CHAPTER III

RESEARCH METHODOLOGY

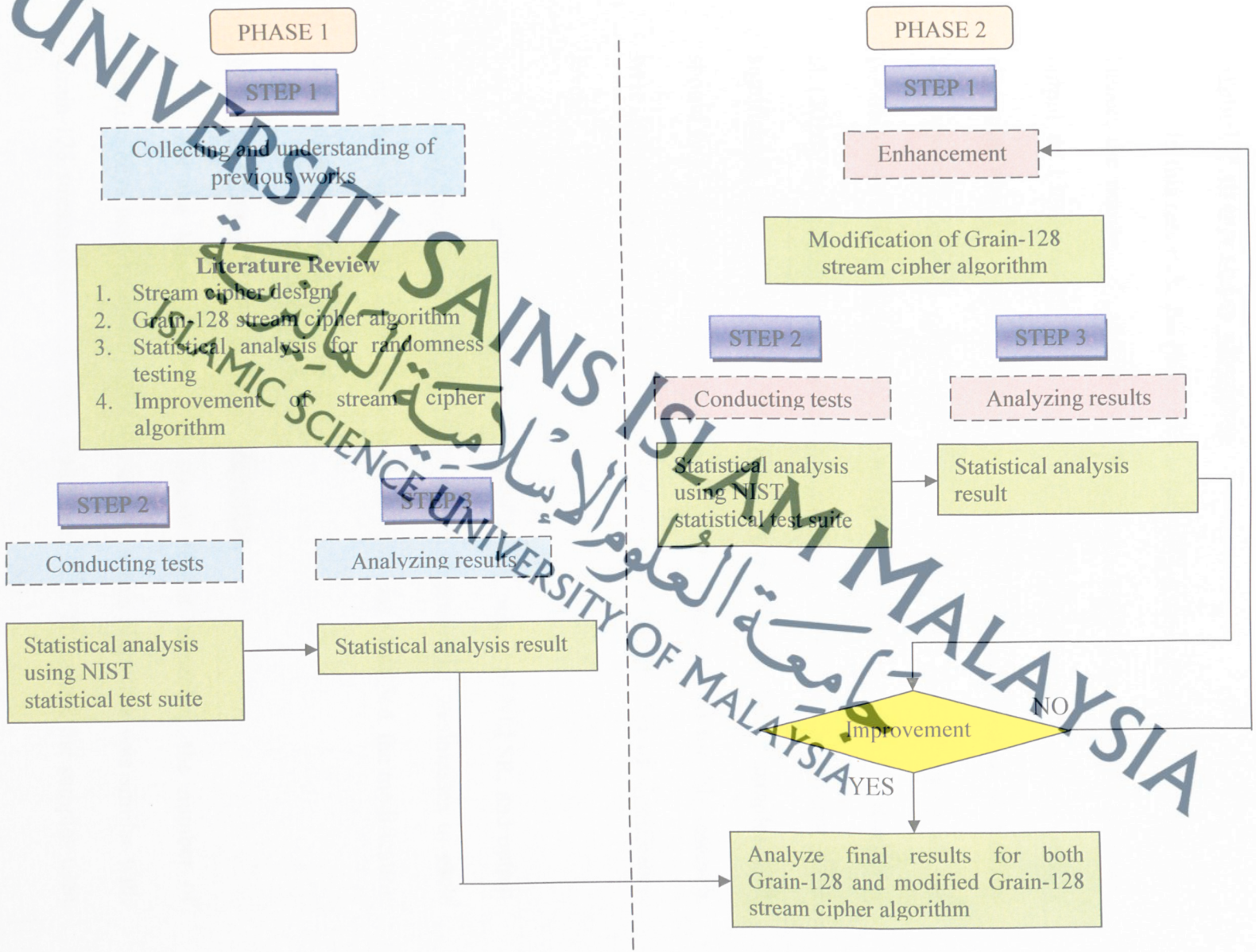
3.1 Research Design

This chapter explains the research design used for conducting this research project. The research design of this project can be divided into 2 phases, which are Phase 1 and Phase 2. Each phase consists of 3 steps. Figure 10 shows the workflow of research design for this project.

