CHAPTER II

LITERATURE REVIEW

2.1 Cryptography

The definition of cryptography is the science of altering communication so that it cannot be understood without having the key.

In cryptography, there are two fundamental ways to use keys for encryption, which are symmetric algorithm and asymmetric algorithm. In symmetric algorithm, the same key is used to encrypt and decrypt the message. There are two types of symmetric algorithm, consists of block cipher and stream cipher. On the contrary, in asymmetric, the keys used to encrypt and decrypt the message are different. Some of the well known asymmetric algorithms are RSA, DSA, and ELGAMAL.

In block cipher, the data is encrypted in block. For example, AES uses a 128-bit block. Other examples of block ciphers are DES (64-bits), 3DES (64-bits), Blowfish (64-bits), Serpent (128-bits), Twofish (128-bits), Skipjack (64-bits), and IDEA (64-bits).

In contrast, the data in stream cipher is encrypted as a stream, one bit at a time. Examples of stream cipher algorithms include RC4, FISH, PIKE, and Grain-128. We discuss the stream cipher in details in the next section.

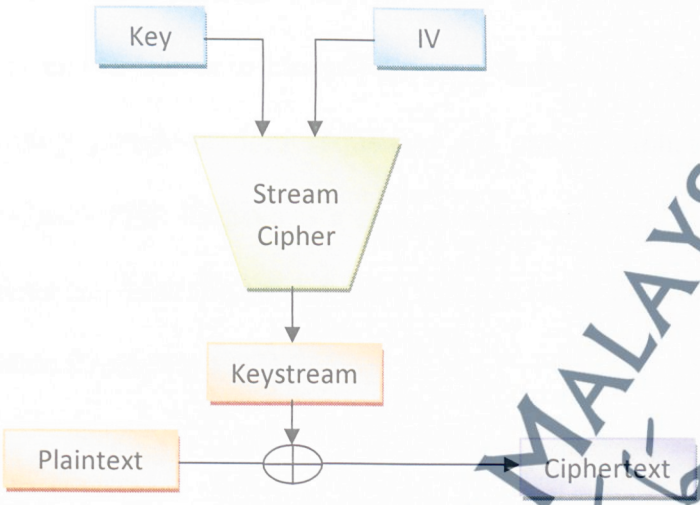
2.2 Stream Cipher Design

Stream cipher is another type of symmetric cipher system other than block cipher and it is one of the important classes of encryption algorithms. The Linear Feedback Shift Register (LFSR) is one of the components used for constructing a stream cipher algorithm. It is because LFSR can be implemented cheaply in hardware and its properties are well understood.

In The Free Encyclopedia (14 April 2013), it is mentioned that the LFSR is used in stream ciphers because it can be implemented easily in hardware and it can be readily analysed mathematically. However, the use of only LFSR in the stream ciphers is not sufficient to provide a good security. Therefore, to overcome this problem, the non-linear Boolean function is applied to remove the linearity and improve unpredictability.

The stream cipher will generate a sequence of bits that will be used as a key called keystream. The keystream then will be combined with plaintext, in an operation known as exclusive-or (XOR) operation, to produce the ciphertext. In stream cipher, the ciphertext is produced one at a time. Figure 2 below shows the overview of a stream cipher process.

Figure 2: Stream cipher process



Stream cipher algorithms can be further categorised into two types, which are synchronous stream ciphers and self-synchronous stream ciphers. In synchronous stream ciphers, the keystream is generated independently of the plaintext and ciphertext, whereas in the self-synchronous stream ciphers, the keystream depends on the previous ciphertext (Bucerzan et al., 2010).

2.2.1 Linear Feedback Shift Register (LFSR)

Linear Feedback Shift Register (LFSR) is the basic component for constructing many generators for stream cipher algorithm. LFSR is suitable for hardware implementation and it produces sequences with good statistical properties. The LFSR is a shift register in which some of its outputs are connected to the input through logic gates, typically using an exclusive-or (XOR). The input of LFSR uses a linear function of its previous state.

2.2.2 Non-Linear Feedback Shift Register (NLFSR)

Non-Linear Feedback Shift Registers (NLFSRs) have been proposed as an alternative to Linear Feedback Shift Registers (LFSRs) for generating pseudo-random sequences for stream ciphers. Non-Linear Feedback Shift Register is a generalisation of Linear Feedback Shift Register in which the current state is a non-linear function of the previous state (Dubrova, 2012).

2.2.3 Boolean Function

A Boolean function is a combination function of Linear Feedback Shift Register and Non-Linear Feedback Shift Register to obtain the output.

2.3 Grain-128 Stream Cipher Algorithm

Grain-128 stream cipher algorithm was introduced by Hell, Johansson, Maximov, and Meier in 2006 (Hell et al., 2006). This algorithm supports 128-bit key and 96-bit IV. Grain-128 is a family of stream ciphers that was submitted to the eSTREAM stream cipher competition. There are three main building blocks in Grain-128, which are a Linear Feedback Shift Register (LFSR), a Non-Linear Feedback Shift Register (NLFSR), and an Output Boolean Function. This algorithm is used with limited resources such as in gate count, power consumption and chip area (Hell et al., 2006). The following explains the description of Grain-128 stream cipher algorithm. Figure 3 and Figure 4 show the process of key initialization and keystream generation, respectively.

Figure 3: The key initialization process

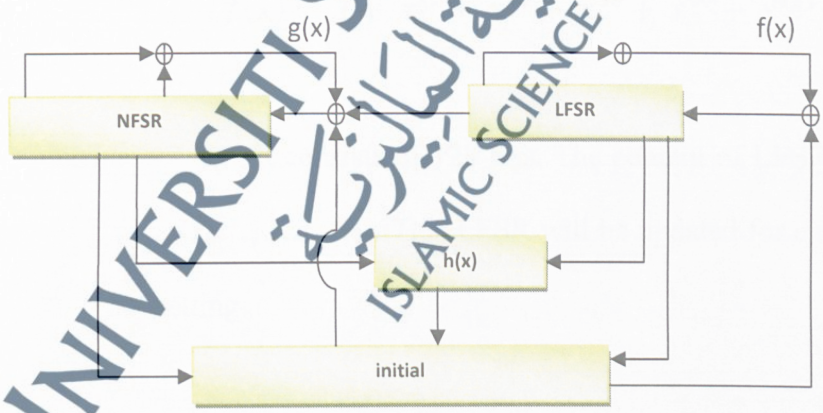
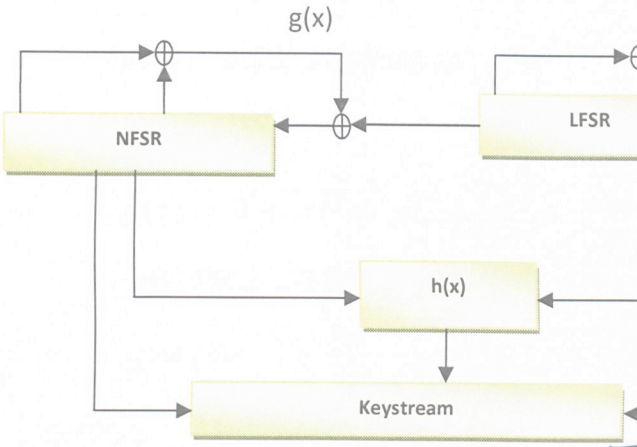


