A New Proposed Design of a Stream Cipher Algorithm: Modified Grain - 128

Norul Hidayah Lot @ Ahmad Zawawi
Cryptography Development
CyberSecurity Malaysia
Email: norul {at} cybersecurity.my

Kamaruzzaman Seman
Faculty of Science and Technology
Islamic Science University of Malaysia
(USIM)Negeri Sembilan, Malaysia

Nurzi Juana Mohd Zaizi
Faculty of Science and Technology
Islamic Science University of Malaysia
(USIM)
Negeri Sembilan, Malaysia

Abstract —**The objective of this research is to propose a new algorithm based on the existing Grain - 128 stream cipher algorithm. The comparison of Grain - 128 and Modified Grain - 128 will be evaluated by using NIST Statistical Test Suite. The NIST Statistical Test Suite is conducted to determine the randomness of both algorithms. Conclusively, the Modified Grain - 128 is random based on 1% of significance level compared to the Grain - 128 which is not random at the same significance level.**

Keywords-component; Grain - 128, stream cipher algorithm, NIST Statistical Test Suite, statistical randomness testing, significance level

I. INTRODUCTION

Currently, stream cipher algorithm is still a choice to be used either in software or hardware. It is because of the main advantage in stream cipher algorithm which will be designed to allow faster keystream generation in software. Besides that, it may also be designed in a smaller size for space requirement in hardware. Therefore, the stream cipher will be an interesting algorithm which is faster in software or smaller in hardware [1]. Consequently, the stream cipher is particularly relevant for specific applications with little computational resources such as cell phone and other small embedded devices.

One of the important criteria in evaluating a stream cipher algorithm is the suitability of the algorithm to act as a random number generator [2] to achieve randomness level. Hence, statistical analysis using randomness test will determine whether the stream cipher is fulfill the qualification requirement [3].

Grain – 128 is one of the stream ciphers algorithm which is very well suited for hardware with good environment conditions target for minimal resources such as gate count, power consumption and chip area [3]-[4]. According to previous work by [1], they stated that there is no 128 bit cipher offering the same security as Grain – 128 stream cipher algorithm. However, there are several attacks that have been done against Grain – 128 since 2006 until 2011 which showed that this algorithm still has weakness. The cryptanalysis attacks suffered by Grain – 128 such as linear approximation [1]-[4], algebraic attack [1]-[4]-[5]-[6], time – memory – data trade off attack [1]-[4] , fault attack [1]-[4]-[7]-[8], distinguishing attack [9]-[10], key – recovery attack [9] , chosen – IV attack [4] , slide attack [11] , differential attack [11] , related – key chosen attack [12] , correlation attack [6] , self – sliding attack [13] , cube attack [14] , and dynamic cube attack [14].

In this paper, a new stream cipher algorithm have been proposed based on the existing Grain - 128 stream cipher algorithm; Modified Grain - 128 stream cipher algorithm to improve the current algorithm.

A short description of Grain - 128 stream cipher algorithm is described in Section II. Meanwhile, in Section III explains the proposed of modification of Grain - 128 stream cipher algorithm in detail. In Section IV, the comparison between Grain - 128 and Modified Grain - 128 are demonstrated. The experimental setup and result and analysis are respectively discussed in Sections VI and VII. Conclusion of this research are finally illustrated in Section VIII.

II. A SHORT DESCRIPTION OF GRAIN-128 STREAM CIPHER ALGORITHM

Grain – 128 stream cipher algorithm was introduced by Hell, Johansson, Maximov and Meier in 2006 [1]-[15]. 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 Linear Feedback Shift Register (LFSR), Non – Linear Feedback Shift Register (NLFSR) and Output Boolean Function. The description of Grain – 128 stream cipher algorithm. Fig. 1 and Fig. 2 show the process of key initialization and keystream generating in stream cipher algorithm, respectively.