For Phase 1, the first step was collecting and understanding the previous works related to this research, such as the literature review on stream cipher design, Grain-128 stream cipher algorithm, and randomness testing using statistical analysis. The second step of this phase was to conduct randomness testing of statistical analysis using NIST Statistical Test Suite. There were 16 types of statistical tests conducted in NIST Statistical Test Suite, as mentioned earlier in Chapter 2. Finally, the third step of this phase was analysing the statistical analysis results of Grain-128 obtained from the statistical analysis using the NIST Statistical Test Suite.

For Phase 2, the first step of this phase was carrying out enhancement on Grain-128 stream cipher algorithm, which was later known as Modified Grain-128 (MG-128) stream cipher algorithm. The enhancement was conducted based on the results and analysis obtained from Grain-128 stream cipher algorithm. The results obtained were further analysed in order to improve the previous algorithm. The second step was conducting the randomness testing against the Modified Grain-128 (MG-128) using the same NIST Statistical Test Suite. Then, the third step was analysing the statistical analysis result of Modified Grain-128 (MG-128) obtained from the experiment conducted. The results were observed and analysed to determine whether the Modified Grain-128 (MG-128) had any improvement compared to the previous algorithm. If the result did not show the desired improvement, Phase 2 would have to be reconstructed. If the result did show the desired improvement, then the aims of conducting this research would be considered successful and the final results for both Grain-128 and Modified Grain-128 stream cipher algorithms must be analysed.

Figure 10: Workflow of research design



3.2 Population and Sample

Grain-128 stream cipher algorithm

In this research, the Grain-128 stream cipher algorithm used a 128-bit key. Hence, the number of possible keys that might be used to produce keystream, output of LFSR, output of NLFSR, and output of Boolean Function of this algorithm was 2^{128}, which is equal to 3.4×10^{38}. Since the number of possible keys was too large, it was impossible to conduct this research using all the possible keys due to infeasible system and time constraint. As stated by Rukhin et al. (2010), the number of samples used should be on the order of the inverse of the significance level, that is, for a level of 0.01 (1%), the number of samples used should have at least 100 samples. Therefore, the samples used for this research were 100 different keys that were chosen randomly for 1%–5% of significance level.

The reason to evaluate the output of LFSR, output of NLFSR, and output of Boolean Function of this algorithm was to discover the weaknesses of each component in this algorithm. Once the weak point was identified, the modification of the algorithm could be constructed.

Modified Grain-128 stream cipher algorithm

For the Modified Grain-128 stream cipher algorithm, the number of possible keys used to produce the keystream of this algorithm was similar with Grain-128 stream cipher algorithm, which was 2^{128}. Therefore, the samples used

for this research were 100 different keys that were chosen randomly for 1%–5% of significance level.

3.3 Research Tools

The instruments used in this research are as follows:

3.3.1 Programming software: C++ programming language

The C++ programming language was applied in order to obtain the following results:

- generate 100 samples × 128-bit different random keys used in Grain-128 and Modified Grain-128 stream cipher algorithms.
- generate 100 keystreams for both Grain-128 and Modified Grain-128 stream cipher algorithms using 100 × 128-bit different random keys.
- generate 100 output of LFSR *(fx)* for Grain-128 stream cipher algorithm using 100 × 128-bit different random keys.
- generate 100 output of NLFSR *(gx)* for Grain-128 stream cipher algorithm using 100 × 128-bit different random keys.
- generate 100 output of Boolean Function *(hx)* for Grain-128 stream cipher algorithm using 100 × 128-bit different random keys.

3.3.2 Statistical analysis: NIST Statistical Test Suite

As mentioned in Chapter 2, in NIST Statistical Test Suite, there are 16 types of tests, each with its own focus and purpose. All the tests were

applied for all types of data, for both Grain-128 and Modified Grain-128 stream cipher algorithms.