Figure 4: The keystream generation process



2.3.1 Description of Grain-128 Stream Cipher Algorithm

a. Linear Feedback Shift Register (LFSR)

The Linear Feedback Shift Register (LFSR), *f(x)*, is primitive polynomial of degree 128 and it is defined as

$$f(x) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128}$$

It consists of 128 bits. The content of LFSR is denoted as $s_t, s_{t+1}, \ldots, s_{t+127}$. This LFSR will be updated for each clock by setting

$$s_{i+128} = s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}$$

b. Non-Linear Feedback Shift Register (NLFSR)

The NLFSR, *g(x)*, is the sum of one linear and one bent function and it is defined as

$$g(x) = 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} + x^{63}x^{67} + x^{69}x^{101} + x^{80}x^{88} + x^{110}x^{111} + x^{115}x^{117}$$

It consists of 128 bits. The content of NLFSR is denoted as $b_t, b_{t+1} \ldots, b_{t+127}$. This NLFSR will be updated for each clock by setting

$$b_{i+128} = s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}$$

c. Output Boolean Function (nonlinear filtering function)

In Output Boolean Function, it consists of 9-input filter function taken from seven (7) bits of input from LFSR and two (2) bits of input from NLFSR. The degree of this function is three (3), denoted as deg($h(x)$) = 3. This function is defined as

$$h(x) = h(x_0, x_1, \ldots x_8) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8$$

d. Keystream, z

In order to generate the keystream, the cipher must firstly be initialized with the key and IV. To construct LFSR, the first 96 bits of LFSR are loaded with 96 bits IV, whereas the last 32 bits of the LFSR are filled with 1s. To construct NLFSR, the 128 bits of NLFSR are loaded with 128 bits key. The process of generating the cipher in the key initialization will be clocked until 256 times. After the cipher is clocked 256 times, the keystream has been generated.

$$h(x) = h(x_0, x_1, \dots x_8) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8$$

d. Keystream, z

In order to generate the keystream, the cipher must firstly be initialized with the key and IV. To construct LFSR, the first 96 bits of LFSR are loaded with 96 bits IV, whereas the last 32 bits of the LFSR are filled with 1s. To construct NLFSR, the 128 bits of NLFSR are loaded with 128 bits key. The process of generating the cipher in the key initialization will be clocked until 256 times. After the cipher is clocked 256 times, the keystream has been generated.

2.4 Randomness Testing

Randomness is used in various fields and it is interpreted in different forms, depending on the field related. For example, in cryptography, the randomness test is important to test the sequence whether it is random or not random. The sequence to be tested can be characterized and described in terms of probability, where the probability of the sequence is random or not random.

Therefore, there are several tools for testing the randomness of the sequence, which are in the form of statistical tests. According to Andrasiu et al. (2010), the National Institute of Standards and Technologies declared that there are more than 200 statistical tests for randomness. However, there are a number of statistical tests that are applied most by the researchers in order to verify the randomness of the sequence. Some of them are listed below:

1. The first statistical test suite was developed by Donald Knuth and was presented in his book entitled "The Art of Computer Programming Volume 2 Seminumerical Algorithms" (Andrasiu et al., 2010). There are 11 types of empirical tests which include The Frequency Test, The Serial Test, The Gap Test, The Poker Test, The Coupon Collector's Test, The Permutation Test, The Run Test, The Maximum-of-t Test, The Collision Test, The Birthday Spacings Test, and The Serial Correlation Test.

2.4 Randomness Testing

Randomness is used in various fields and it is interpreted in different forms, depending on the field related. For example, in cryptography, the randomness test is important to test the sequence whether it is random or not random. The sequence to be tested can be characterized and described in terms of probability, where the probability of the sequence is random or not random.

Therefore, there are several tools for testing the randomness of the sequence, which are in the form of statistical tests. According to Andrasiu et al. (2010), the National Institute of Standards and Technologies declared that there are more than 200 statistical tests for randomness. However, there are a number of statistical tests that are applied most by the researchers in order to verify the randomness of the sequence. Some of them are listed below:

1. The first statistical test suite was developed by Donald Knuth and was presented in his book entitled "The Art of Computer Programming Volume 2 Seminumerical Algorithms" (Andrasiu et al., 2010). There are 11 types of empirical tests which include The Frequency Test, The Serial Test, The Gap Test, The Poker Test, The Coupon Collector's Test, The Permutation Test, The Run Test, The Maximum-of-t Test, The Collision Test, The Birthday Spacings Test, and The Serial Correlation Test.

2. DIEHARD Suite of Statistical Test

The DIEHARD Suite of Statistical Test was introduced by George Marsaglia. This statistical test suite returns *p*-value that should be uniform on [0,1] if the input file contains independent random bits. The DIEHARD test suite is based on the probabilities or distribution of different outcomes (Rotz et al., 2001). It consists of 18 types of tests: The Birthday Spacings Test, The Overlapping 5-Permutations Test, The Binary Rank Test (for 31 × 31 matrices), The Binary Rank Test (for 32 × 32 matrices), The Binary Rank Test (for 6 × 8 matrices), The Bitstream Test, The Overlapping-Pairs-Sparse-Occupancy Test (OPSO), The Overlapping-Quadruples-Sparse-Occupancy Test (OQSO), The DNA Test, The Count-The-1's Test in a stream of bytes, The Count-The-1's Test in specific bytes, The Parking Lot Test, The Minimum Distance Test, The 3DSPHERES Test, The Squeeze Test, The Overlapping Sums Test, The Runs Test, and The Craps Test.

3. DIEHARDER

The DIEHARDER battery of test is a further development of the DIEHARD Suite of Statistical Test. It consists of 17 types of statistical tests from DIEHARD Suite of Statistical Test, Three tests from NIST Statistical Test Suite and new statistical tests. Total number of statistical tests in DIEHARDER is 25 types of tests, which consist of Birthday Spacing Test, OPERM 5 Test, 32 × 32 Matrix Rank Test,

6 × 8 Matrix Rank Test, Bitstream Test, OPSO Test, OQSO Test, DNA Test, Count 1 Stream Test, Count 1 Byte Test, Parking Lot Test, 2d Sphere Test, 3d Sphere Test, Squeeze Test, Runs Test, Craps Test, Marsaglia Tsang G CD Test, STS Monobit Test, STS Runs Test, STS Serial Test, RGB Bitdistance Test, RGB Minimum Bit Distance Test, RGB Permutations Test, RGB Langged Sum Test, and KS Test.