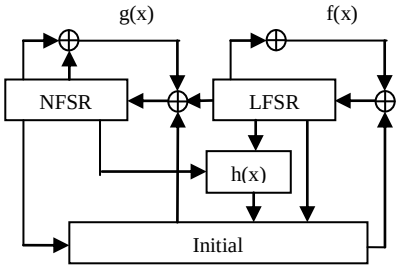

Figure 1: The process of key initialization

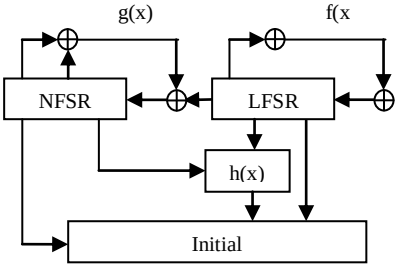

Figure 2: The process of keystream generating

A. Linear Feedback Shift Register (LFSR)

The Linear Feedback Shift Register (LFSR), *f(x)*, is a primitive polynomial of degree 128.It is defined as below;

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

It consists of 128 - bits. The content of LFSR is denoted as $s_t,\ldots,s_{t+127}$. This building block will be updated for each clock by the equation below;

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

B. Non - Linear Feedback Shift Register (NLFSR)

The NLFSR, *g(x)*, is the sum of a linear function and a bent function. 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 $b_t,\ldots,b_{t+127}$. This building block will be updated for each clock by the equation below;

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

C. Output Boolean Function

Output Boolean Function consists of 9 – input filter function taken from 7 – bit of input from LFSR and 2 – bit of input from NLFSR. The degree of this function is 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

In order to generate a keystream, the cipher must be initialized with the key and IV as a first step. To construct LFSR, the first 96 – bits of this building block are loaded with 96 – bits IV, whereas, the last 32 bits of the LFSR are filled with 1s. To construct NLFSR, 128 bits of this building block are loaded with 128-bits key. Process of generating the cipher in key initialization will be clocked until 256 times. After the cipher is clocked 256 times, the keystream has been generated.

III. THE PROPOSED OF MODIFICATION OF GRAIN-128 STREAM CIPHER ALGORITHM

The modification against Grain - 128 stream cipher algorithm has been done to produce new algorithm which is known as Modified Grain - 128 stream cipher algorithm. The structure and flow of Modified Grain - 128 is still similar with Grain - 128, whereas three main building blocks are used consist of Linear Feedback Shift Register (LFSR), Non - Linear Feedback Shift Register (NLFSR) and Output Boolean Function. However, there are several functions which have been changed to strengthen of the existing algorithm. The experimental setup for the Modified Grain - 128 will be discussed. Lastly, the result and analysis from the study will be demonstrated in the last section of this paper.

A. Linear Feedback Shift Register (LFSR)

Modified Grain - 128 stream cipher algorithm uses five Linear Feedback Shift Registers (LFSRs) namely as $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$ with size 37, 31, 16, 19 and 25, respectively. All the five LFSRs are primitive polynomial. Below are the lists of LFSRs used in Modified Grain - 128.

- $LFSR_1 = f_1(x) = 1 + x^{25} + x^{27} + x^{35} + x^{37}$
- $LFSR_2 = f_2(x) = 1 + x^{24} + x^{31}$
- $LFSR_3 = f_3(x) = 1 + x^2 + x^4 + x^5 + x^8 + x^9 + x^{12} + x^{13} + x^{15} + x^{16}$
- $LFSR_4 = f_4(x) = 1 + x^9 + x^{14} + x^{15} + x^{17} + x^{19}$
- $LFSR_5 = f_5(x) = 1 + x^{21} + x^{22} + x^{25}$

All the five LFSRs will be updated for each clock that will be explained later in the next section.