3.3.3 Hardware: Intel® Core ™ Duo CPU P8700 @ 2.53GHz, 1.59 GHz, 2.96 GB of RAM.

3.4 Experimental Setup

The data used in this research was a primary data which was obtained from the experiment and observation.

As mentioned before, four types of data were produced and used in this research, which are keystream, output from LFSR *(fx)*, output from NLFSR *(gx)*, and output Boolean Function *(hx)*. Since the samples used in this research were 100 samples, each data produced 100 sequences. All the produced sequences are listed below.

Grain-128 stream cipher algorithm

1. 100 sequences of keystream; each sequence with 1 million (1,000,000) bits.
2. 100 sequences of output of LFSR; each sequence with 1 million (1,000,000) bits.
3. 100 sequences of output of NLFSR; each sequence with 1 million (1,000,000) bits.
4. 100 sequences of output of Boolean Function; each sequence with 1 million (1,000,000) bits.

All the sequences above were obtained from Grain-128 stream cipher algorithm.

Modified Grain-128 stream cipher algorithm

1. 100 sequences of keystream; each sequence with 1 million (1,000,000) bits.

All the sequences above were obtained from Modified Grain-128 stream cipher algorithm.

All the data listed above were collected by generating Grain-128 and Modified Grain-128 using C++ programming language. The generation of data for both algorithms is illustrated in Figure 11 and Figure 12, respectively.