4. Scalable Parallel Pseudorandom Number Generator (SPRNG)

The tests in SPRNG are based on Knuth's Book (Andrasiu et al., 2010). SPRNG has an excellent property on generating large scale random number in parallel. Therefore, it is suited to parallel Monte Carlo applications. In SPRNG, there are nine types of tests, which include Collisions Test, Coupon Collectors Test, Equal-Distribution Test, Gap Test, Maximum-of-t Test, Permutations Test, Poker Test, Runs Up Test, and Serial Test (Li, 2012).

5. Crypt-XS Suite of Statistical Test

The Crypt-XS Suite of Statistical Test was developed by researchers at the Information Security Research Centre. As stated by Andrasiu et al. (2010) in their research titled "Statistical Evaluation of Cryptographic Algorithm", there are six types of statistical test in the Crypt-XS, which include The Frequency Test, The Binary Derivative

Test, The Change Point Test, The Runs Test, The Sequence Complexity Test, and The Linear Complexity Test.

6. NIST Statistical Test Suite

The most recent suite of statistical tests is the NIST Statistical Test Suite. It was developed through collaboration between the Computer Security Division and the Statistical Engineering Division at National Institute of Standard and Technology (NIST). Zhao et al. (2011) mentioned in their study that the NIST Statistical Test Suite is highly regarded among the list of batteries of tests. Moreover, the NIST Statistical Test Suite is one of the cryptographic tools involved in the evaluation of the candidates of Advanced Encryption Standard (Andrasiu et al., 2010). This statistical test consists of 16 types of tests: Frequency Test, Runs Test, Spectral Test, Lempel-Ziv Test, Random Excursion Variant Test, Maurer's Universal Test, Longest Runs of Ones Test, Random Excursion Test, Binary Matrix Rank Test, Block Frequency Test, Non-Overlapping Test, Overlapping Test, Linear Complexity Test, Serial Test, Approximate Entropy Test, and Cumulative Sums Test. The NIST Statistical Test Suite is discussed in details because this statistical test was used for this research project.

2.5 NIST Statistical Test Suite

NIST Statistical Test Suite is a statistical package that was developed to test the randomness of binary sequences produced by either hardware- or software-based cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. A number of tests in the test suite have the standard normal and the chi-square (χ^2) as reference distributions. If the sequence being tested is in fact non-random, the calculated test statistic will fall in extreme region of the reference distribution (Rukhin et al., 2010).

In this statistical test, the result of each test is summarized by one or more values known as *p*-value(s). The *p*-value(s) is the probability of the tested sequence to be random or not random. It has a uniform distribution on [0,1]. If the *p*-value(s) is greater than the significant level used (α), it can be concluded that the tested sequence is random (Zhao et al., 2011). In order to compute the *p*-values of each test in the NIST Statistical Test Suite, some special mathematical functions are required. Table 1 below shows the list of mathematical functions used.

Table 1: Mathematical functions used in NIST Statistical Test Suite

Mathematical Function	Test Type
Complementary Error Function $$erfc(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2}\, du$$	Frequency Test Runs Test Spectral Test Lempel-Ziv Test Random Excursion Variant Test Maurer's Universal Test
Incomplete Gamma Function $$Q(a,x) \equiv 1 - P(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)}$$ $$\equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t}\, t^{a-1}\, dt$$ Where $Q(a,0) = 1$ *and* $Q(a,\infty) = 0$ $$P(a,x) = \frac{1}{\Gamma(a)} \int_0^{x} e^{-t} t^{a-1}\, dt$$ Where $P(a,0) = 1$ *and* $P(a,\infty) = 0$ $$\Gamma(a) = \int_0^{\infty} t^{a-1} e^{-t}\, dt$$	Longest Runs of Ones Test Random Excursion Test Binary Matrix Rank Test Block Frequency Test Non-Overlapping Test Overlapping Test Linear Complexity Test Serial Test Approximate Entropy Test
Cumulative Distribution Function $$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2}\, du$$	Cumulative Sums Test

Listed below are the 16 types of tests in the NIST statistical test suite for performing the evaluation of randomness testing, along with the detailed explanation of each test. These tests have been discussed by Rukhin et al. (2010).

2.5.1 The Frequency Test

This test focuses on the proportion of zeros and ones for the entire sequence. The purpose of Frequency 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 description for Frequency Test is as follows:

i. The zeros and ones of the input sequence, (ε) are converted to values of -1 and $+1$ and are added together to produce $s_n = x_1 + x_2 + \ldots + x_n$ where $x_i = 2\varepsilon_i - 1$.

ii. Compute the test statistics $s_{obs} = \frac{|s_n|}{\sqrt{n}}$

iii. Compute $p - value = erfc(\frac{s_{obs}}{\sqrt{2}})$ where $erfc$ is complementary error function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.2 The Block Frequency Test

This test focuses on the ratio of ones within M-bit blocks, where M is the length of each block. The purpose of Block Frequency Test is to determine whether the number of ones in an M-bit block is approximately $M/2$, where M is the length of each block.

The description for Block Frequency Test is as follows:

i. The n-bit of input sequence is divided into non-overlapping blocks, each of M-bit, $N = \left\lfloor \frac{n}{M} \right\rfloor$. The extra bits are discarded.

ii. Determine the proportion π_i of ones in each M-bit block using the equation $\pi_i = \frac{1}{M} \sum_{j=1}^{M} \varepsilon_{(i-1)M+j}$ where $1 \leq i \leq N$.

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

iv. Compute $p - value = igamc(\frac{N}{2}, \frac{\chi^2(obs)}{2})$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.3 The Runs Test

This test focuses on the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. The purpose of Runs Test is to determine whether the number of runs of ones and zeros of various lengths is as expected for random sequence.

The description for Runs Test is as follows:

i. Compute the pre-test proportion π of ones in the input sequence, $\pi = \frac{\sum_j \varepsilon_j}{n}$.

ii. If $|n - 0.5| \geq \tau$, it is not necessary to perform Runs Test since the frequency test fails for the sequence. The Runs Test is performed if $|n - 0.5| < \tau$, where τ is defined as $\tau = \frac{2}{\sqrt{n}}$.

iii. Compute the test statistic $v_n(obs) = \sum_{k=1}^{n-1} r(k) + 1$ where $r(k) = 0$ if $\varepsilon_k = \varepsilon_{k+1}$ and $r(k) = 1$ otherwise.

iv. Compute $p - value = erfc\left(\frac{|v_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n\pi}(1-\pi)}\right)$, where $erfc$ is complementary error function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.4 The Longest Runs of Ones Test

This test focuses on the longest run of ones within M-bit blocks, where M is the length of each block. The purpose of Longest Runs of Ones Test is to determine whether the longest run of ones is consistent with the longest run of ones that would be expected in a random sequence.

