted to the values -1 and $+1$ using the operation $X_i = 2\varepsilon - 1$ on each bit from the input sequence.

Thereafter, the partial sum S_i of successively larger subsequences, each starting with X_1, are computed to form the set $S = \{S_i\}$. Again a new sequence S' is created by appending zeros before and after the set S and for each of the non-zero states x (cfr. section 16) a value $\xi(x)$ is computed. $\xi(x)$ is the total number of times that state x occurs across all J cycles.

Finally, eight P-values are computed for each value of x as follows

$$P\text{-value} = \text{erfc}\left(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2)}}\right) \tag{36}$$

As with the random excursions test, it is recommended for the input sequence to have at least 10^6 bits. The rules for accepting the null hypothesis of randomness using the P-values from this test are the same as for the random excursions test, i.e. the sequence can only be accepted as random if all resulting P-values from this test are ≥ 0.01. Otherwise, further sequences have to be examined from the random generator in order to give conclusive results.

References

[1] *Quantum aspects of cryptography: From qutrit symmetric informationally complete projective operator valued measure key encryption to randomness quality control.* Frederik Vanden Berghe, Ph.D. thesis, Vrije Universiteit Brussels, 2011-2012.

[2] *Statistical test suite for random and pseudorandom number generators for cryptographic applications.* A. Rukhin and J. Soto and J. Nechvatal and M. Smid and E. Barker and S. Leigh and M. Levenson and M. Vangel and D. Banks and A. Heckert and J. Dray and S. Vo,
NIST Special Publication **800**, 22(2001).

[3] *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.* http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf

[4] *The Handbook of Applied Cryptography.* A. Menezes and P. Van Oorschot and S. Vanstone,
CRC Press, 1997