Figure 11: Generating the data for Grain-128

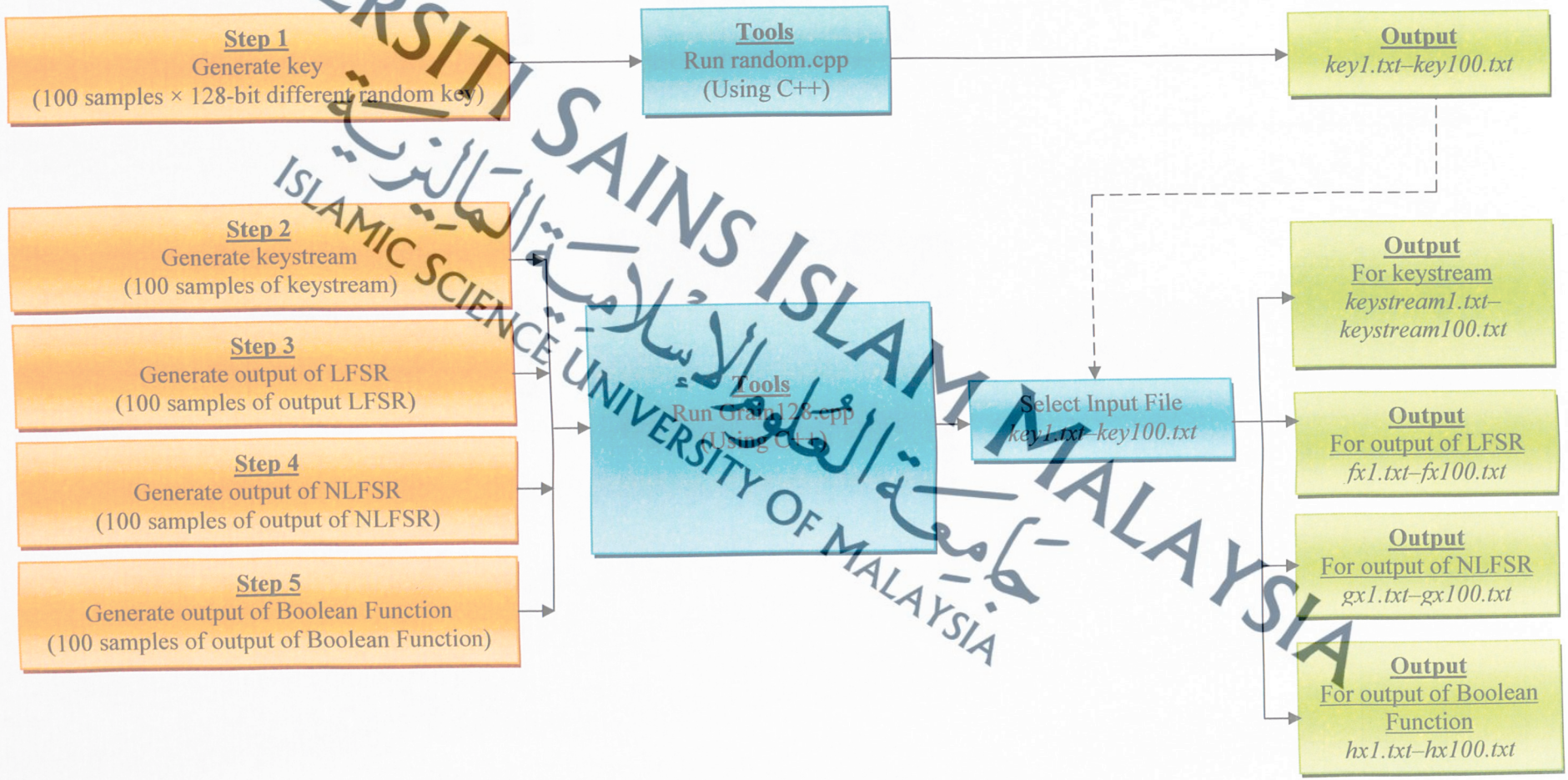
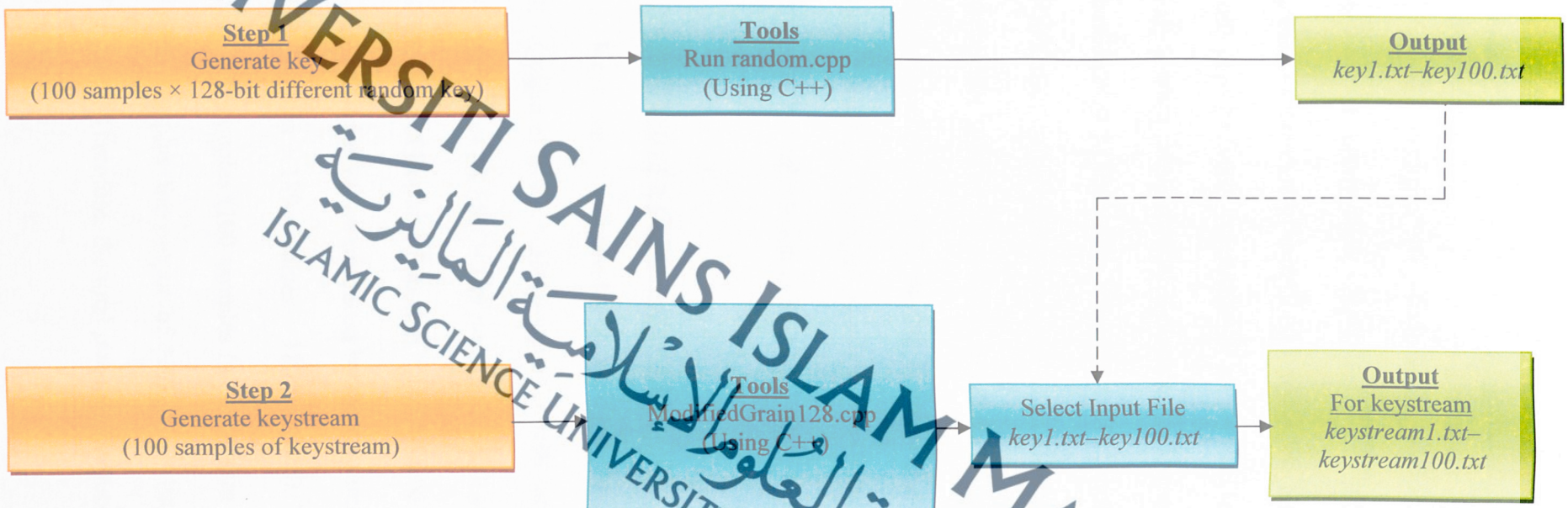