The description for Longest Runs of Ones Test is as follows:

i. Divide the input sequence into M-bit blocks.

ii. Tabulate the frequencies v_i of the longest runs of ones in each block, where each cell contains the number of runs of ones of a given length as in Table 2 below.

Table 2: List of M values, minimum bit sequence and frequencies v_i

v_i	v_0	v_1	v_2	v_3	v_4	v_5	v_6
$M = 8$ **Minimum** $n = 128$	≤ 1	2	3	≥ 4			
$M = 128$ **Minimum** $n = 6272$	≤ 4	5	6	7	8	≥ 9	
$M = 10{,}000$ **Minimum** $n = 750{,}000$	≤ 10	11	12	13	14	15	≥ 16

iii. Values of $K, N, and\ \pi_i$ are determined by the values of M as follows:

Table 3: Values of M, K, N, and π_i

M	8	128	10,000
K	3	5	6
N	16	49	75
π_0	0.2148	0.1174	0.0882
π_1	0.3672	0.2430	0.2092
π_2	0.2305	0.2493	0.2483
π_3	0.1875	0.1752	0.1933
π_4	-	0.1027	0.1208
π_5	-	0.1124	0.0675
π_6	-	-	0.0727

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

v. Compute $p - value = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.5 The Binary Matrix Rank Test

This test focuses on the rank of disjoint sub-matrices of the whole sequence. The purpose of Binary Matrix Rank Test is to check for linear dependence among fixed length substrings of the original sequence.

The description for Binary Matrix Rank Test is as follows:

i. Divide the input sequence into $M.Q$-bit disjoint blocks, where M is the number of rows in each matrix and Q is the number of column in each matrix. There will exist $N = \left\lfloor \frac{n}{MQ} \right\rfloor$ blocks. The extra bits are discarded.

ii. Determine the binary rank, (R_ℓ) of each matrix, where $\ell = 1, \ldots N$. $a_{i,j}$ = Each element in M by N matrix. For forward row, start $i = 1$. For backward row, start $i = M$.

Step 1: If $a_{i,j} = 0$ (if elements on the diagonal $\neq 1$), swap elements in i^{th} row with elements in row that contains a '1' in the i^{th} column. Matrix cannot be altered if no other rows contain a '1' in the i^{th} column.

Step 2: If $a_{i,j} = 1$, if any other row contains a '1' in the i^{th} column, replace each element in that row with the exclusive-or (XOR) operation of that element and the elements in the i^{th} row.

iii. 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.

iv. Compute $\chi^2(obs) = \frac{(F_M - 0.2888.N)^2}{0.2888.N} + \frac{(F_{M-1} - 0.5776.N)^2}{0.5776.N} + \frac{(N - F_M - F_{M-1} - 0.1336.N)^2}{0.1336.N}$

v. Compute $p - value = e^{x^{2(obs)/2}}$

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.6 The Discrete Fourier Transform (Spectral) Test

This test focuses on the peak heights in the Discrete Fourier Transform of the sequence. The purpose of Discrete Fourier Transform (Spectral) Test is to detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness.

The description for Discrete Fourier Transform (Spectral) Test is as follows:

i. The zeros and ones of the input sequence (ε) are converted to values of −1 and +1 and are added together to produce $s_n = x_1 + x_2 + \ldots + x_n$ where $x_i = 2\varepsilon_i - 1$.

ii. A Discrete Fourier Transform (DFT) is applied on s_n to produce $S = DFT(s_n)$ using $f_i = \sum_{k=1}^{n} s_k \exp(\frac{2\pi_i(k-1)j}{n})$ where $j = 0, \ldots n-1, i \equiv \sqrt{-1}$ (Rukhin et al., 2010).

iii. Calculate $= modulus(S') \equiv |S'|$, where S' is the substring consisting of the first $n/2$ element in S, and the modulus function produces a sequence of peak heights.

iv. Compute $T = \sqrt{\left(log\frac{1}{0.05}\right)n} =$ 95% peak height threshold value. Under an assumption of randomness, 95% of the values obtained from the test should not exceed T.

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

vi. Compute N_1 = the actual observed number of peaks in M that are less than T.

vii. Compute $d \equiv \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/2}}$

iv. Compute $P - value = erfc(\frac{|d|}{\sqrt{2}})$ where $erfc$ is complementary error function.

The decision rule of this test is, if p-value ≥ α, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.7 The Non-Overlapping Template Test

This test focuses on the number of occurrences of pre-specified target strings. The purpose of Non-Overlapping Template Test is to detect the sequences that exhibit too many occurrences of a given non-periodic pattern.

The description for Non-Overlapping Template Test is as follows:

i. The n-bit of input sequence is divided into N-bit blocks, each of M-bit, $N = \left\lfloor \frac{n}{M} \right\rfloor$. The extra bits are discarded.

ii. Number of times the specific pattern, W_j is found in the j^{th} block, where $1 \leq j \leq N$.

iii. Compute $\mu = \frac{(M-m+1)}{2^m}$ and $\sigma^2 = M(\frac{1}{2^m} - \frac{2m-1}{2^{2m}})$ where m is length of template.

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

v. Compute $p - value = igamc(\frac{N}{2}, \frac{\chi^2(obs)}{2})$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value ≥ α, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.8 The Overlapping Template Test

This test focuses on the number of occurrences of pre-specified target strings. The purpose of Overlapping Template Test is to detect the sequences that show too many occurrences of a given non-periodic pattern.

The description for Overlapping Template Test is as follows:

i. The n-bit of input sequence is divided into N-bit blocks, each of M-bit, $N = \left\lfloor \frac{n}{M} \right\rfloor$. The extra bits are discarded.

ii. Calculate the number of occurrences, v_i of B in each N block, where B is the m-bit (length of template) to be matched.

iii. Compute $\lambda = \frac{(M-m+1)}{2^m}$ and $\eta = \frac{\lambda}{2}$.

iv. Compute $\chi^2(obs) = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i}$ where $\pi_0 = 0.364091$, $\pi_1 = 0.185659$, $\pi_2 = 0.139381$, $\pi_3 = 0.100571$, $\pi_4 = 0.070432$, $\pi_5 = 0.139865$.

v. Compute $p - value = igamc\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right)$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.9 The Maurer's Universal Statistical Test

This test focuses on the number of bits between matching patterns, which is a measurement that is related to the length of a compressed sequence. The purpose of Maurer's Universal Statistical Test is to detect whether the sequence is enough to be significantly compressed without loss of information.