B. Non - Linear Feedback Shift Register (NLFSR)

Modified Grain - 128 stream cipher algorithm uses the same NLFSR as in Grain - 128 stream cipher algorithm. However, the NLFSR will be updated for each clock with different setting. It– will be explained later in the next section. The NLFSR used is as follows:

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

C. Boolean Function

For the Boolean Function, it uses the same Boolean Function as in Grain - 128 stream cipher algorithm. However, the input function taken is different. 4 - bit of inputs are taken from NLFSR and 1 - bit input is taken from each of $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$ respectively. The function used is as follows:

$$h(x) = b_{i+12}s1_i + b_{i+13}s2_i + b_{i+95}s3_i + b_{i+60}s4_i + b_{i+12}b_{i+95}s5_i$$

where $b_{i+12}, b_{i+13}, b_{i+95}, b_{i+60}$ are taken from NLFSR and $s1_i, s2_i, s3_i, s4_i, s5_i$ are respectively taken from $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$.

D. Keystream

In order to generate the keystream of Modified Grain - 128, the cipher must be firstly initialized with the key and IV. To construct all the five LFSRs, the first 37 - bit of key are loaded for $LFSR_1$. For the second LFSR, the 38th - bit until 68th - bit of key are loaded for $LFSR_2$. It is continued with the third LFSR, where the 69th - bit until 84th bit of key are loaded for $LFSR_3$. For $LFSR_4$, it is taken from 85th - bit until 103th - bit of key. Lastly, the rest bit of key are loaded for $LFSR_5$. To construct NLFSR, the first 96 - bit of NLFSR are loaded with 96 - bits IV. Whereas, the last 32 - bits of the NLFSR are filled with 1s.

The structure of Modified Grain - 128 is illustrated in Fig. 3 and Fig. 4. Fig. 3 shows the process of key initialization of Modified Grain - 128 stream cipher algorithm. Meanwhile, Fig. 4 shows the process of generate the keystream of Modified Grain - 128 stream cipher algorithm.

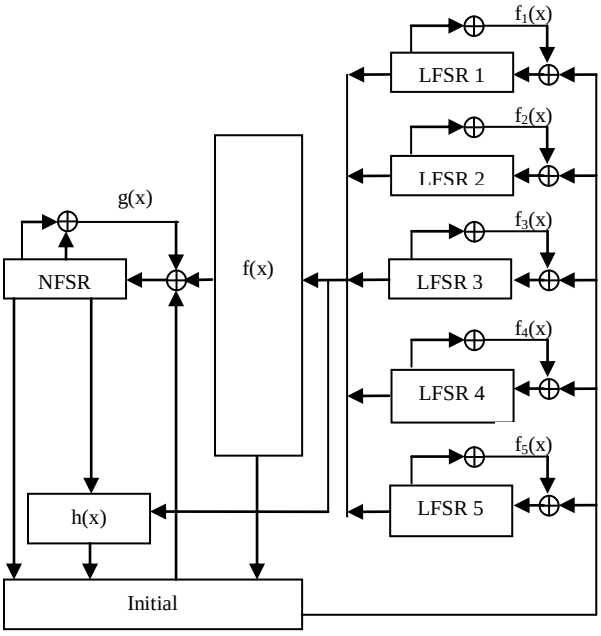

Figure 3: Key initialization process for Modified Grain – 128

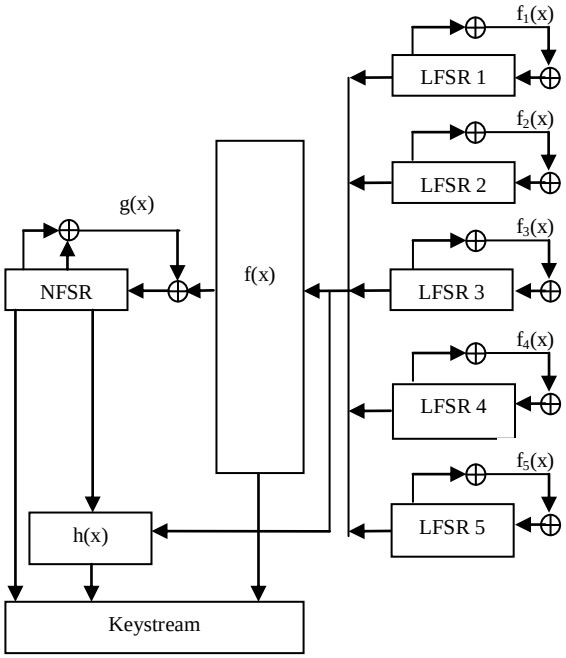

Figure 4: Generating of keystream process for Modified Grain – 128

E. Key Initialization of Modified Grain - 128 Process

In order to generate key initialization, the cipher must be firstly initialized with the key and the IV. The initialization of the key and the IV is done as follows:

Step 1 : To construct LFSR and generate the bit sequence from output of five LFSRs

In Step 1, the LFSRs are constructed by using the assigned key. Each LFSR is loaded with the 128 bits of the key. Fig. 5 shows the process of constructing the LFSR.

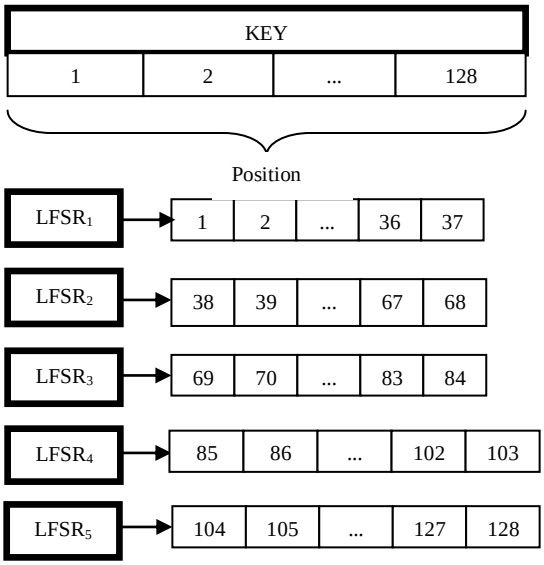

Figure 5: The process of constructing LFSRs

The bit sequence from output of five independent LFSRs, namely $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$ will be generated. For the configuration, the feedback tapping is based on the primitive polynomial used for each LFSR. Each LFSR will produce bit sequence namely S_1, S_2, S_3, S_4 and S_5 respectively.

Step 2: To construct NLFSR and generate the bit sequence from output of NLFSR

In Step 2, the NLFSR is constructed by using the IV. The first 96 bits of NLFSR are loaded with IV bits. Meanwhile, the last 32 bits of NLFSR are filled with 1s. Fig. 6 shows the process of constructing the NLFSR.

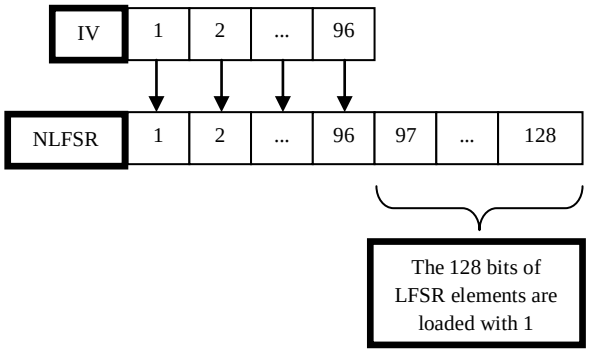

Figure 6: The process of constructing NLFSR

The NLFSR will be updated for each clock by setting

$$g(x)_{i+128} = f(x) + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}$$

Step 3: To obtain value of f(x)

To obtain the value of f(x) in Step 3, each bit sequence of LFSR will be XORED as the following below.

$$f(x) = S_1 + S_2 + S_3 + S_4 + S_5$$

Step 4: To obtain value of Boolean function, h(x)

In Step 4, 9 inputs are taken to obtain the value of h(x). 4 bit - input are taken from NLFSR and 1 - bit input are taken from each of $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$. It may be defined as

$$h(x) = b_{i+12}S_1 + b_{i+13}S_2 + b_{i+95}S_3 + b_{i+60}S_4 + b_{i+12}b_{i+95}S_5$$

where $b_{i+12}, b_{i+13}, b_{i+95}, b_{i+60}$ are taken from NLFSR and $s1_i, s2_i, s3_i, s4_i, s5_i$ are respectively taken from $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$.