Figure 12: Generating the data for Modified Grain-128



To perform the randomness analysis, data was collected from the experiment using NIST Statistical Test Suite tool. The four types of data generated previously were inserted into the NIST Statistical Test Suite tool. The tool produced test value known as *p*-value, as described in Chapter 2. The *p*-values were then collected and analysed. These values were used to determine the randomness of each data, whether it was random or non-random. There were 16 NIST test values produced, each with a different number of *p*-values. Table 7 shows the number of *p*-values obtained for all 16 statistical tests for each data used. Based on Table 7, the total number of *p*-values produced for each data was 189. Therefore, the total number of *p*-values produced in this research was 321,300. The calculation to obtain all the *p*-values is as follows:

Grain-128 stream cipher algorithm

1. Keystream = 189 *p*-value for each sample.
2. Output of LFSR = 189 *p*-value for each sample.
3. Output of NLFSR = 189 *p*-value for each sample.
4. Output of Boolean Function = 189 *p*-value for each sample.

Total *p*-values produced by the four data above were 756 (189 *p*-values + 189 *p*-values + 189 *p*-values + 189 *p*-values). Total samples used in this research were 400 samples (100 samples for keystream + 100 samples for output of LFSR + 100 samples for output of NLFSR + 100 samples for output of Boolean Function). Therefore, the total *p*-values produced for Grain-128 were 75,600 (756 × 100).

Modified Grain-128 stream cipher algorithm

1. Keystream = 189 *p*-values for each sample.

Total *p*-values produced were 189. Total samples used in this research were 100 samples. Therefore, the total *p*-values produced for Modified Grain-128 were 18,900 (189 × 100).

From the calculations above, the total *p*-values produced from both Grain-128 and Modified Grain-128 were 94,500 (18,900 + 75,600).

Table 7: Number of *p*-values obtained per sample

Statistical Test	Number of *p*-value produced			
	Keystream	Output LFSR	Output NLFSR	Output Boolean Function
Frequency Test	1	1	1	1
Runs Test	1	1	1	1
Longest Runs of Ones Test	1	1	1	1
Spectral DFT	1	1	1	1
Lempel-Ziv Complexity Test	1	1	1	1
Cumulative Sums Test	2	2	2	2
Random Excursion Variant Test	18	18	18	18
Random Excursion Test	8	8	8	8
Binary Matrix Rank Test	1	1	1	1
Block Frequency Test	1	1	1	1
Non-Overlapping Test	148	148	148	148
Overlapping Test	1	1	1	1
Maurer's Universal Test	1	1	1	1
Linear Complexity Test	1	1	1	1
Serial Test	2	2	2	2
Approximate Entropy Test	1	1	1	1
Total	**189**	**189**	**189**	**189**

The process of conducting the statistical test for each data is illustrated in Figure 13.