The description for Maurer's Universal Statistical Test is as follows:

i. n-bit of input sequence is divided into two blocks, which are the initialization segment and the test segment. The initialization segment consists of QL-bit non-overlapping blocks, where Q is the number of blocks in the initialization sequence and L is the length of each block, while the test segment consists of KL-bit non-overlapping blocks, where $K = \lfloor {}^{n}/_{L} \rfloor - Q$. The extra bits are discarded.

ii. The first Q blocks are used to initialize the test, whereas the remaining K blocks are the test blocks.

iii. A table is created using the initialization segment for each possible L-bit value. The block number of the last occurrence of each L-bit block is noted in the table.

iv. Examine each of the K blocks in the test segment and determine the number of blocks as the last occurrence of the same L-bit block $(i - T_j)$, where T_j is the table entry

corresponding to the decimal representation of the contents of the i^{th} L-bit block.

v. Replace the value in the table with the location of the current block. $(i = T_j)$.

vi. Add the calculated distance between re-occurrences of the same L-bit block to an accumulating log_2 sum of all the differences detected in the K blocks. $sum = sum + log_2(i - T_{j)}$.

vii. Compute $f_n = \frac{1}{K}\sum_{i=Q+1}^{Q+K} log_2(i - T_j)$.

viii. Compute $p - value = erfc(|f_n - \frac{expectedValue(L)}{\sqrt{2}\sigma}|)$

where $erfc$ is complementary error function. The $expectedValue(L)$ and σ are taken from a table of pre-computed values, as presented in Table 4 below:

Table 4: Values of *L*, *expectedValue* and α

L	*expectedValue*	σ
6	5.2177052	2.954
7	6.1962507	3.125
8	7.1836656	3.238
9	8.1764248	3.311
10	9.1723243	3.356
11	10.170032	3.384
12	11.168765	3.401
13	12.168070	3.410
14	13.167693	3.416
15	14.167488	3.419
16	15.167379	3.421

The decision rules of this test is, if p-value ≥ α, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.10 The Lempel-Ziv Compression Test

This test focuses on the number of cumulatively distinct patterns (words) in the sequence. The purpose of Lempel-Ziv Test is to determine how far the tested sequence can be compressed.

The description for Lempel-Ziv Compression Test is as follows:

i. Parse the sequence into consecutive, disjoint, and distinct words that will form a "dictionary" of words, W_{obs} in the sequence.

ii. Compute $P - value = \frac{1}{2}\, erfc\left(\frac{\mu - W_{obs}}{\sqrt{2\sigma^2}}\right)$ where $erfc$ is complementary error function, $\mu = 69586.25$ and $\sigma = \sqrt{70.448718}$ when $n = 1{,}000{,}000$. For other values of n, the values of μ and σ need to be calculated. There are no known theory available to determine the exact values of μ and σ. These values are computed (under an assumption of randomness) using SHA-1.

The decision rule of this test is, if p-value ≥ α, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.11 The Linear Complexity Test

This test focuses on the length of Linear Feedback Shift Register (LFSR). The purpose of Linear Complexity Test is to determine whether the sequence is enough to be random.

The description for Linear Complexity Test is as follows:

i. The n-bit of input sequence is divided into N blocks, each of M-bit, where $n = MN$.

ii. The Berlekamp-Massey Algorithm is applied in order to obtain an LFSR (Rukhin et al., 2010).

iii. Compute $\mu = \frac{M}{2} + \frac{9+(-1)^{M+1}}{36} - \frac{\frac{M}{3}+\frac{2}{9}}{2^M}$ and $T_i = (-1)^M(L_i - \mu) + \frac{2}{9}$.

iv. N blocks are divided into seven (7) fixed groups (v_i), as shown below:

$$v_0(T_i \leq -2.5)$$
$$v_1(-2.5 < T_i \leq -1.5)$$
$$v_2(-1.5 < T_i \leq -0.5)$$
$$v_3(-0.5 < T_i \leq 0.5)$$
$$v_4(0.5 < T_i \leq 1.5)$$
$$v_5(1.5 < T_i \leq 2.5)$$
$$v_6(T_i > 2.5)$$

v. Compute $\chi^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$, where K is the number of degree of freedom and it has been hard-coded into the test, $K = 6$. The values of theoretical probabilities, π_i are: $\pi_0 = 0.01047, \pi_1 = 0.03125, \pi_2 = 0.125, \pi_3 = 0.5, \pi_4 = 0.25\ \pi_5 = 0.0625, \pi_6 = 0.02078.$

vi. Compute $p - value = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.12 The Serial Test

This test focuses on the frequency of all possible overlapping m-bit patterns across the whole sequence where m refers to the length of each block. The purpose of Serial Test is to determine whether the number of occurrences of m-bit overlapping patterns is approximately the same as would be expected for a random sequence. m-bit refers to the length in bits of each block.

The description for Serial Test is as follows:

i. Form an augmented sequence ε'. Extend the input sequence by appending the first $m - 1$ bits to the end of the sequence for distinct values of n.

ii. Determine the frequency of all possible overlapping m-bit blocks, all the possible overlapping $(m - 1)$-bit blocks, and all possible overlapping $(m - 2)$-bit blocks.

iii. Compute:

$$\psi_m^2 = \frac{2^m}{n} \sum_{i_1 \ldots i_m} \left(v_{i_1 \ldots i_m} - \frac{n}{2^m}\right)^2 = \frac{2^m}{n} \sum_{i_1 \ldots i_m} v_{i_1 \ldots i_m}^2 - n$$

$$\psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_1 \ldots i_{m-1}} (v_{i_1 \ldots i_{m-1}} - \frac{n}{2^{m-1}})^2 =$$

$$\frac{2^{m-1}}{n} \sum_{i_1 \ldots i_{m-1}} v_{i_1 \ldots i_{m-1}}^2 - n$$

$$\psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_1 \ldots i_{m-2}} (v_{i_1 \ldots i_{m-2}} - \frac{n}{2^{m-2}})^2 =$$

$$\frac{2^{m-2}}{n} \sum_{i_1 \ldots i_{m-2}} v_{i_1 \ldots i_{m-2}}^2 - n$$

iv. Compute $\nabla \psi_m^2 = \psi_m^2 - \psi_{m-1}^2$ and $\nabla^2 \psi_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2$

v. Compute $p - value\ 1 = igamc\left(2^{m-2}, \nabla \psi_m^2\right)$ and $p - value\ 2 = igamc\left(2^{m-3}, \nabla^2 \psi_m^2\right)$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value ≥ α, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.13 The Approximate Entropy Test

This test focuses on the frequency of all possible overlapping m-bit patterns across the whole sequence. m-bit refers to the length of each block. The purpose of Approximate Entropy Test is to compare the frequency of overlapping blocks of two conservative/adjacent lengths (m and m + 1) against the expected result for a normally distributed sequence. m refers to the length of each blocks.