Step 5: To obtain value of Initial

In Step 5, the initial value is obtained by applying XOR operation to the three bits of f(x), g(x) and h(x), where the output function can be defined as:

$$\boldsymbol{initial} = \sum_{j \in A} b_{i+j} + h(x) \oplus f(x)$$

where A = {2,15,36,45,64,73,89}

The initial is then will fed back and XOR with the input of NLFSR, $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$, respectively. The initial will be clocked 256 times before producing the keystream.

F. Generate the Keystream of Modified Grain - 128 Processes

For generating the keystream of Modified Grain - 128, the process of producing keystream are similar with the key initialization process from step 1 until step 4. However, in step 5, there is different in order to obtain the output of keystream, where the output of keystream is not fed back to the NLFSR, $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$.

IV. COMPARISON BETWEEN GRAIN-128 AND MODIFIED GRAIN-128

This section explains the comparison between Grain - 128 and Modified Grain - 128 stream ciphers. Each main building block used in both algorithms will be discussed which consist of Linear Feedback Shift Register (LFSR), Non - Linear Feedback Shift Register (NLFSR) and Boolean Function. In addition, the keystream for each algorithm will be discussed. The comparison between both algorithms is described in Table 1 below.

V. 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 on 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 normal standard 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 [3].

NIST Statistical Test Suite can be divided into two categories, which are Parameterized Test Selection and Non-Parameterized Test Selection. The Parameterized Test Selection requires user 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. Whereas, the Non-Parameterized Test Selection does not require user to enter any parameter in obtaining the p - value for each test. The tests are divided according to their categories as per listed below.

TABLE I : Comparison Between Grain - 128 and Modified Grain - 128 Stream Cipher Algorithms

	Grain - 128	**Modified Grain - 128**
Linear Feedback Shift Register	Grain - 128 uses 1 LFSR with 128 bit and it is primitive polynomial. The LFSR used can be defined as below: $f(x) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128}$ Then, the LFSR will be updated for each clock by setting: $s_{i+128} = s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}$	Modified Grain - 128 uses 5 LFSRs with each LFSR is primitive polynomial. The LFSRs are listed as below: $LFSR_1 = f_1(x) = 1 + x^{25} + x^{27} + x^{35} + x^{37}$ $LFSR_2 = f_2(x) = 1 + x^{24} + x^{31}$ $LFSR_3 = f_3(x) = 1 + x^2 + x^4 + x^5 + x^8 + x^9 + x^{12} + x^{13} + x^{15} + x^{16}$ $LFSR_4 = f_4(x) = 1 + x^9 + x^{14} + x^{15} + x^{17} + x^{19}$ $LFSR_5 = f_5(x) = 1 + x^{21} + x^{22} + x^{25}$ Then, the LFSR will be updated for each clock by setting: $f(x)_{i+128} = s1_i + s2_i + s3_i + s4_i + s5_i$
Non - Linear Feedback Shift Register	1 NLFSR with 128 bit is used in Grain - 128. The NLFSR used is the sum of one linear and one bent function. The NLFSR used can be defined as below: $g(x) = 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} + x^{63}x^{67} + x^{69}x^{101} + x^{80}x^{88} + x^{110}x^{111} + x^{115}x^{117}$ Then, this NLFSR will be updated for each clock by setting: $b_{i+128} = s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+94}$	NLFSR used is similar as in Grain - 128 which is as following: $g(x) = 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} + x^{63}x^{67} + x^{69}x^{101} + x^{80}x^{88} + x^{110}x^{111} + x^{115}x^{117}$ Then, this NLFSR will be updated for each clock by setting: $b_{i+128} = f(x) + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + b_{i+17}b_{i+18} + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}$
Output Boolean Function	The Boolean Function used consists of 9 – input filter function taken from 7 – bit of input from LFSR and 2 – bit of input from NLFSR. $h(x) = h(h_0,x_1,...x_8) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8$	The Boolean Function used consists of 9 input filter function taken from 4 - bit of input from NLFSR and 1 - bit input from each of $LFSR_1$, $LFSR_2$, $LFSR_3$, $LFSR_4$ and $LFSR_5$. $h(x) = b_{i+12}S_1 + b_{i+13}S_2 + b_{i+95}S_3 + b_{i+60}S_4 + b_{i+12}b_{i+95}S_5$
Keystream	The formula used to obtain the keystream is as follows: $z_i = \sum_{j\in A} b_{i+j} + h(x) \oplus s_{i+93}$ where A = {2,15,36,45,64,73,89}	The formula used to get the keystream is as follows: $z_i = \sum_{j\in A} b_{i+j} + h(x) \oplus f(x)$ where A = {2,15,36,45,64,73,89}