Figure 13: The process of conducting the statistical test

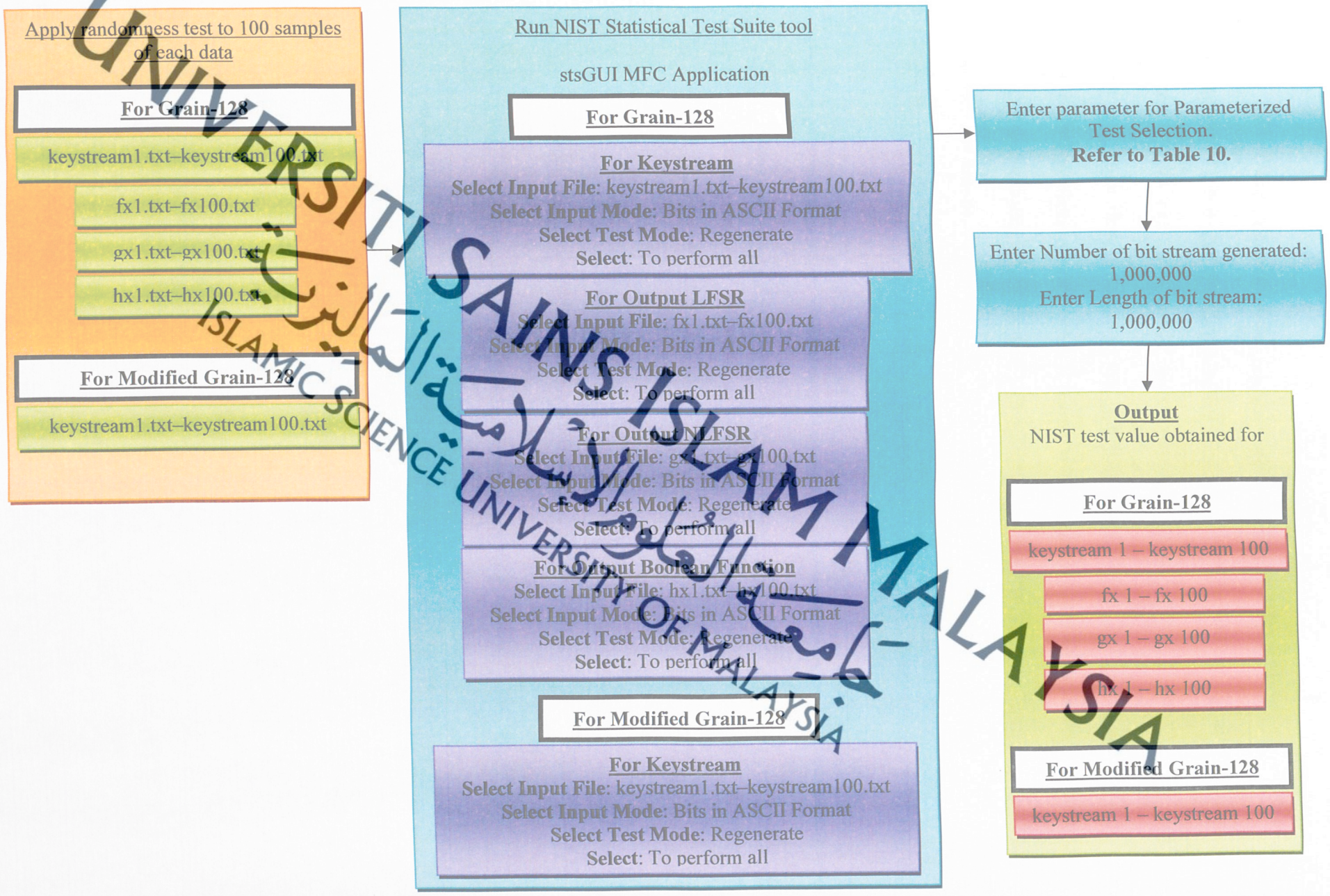


Table 8 and Table 9 show the number of minimum bits of sequence length (n) required for each test and the parameter considered for Parameterized Test Selection, respectively. The sequence tested was only considered as random if the p-value was more than or equal to the significance level used (α). For this research, the significance level used was chosen in the range of 1%-5% or equal to 0.01–0.05. During analysis, the p-value of less than the significance level was observed. The number of rejected p-value later determined the randomness of the sequence.

Table 8: Number of minimum bits of sequence required

Non-Parameterized Test Selection		
Test	**Minimum Requirement**	**Used in Research**
1. Frequency Test	$n \geq 100$	1,000,000
2. Runs Test	$n \geq 100$	1,000,000
3. Longest Runs of Ones Test	Length of each block, M was chosen in accordance to minimum n. These were fixed in the test code. **minimum *n*** — ***M*** 128 — 8 6,272 — 128 750,000 — 10,000	1,000,000
4. Spectral DFT	$n \geq 1{,}000$	1,000,000
5. Lempel-Ziv Complexity Test	$n \geq 10^6$	1,000,000
6. Cumulative Sums Test	$n \geq 100$	1,000,000
7. Random Excursion Variant Test	$n \geq 10^6$	1,000,000
8. Random Excursion Test	$n \geq 10^6$	1,000,000
9. Binary Matrix Rank Test	$n \geq 38MQ \geq 38(32)(32) \geq 38{,}912$ $M = Q = 32$ where M and Q were the number of rows and columns in each matrix (fixed in the test code)	1,000,000
Parameterized Test Selection		
1. Block Frequency Test	$n \geq 100$	1,000,000
2. Non-Overlapping Test	Not specific	1,000,000
3. Overlapping Test	$n \geq 10^6$	1,000,000
4. Maurer's Universal Test	$n \geq 387{,}840$	1,000,000
5. Linear Complexity Test	$n \geq 10^6$	1,000,000
6. Serial Test	Not specific	1,000,000
7. Approximate Entropy Test	Not specific	1,000,000