The description for Approximate Entropy Test is as follows:

i. Augment the n-bit of input sequence to create n overlapping m-bit sequence by appending $m-1$ bits from the beginning of the sequence to the end of the sequence.

ii. Compute the overlapping m-bit blocks where the calculated count, 2^m = possible m-bit strings.

iii. Compute $C_i^m = \frac{\#i}{n}$ for each value of i.

iv. Compute $\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i$, where $\pi_i = C_j^3$ and $j = log_2 i$

v. **Repeat steps i–iv, replacing m by $m+1$**

vi. Compute the test statistics: $\chi^2 = 2n[\log 2 - ApEn(m)]$, where $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$

vii. Compute $p-value = igamc(2^{m-1}, \frac{\chi^2}{2})$ where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.14 The Cumulative Sums Test

This test focuses on 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 Cumulative Sums Test is to determine

whether the sum of the partial sequences occurring in the tested sequence is too large or too small.

The description for Cumulative Sums Test is as follows:

i. The zeros and ones of the input sequence (ε) are converted to values of -1 and $+1$ and are added together to produce $s_n = x_1 + x_2 + \ldots + x_n$ where $x_i = 2\varepsilon_i - 1$.

ii. Compute partial sums S_i of successively larger subsequences, each starting with X_1 ($if\ mode = 0$), as shown in Table 5 below.

Table 5: Compute partial sums

Mode = 0 (forward)	Mode = 1 (backward)
$S_1 = X_1$	$S_1 = X_n$
$S_2 = X_1 + X_2$	$S_2 = X_n + X_{n-1}$
$S_3 = X_1 + X_2 + X_3$	$S_3 = X_n + X_{n-1} + X_{n-2}$
....	....
$S_n = X_1 + X_2 + X_3 + \ldots X_n$	$S_n = X_n + X_{n-1} + X_{n-2} + \ldots X_1$

iii. Compute the test statistic $z = max_{1 \ll k \ll n} |S_k|$ where $max_{1 \ll k \ll n} |S_k|$ is the largest of the absolute values of the partial sums S_k.

iv. Compute $p\text{-}value = 1 - \sum_{k=(\frac{-n}{z}+1)/4}^{(\frac{n}{z}-1)/4} [\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi(\frac{(4k-1)z}{\sqrt{n}})] + \sum_{k=(\frac{-n}{z}-3)/4}^{(\frac{n}{z}-1)/4} [\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi(\frac{(4k-1)z}{\sqrt{n}})]$ where Φ is the Standard Normal Cumulative Probability Distribution Function.

v. Repeat step ii–iv for X_n ($if\ mode = 1$).

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.15 The Random Excursion Test

This test focuses on the number of cycle having exactly k visits in a cumulative sum random walk. The purpose of Random Excursion Test is to determine if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence. The description for Random Excursion Test is as follows:

i. The zeros and ones of the input sequence (ε) are converted to values of -1 and $+1$ and are added together to produce $s_n = x_1 + x_2 + \ldots + x_n$ where $x_i = 2\varepsilon_i - 1$.

ii. Compute the partial sums, S_i of successively larger subsequences, each starting with x_1. Form the set $S = \{S_i\}$

$S_1 = X_1$

$S_2 = X_1 + X_2$

$S_3 = X_1 + X_2 + X_3$

...

$S_n = X_1 + X_2 + X_3 + \ldots X_n$

iii. Form a new sequence S' by attaching zeros before and after the set S. The new sequence, $S' = 0, s_1, s_2, \ldots s_n, 0$

iv. Let J = the total number of zeros crossing in S'.

v. For each cycle and for each non-zero state, $-4 \leq x \leq -1$ *and* $1 \leq x \leq 4$, compute the frequency of each x within each cycle.

vi. For each of the eight states of x, compute $v_k(x)$ = the total number of cycle in which state x occurs k times among all cycles, for $k = 0, 1, 2, \ldots, 5$.

vii. Compute $\chi^2(obs) = \sum_{k=0}^{5} \frac{(v_i(x) - J\pi_k(x))^2}{J\pi_{k(x)}}$ for each of the eight states of x, where $\pi_{k(x)}$ is the probability that the state x occurs k times in a random distribution, as in Table 6 below (Rukhin et al., 2010).

Table 6: The Theoretical Probabilities

	π_0	π_1	π_2	π_3	π_4	π_5
$x = 1$	0.5000	0.2500	0.1250	0.0625	0.0312	0.0312
$x = 2$	0.7500	0.0625	0.0469	0.0352	0.0264	0.0791
$x = 3$	0.8333	0.0278	0.0231	0.0193	0.0161	0.0804
$x = 4$	0.8750	0.0156	0.0137	0.0120	0.0105	0.0733
$x = 5$	0.9000	0.0100	0.0090	0.0081	0.0073	0.0656
$x = 6$	0.9167	0.0069	0.0064	0.0058	0.0053	0.0588
$x = 7$	0.9286	0.0051	0.0047	0.0044	0.0041	0.0531

viii. Compute $p - value = igamc\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right)$ for each state of x, where $igamc$ is incomplete gamma function.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

2.5.16 The Random Excursion Variant Test

This test focuses on the total number of times a particular state is visited in a cumulative sum random walk. The purpose of Random Excursion Variant Test is to detect deviations from the distributions of the number of visits of a random walk to a certain state.

The description for Random Excursion Variant Test is as follows:

i. The zeros and ones of the input sequence (ε) are converted to values of -1 and $+1$ and are added together to produce $s_n = x_1 + x_2 + \ldots + x_n$ where $x_i = 2\varepsilon_i - 1$.

ii. Compute the partial sums S_i of successively larger subsequences, each starting with x_1. Form the set $S = \{S_i\}$

iii. Form a new sequence S' by attaching zeros before and after the set S. The new sequence, $S' = 0, s_1, s_2, \ldots s_n, 0$

iv. For each of the eighteen non-zero states of x, compute $\xi(x)$ = the total number of times that state x occurred across all J cycles.

v. For each $\xi(x)$, compute $P - value = erfc(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2}})$, where $erfc$ is complementary error function. All the eighteen $P - values$ are computed.

The decision rule of this test is, if p-value $\geq \alpha$, the tested sequence is random. Otherwise, the tested sequence is not random. α is significance level used.