A. Parameterized Test Selection

- Block Frequency Test
- Overlapping Template Test
- Non-Overlapping Template Test
- Serial Test
- Approximate Entropy Test
- Linear Complexity Test
- Maurer's Universal Test

B. Non-Parameterized Test Selection

- Cumulative Sums (Forward/Reverse) Test
- Runs Test
- Longest Runs of Ones Test
- Binary Matrix Rank Test
- Spectral (Discrete Fourier Transform) Test
- Random Excursion Test
- Random Excursion Variant Test
- Lempel-Ziv Complexity Test
- Frequency Test

TABLE 2 : List of NIST Statistical Test Suite

NIST Statistical Test Suite	**Number of p - value**
Non - Parameterized Test Selection	
1. Frequency Test	1
2. Runs Test	1
3. Longest Runs of Ones Test	1
4. Spectral (Discrete Fourier Transform) Test	1
5. Lempel - Ziv Complexity Test	1
6. Cumulative Sums Test	2
7. Random Excursion Variant Test	18
8. Random Excursion Test	8
9. Binary Matrix Rank Test	1
Parameterized Test Selection	
1. Block Frequency Test	1
2. Non - Overlapping Template Test	148
3. Overlapping Template Test	1
4. Maurer's Universal Test	1
5. Linear Complexity Test	1
6. Serial Test	2
7. Approximate Entropy Test	1

VI. EXPERIMENTAL SETUP

The randomness test used is based on the application of the NIST Statistical Test Suite. Eleven out of sixteen tests provided only one p - value. While, two tests out of sixteen provided two p - values. The other three tests provided eight, eighteen and 148 p - values respectively. Table 2 above lists the number of p - value(s) obtained for each statistical test [15].

The description and requirement for each statistical test as per listed below, need to be considered prior to conducting the experiment.

The randomness testing activities are based on the application of the NIST Statistical Test Suite. Table 3 shows the requirement for parameter(s) value that need to be considered in conducting the experiment for the Parameterized Test Selection. 100 samples are generated for each algorithm in this study.

In this research, the significance level was fixed at 1% (0.01). The maximum number of rejection in binary sequences for each algorithm at the chosen significance level was computed using the formula (1)

$$s\,(\alpha + 3)\sqrt{\frac{\alpha(1-\alpha)}{s}} \qquad (1)$$

The maximum number of rejection rate should be as shown in Table 4. As evaluation, only 67 samples are tested for the Random Excursion Variant Test and the Random Excursion Test. It is because only these samples have the number of cycle exceeding 500. The numbers of cycles for the other 33 samples are not exceeding 500. Therefore, 33 samples with the number of cycle which is not exceeding 500 are not evaluated.