Table 9: Parameter value(s) required for Parameterized test selection

Statistical test	Requirement	Parameter(s)
Block Frequency Test	$N = n/M$ & $N < 100$ $n \geq MN$ $M \geq 20$ & $M \geq 0.01n$ Where n = bit sequence N = number of blocks	Block length, M
Overlapping Templates Test	$n \geq MN$ $M > 0.01n$ N should be chosen so that $N.(\min \pi i) > 5$ $\lambda = (M - m + 1) / 2^m \approx 2$ m should be chosen so that $m \approx \log_2 M$ Choose K so that $K \approx 2\lambda$. Where n = bit sequence M = the length bits of a substring NIST recommends to choose m = 9 or 10	Template length, m
Non-Overlapping Templates Test	$N \leq 100$ & $N = n/M$ $M > 0.01n$ where n = bit sequence N = number of independent blocks = n/M = 8 (fixed in the test code) M = the length bits of a substring NIST recommends to choose m = 9 or 10	Template length, m
Serial Test	Block length, m should be selected such that $m < [\log_2 n] - 2$ where n = bit sequence	Block length, m
Approximate Entropy Test	Block length, m should be selected such that $m < [\log_2 n] - 5$ where n = bit sequence	Block length, m
Linear Complexity Test	$500 \leq M \leq 5000$ $N \geq 200$ where n = bit sequence N is number of independent blocks = n/M	Block length, M
Universal Test	$6 \leq L \leq 16$ $Q = 10 * 2^L$ The values of L, Q, and n should be chosen as below. **minimum 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 $\geq 10,342,400$ / 10 / 10240 $\geq 22,753,280$ / 11 / 20480 $\geq 49,643,520$ / 12 / 40960 where n = bit sequence	1. Number of blocks, Q 2. Block length, L

For every significance level selected, the maximum number of binary sequences that was expected to be rejected must be computed by using the following formula:

$$s\left(\alpha + 3\sqrt{\frac{\alpha(1-\alpha)}{s}}\right)$$

Where s is the sample size used and α is the significance level used. This test used 100 samples for each type of data for both Grain-128 and Modified Grain-128 stream cipher algorithms, respectively.

Because the total number of p-value produced for each test was different, it would affect the calculation of maximum number of rejection rate. For example, in the Frequency Test, the number of p-value produced was one (1), the samples used were 100 samples and the significance level used was 1% (0.01). Therefore, by using the above formula, the total number of rejection rate should not exceed 3 sequences.

However, for Non-Overlapping Template Test, the p-values produced were 148. This research used 100 samples, so the total number of p-values produced for this test was 14,800. Therefore, to calculate the maximum number of rejection rate, the sample size, s should become 14,800.

For Serial Test and Cumulative Sums Test, these tests produced 2 p-values, respectively. Therefore, the total number of p-values for each test using 100 samples was 200. Hence, the sample size, s should become 200 to compute the maximum number of rejection rate.

Lastly, for Random Excursion Variant and Random Excursion Tests, these tests produced 18 *p*-values and 8 *p*-values, respectively. However, only samples with the number of cycles exceeding 500 were evaluated.

3.5 Summary

The expected outcomes of this research are as follows:

1. The randomness of Grain-128 stream cipher algorithm can be determined using statistical analysis.
2. The Modified Grain-128 stream cipher algorithm can be constructed based on statistical analysis obtained from Grain-128.
3. The results of statistical analysis of Modified Grain-128 can be compared with Grain-128. Modified Grain-128 stream cipher algorithm with the sufficient results of statistical analysis can be proposed to be used for future encryption application.