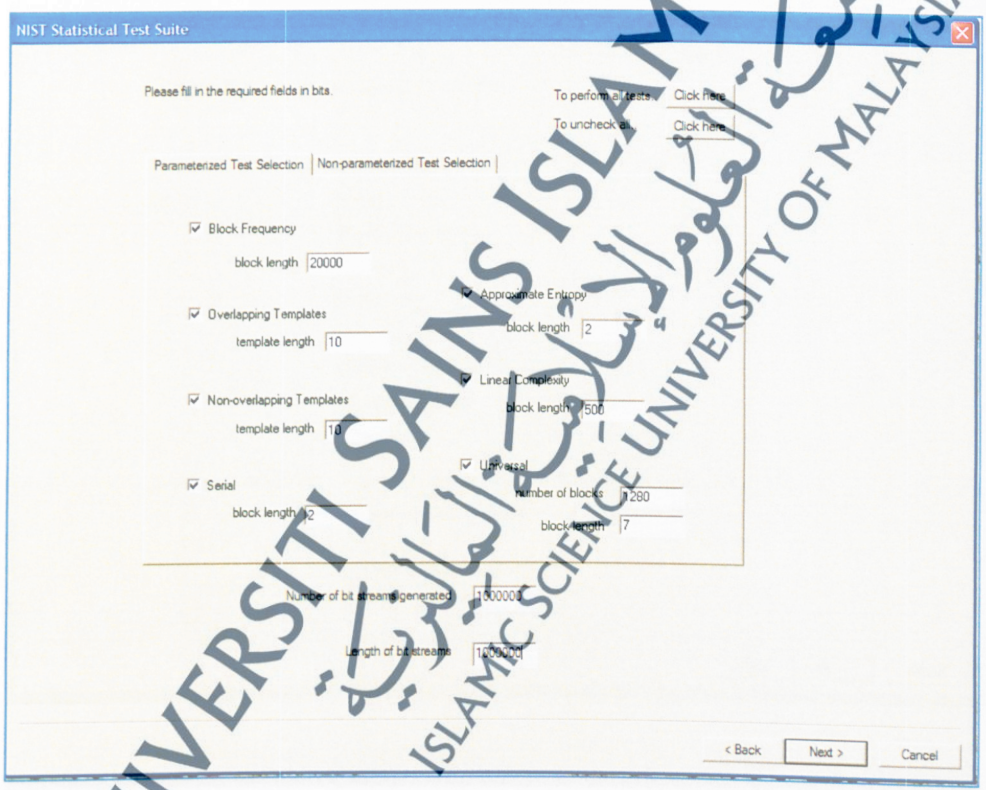
For all the 16 tests in NIST Statistical Test Suite, it can be divided into 2 categories, which are Parameterized Test Selection and Non-Parameterized Test Selection. The Block Frequency Test, Overlapping Template Test, Non-Overlapping Templates Test, Serial Test, Approximate Entropy Test, Linear Complexity Test, and Maurer's Universal Test are classified as Parameterized Test Selection where the user is required to define one or more parameter value(s) such as the block size of input sample, the number of block per input sample, and the length in bit of each template. The other tests are classified as Non-Parameterized Test Selection where the user is not required to enter any parameter in obtaining the *p*-value for each test. Figure 4–11 below show the step by step of Graphical User Interface (GUI) for NIST Statistical Test Suite that was used in this research.

Step 1: Click on the 'Input File' and choose the file to be used. For 'Select input mode', select 'Bits in ASCII format'. For 'Test Mode', select 'Regenerate'. Then, click 'Next' for further action. Figure 5 shows the representation of the above actions.

Figure 5: Representation 1 of NIST Statistical Test Suite

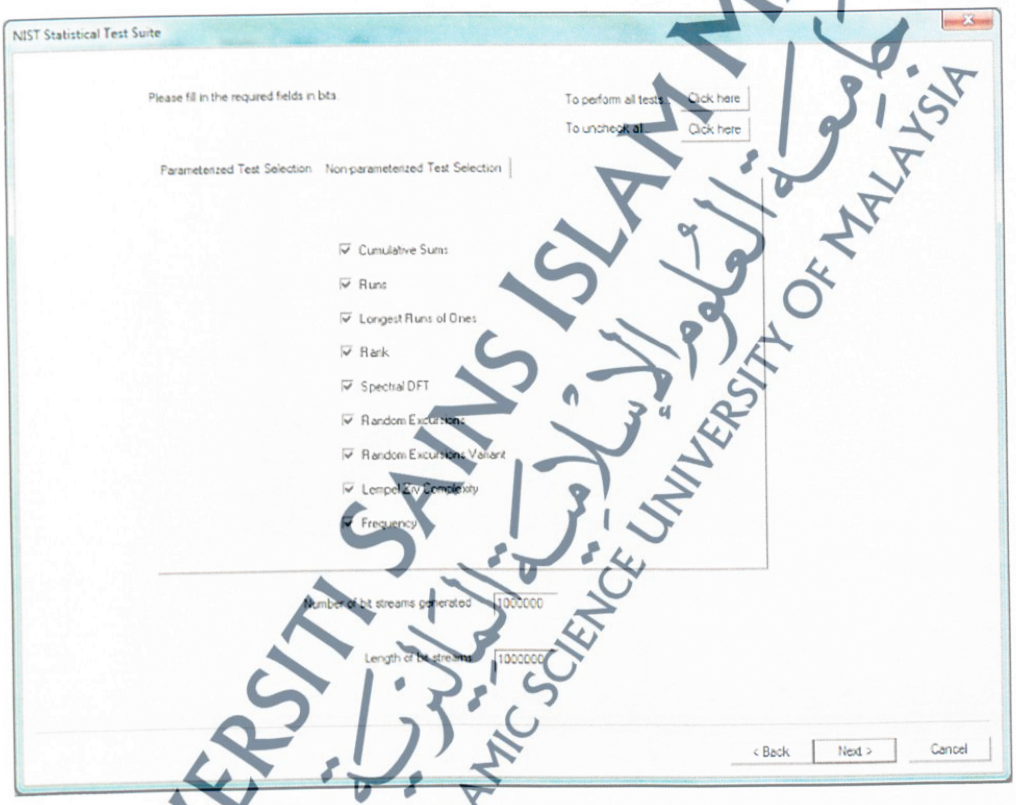
Step 2: NIST Statistical Test Suite has 16 types of tests. To carry out all the 16 tests, select 'To perform all tests'. All the 16 tests will be ticked automatically. For Parameterized Test Selection, one or two parameter value(s) need to be entered. Figure 6 shows the representation of the above actions. Then, click 'Next' for further action. The requirement of parameter value(s) for each test is discussed in details in Chapter 3.