TABLE 3: Parameter value(s) required for Parameterized test selection

Test	Requirement	Used in Research
Block Frequency Test	$N < 100$	N = n/M = 1,000,000/20,000 = 50
	$n \geq 100$ and $n \geq MN$	n = 1,000,000 and n ≥ MN = 20,000 x 50 = 1,000,000
	$M \geq 20$ $M \geq 0.01n$	**M = 20,000 (Block Length)** M ≥ 0.01n = 0.01 x 1,000,000 = 10,000
Non – Overlapping Test	$n \geq 1{,}000{,}000$	n = 1,000,000
	$M \geq 0.01n$ = 0.01x 1,000,000 = 10,000	M = n/N = 1,000,000/100 = 10,000
	$N \leq 100$	N = 8 (fixed)
	NIST recommends to choose m = 9 or 10	**m = 10 (Template Length)**
Overlapping Test	$N \leq 100$	N = 8 (fixed)
	$M \geq 0.01n$ = 0.01x 1,000,000 = 10,000	M = n/N = 1,000,000/100 = 10,000
	n is not specific	n = 1,000,000
	NIST recommends to choose m = 9 or 10	**m = 10 (Template Length)**
Maurer's Universal Test	$6 \leq L \leq 16$	**L = 7 (Block Length)**
	$Q = 10 \times 2^L$	**Q =** $10 \times 2^L = 10 \times 2^7$ = **1,280 (Number of Block)**
	$n \geq 904{,}960$	n = 1,000,000
Linear Complexity Test	$500 \leq M \leq 5{,}000$	**M = 2,000 (Block Length)**
	$n \geq 1{,}000{,}000$	n = 1,000,000
	$N \geq 200$	N = n/M = 1,000,000/2,000 = 500
Serial Test	$m < [\log_2 n] - 2$	**m = 2 (Block Length)**
Approximate Entropy Test	$m < [\log_2 n] - 5$	m = 2 (Block Length)

TABLE 4: Number of maximum rejection for keystream with 1% of significance level

Number of maximum rejection		
Most of the NIST tests (based on 100 p-value)	3 samples	
Non-Overlapping (based on 14,800 p-value)	184 samples	
Random Excursion Variant	Grain – 128 (based on 64 samples)	21 samples
	Modified Grain – 128 (based on 67 samples)	22 samples
Random Excursion	Grain – 128 (based on 64 samples)	11 samples
	Modified Grain – 128 (based on 67 samples)	12 samples

VII. RESULT AND ANALYSIS

Table 5 shows the comparison NIST Statistical Test results between Grain - 128 and Modified Grain - 128 with 1% of significance level. From the result obtained, it is proven that there are 2 statistical tests failed in Grain - 128 stream cipher which are Lempel Ziv Complexity test and Linear Complexity test. Both tests have exceeded the maximum number of rejection with 1% of significance level, which are 5 and 4 respectively. Therefore, it can be concluded that the Grain - 128 is non - random for 1% of significance level.

Meanwhile, for Modified Grain - 128 stream cipher, it is shown that all 16 NIST Statistical Tests have passed the statistical test which is the number of rejection is still lower than the maximum number of rejection. Therefore, it can be

concluded that Modified Grain - 128 is random for 1% of significance level.

TABLE 5 : NIST result for number of rejection of Grain - 128 & Modified Grain - 128 with 1% of significance level.

Statistical Test	**Number of sequences at 1% significance level**			
	Grain - 128		**Modified Grain - 128**	
	Result	**Pass /Failure**	**Result**	**Pass /Failure**
Non-Parameterized Test Selection				
Frequency	0	Pass	0	Pass
Runs	0	Pass	0	Pass
Longest Runs of Ones	1	Pass	0	Pass
Spectral DFT	0	Pass	1	Pass
Lempel – Ziv Complexity	5	Failure	1	Pass
Cumulative Sums - Forward - Reverse	 0 0	 Pass Pass	 0 0	 Pass Pass
Random Excursion Variant	5	Pass	8	Pass
Random Excursion	3	Pass	10	Pass
Binary Matrix Rank	1	Pass	0	Pass
Parameterized Test Selection				
Block Frequency	1	Pass	0	Pass
Non – Overlapping	147	Pass	143	Pass
Overlapping	1	Pass	2	Pass
Maurer's Universal	1	Pass	0	Pass
Linear Complexity	4	Failure	1	Pass
Serial Test - P value 1 - P value 2	 0 0	 Pass Pass	 0 0	 Pass Pass
Approximate Entropy	0	Pass	0	Pass

VIII. CONCLUSION

In this research, we have presented a new stream cipher algorithm, Modified Grain - 128. Based on the result obtained from the experiment conducted, we can conclude that the keystream of Modified Grain - 128 stream cipher is pass for all 16 NIST Statistical Test. Therefore, the Modified Grain - 128 is random for 1% of significance level. In the future, this algorithm can be applied for the application with little computational resources such as for cell phone or other small embedded devices.