Figure 6: Representation 2 of NIST Statistical Test Suite



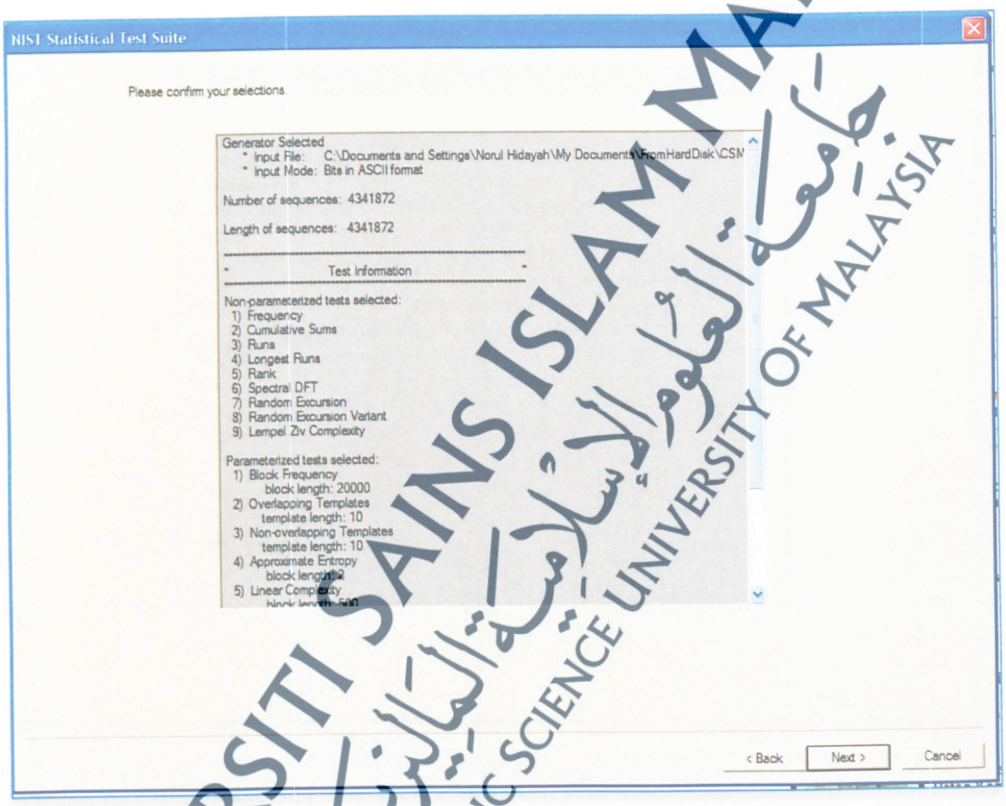
Step 3: For Non-Parameterized Test Selection, parameter value is not required. Then, the values for the 'Number of bit streams generated' and the 'Length of bit streams' need to be entered for the length of binary sequence to be tested. Then, click 'Next' for further action. Figure 7 shows the representation of the above actions.

Figure 7: Representation 3 of NIST Statistical Test Suite



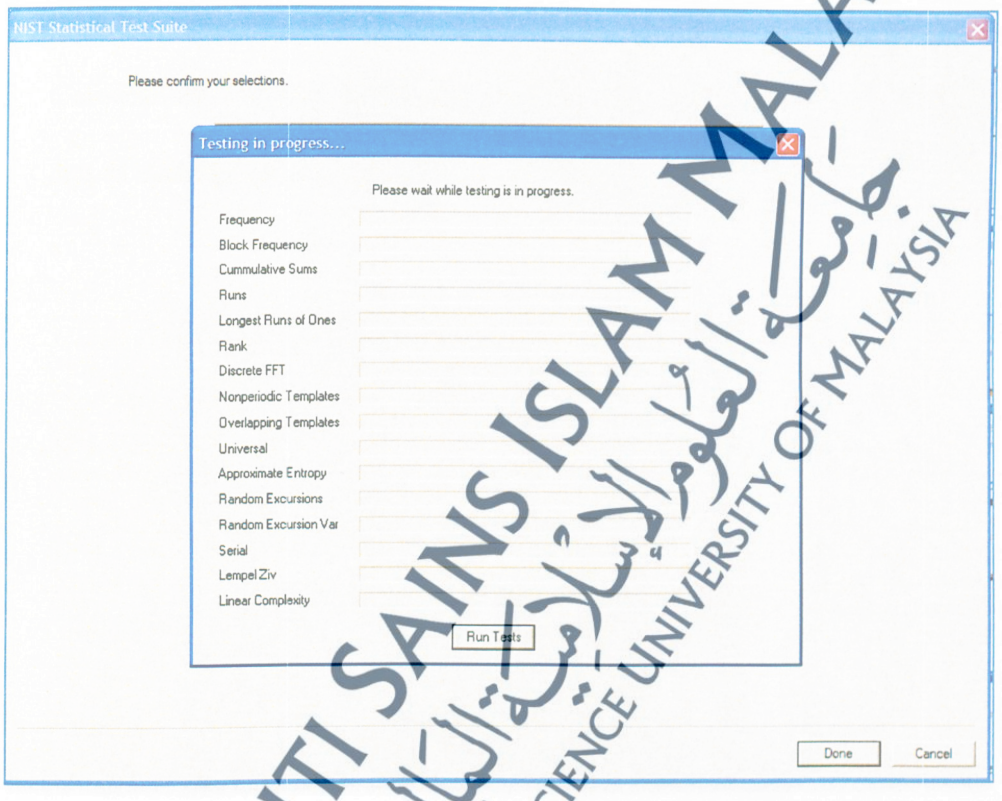
Step 4: The dialog box will show all information such as input file, input mode, number of sequences, length of sequences, and test information of NIST Statistical Tests. Then, click 'Next' for further action. Figure 8 shows the representation of the above actions.

Figure 8: Representation 4 of NIST Statistical Test Suite



Step 5: Finally, click 'Run Tests' and the progress bar for each test will appear to show the progress until the tests have been completed. Figure 9 shows the representation of the above action.

Figure 9: Representation 5 of NIST Statistical Test Suite



Before the statistical test is carried out, some assumptions must be made on the binary sequence.

1. Uniformity: it means that for any point in the generation of a sequence of random or pseudorandom bits, the occurrence of zero and one is the same. For example, the probability of each zero and one is exactly ½. The

expected occurrences number of zeros and ones are equal to $n/2$, where n is the length of sequence to be tested.

2. Scalability: it means that any test applied to a sequence can also be applied to subsequences at random. If the sequence is random, then any subsequence should also be random. Therefore, any subsequence should pass any tests for randomness.
3. Consistency: it means that the behaviour of a generator must be consistent across starting values (seeds).