ACKNOWLEDGEMENT

We would like to acknowledge the help of Faculty of Science and Technology, Universiti Sains Islam Malaysia for supporting this research. A special thanks is also convey to CyberSecurity Malaysia for the guidance in completing this research.

REFERENCES

[1] M. Hell, T. Johansson, A. Maximov, and W. Meier, A stream cipher proposal: Grain – 128, Information Theory, IEEE International Symposium, 2006, pp. 1614-1618.

[2] Department of Commerce, Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard, Federal Register, The Daily Journal of the United States Government,12 September 1997.

[3] A. RukhinJ. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S.Vo, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22, 2010.

[4] M. Hell, T. Johansson, A. Maximov, and W. Meier, The Grain family of stream ciphers, New Stream Cipher Designs: The eSTREAM Finalist, LNCS 4986, 2008, pp. 179- 190.

[5] M. Afzal, and A. Masood, Algebraic cryptanalysis of a nlfsr based stream cipher, International Conference on Information & Communication Technologies: from Theory to Applications, ICTTA'08 IEEE, 2008.

[6] C. Berbain, H. Gilbert, and A. Joux, Algebraic and correlation attacks against linearly filtered non linear feedback shift registers, Selected Areas in Cryptography-SAC, Lecture Notes in Computer Science, R. Avanzi, L. Keliher, and F. Sica, Eds., Vol. 5381. Springer- Verlag, 2009, pp. 184-198.

[7] A. Berzati, C. Canovas, G. Castagnos, B. Debraize, L. Goubin, A. Gouget, P. Paillier, and S. Salgado, Fault Analysis of Grain-128, Hardware Oriented Security and trust, IEEE International Workshop, 2009, pp. 7-14.

[8] S. Karmakar, and D. R. Chowdhury, Fault Analysis of Grain – 128 by Targeting NFSR, AFRICACRYPT 2011, LNCS 6737,2011, pp. 298 – 315

[9] A. Maximov, Cryptanalysis of the "Grain" family of stream ciphers, ACM Symposium on Information, Computer and Communications Security (ASIACCS'06), 2006, pp. 283-288.

[10] S. Knellwolf, W. Meier, and M. N. Plasencia, Conditional Differential Cryptanalysis of NLFSR Based Cryptosystems, International Association for Cryptology Research, 2010, pp. 130 – 145.

[11] C. D. Cannière, O. Kucuk, and B. Preneel, Analysis of Grain's initialization algorithm, Progress in Cryptology – AFRICACRYPT 2008, Lecture Notes in Computer Science, Springer-Verlag, Vol. 50232008, pp. 276–289.

[12] Y. Lee, K. Jeong, J. Sung, and S. Hong, Related-Key Chosen IV Attacks on Grain-v1 and Grain-128, Y. Mu, W. Susilo, and J. Seberry (Eds.), ACISP 2008, LNCS 5107, 2008, pp. 321-335.

[13] H. Zhang, and X. Wang, Cryptanalysis of stream cipher Grain family, Cryptology ePrint Archive, Report 2009/109, 2009.

[14] I. Dinur, and A. Shamir, Breaking Grain-128 with dynamic cube attacks, Fast Software Encryption 2011, ser. To be published in Lecture Notes in Computer Science, A. Joux, Ed. Springer-Verlag,, 2011.

[15] Norul Hidayah Lot @ Ahmad Zawawi, Kamaruzzaman Seman and Nurzi Juana Mohd Zaizi, Randomness analysis on grain - 128 stream cipher, International Conference o Mathematical Sciences and Statistics 2013 (ICMSS2013).