Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

A Modular approach on Statistical Randomness Study of bit sequences

J K M Sadique Uz Zaman
SRF, Dept of Radio Physics and Electronics, University of Calcutta, Kolkata – 700009, India
Sangeet Saha
JRF, A. K. Choudhury School of IT, University of Calcutta, Kolkata – 700009, India
Ranjan Ghosh
Dept of Radio Physics and Electronics, University of Calcutta, Kolkata – 700009, India

----------**ABSTRACT**----------

Randomness studies of bit sequences, created either by a ciphering algorithm or by a pseudorandom bit generator are a subject of prolonged research interest. During the recent past the 15 statistical tests of NIST turn out to be the most important as well as dependable tool for the same. For searching a right pseudorandom bit generator from among many such algorithms, large time is required to run the complete NIST statistical test suite. In this paper three test modules are considered in succession to reduce the searching time. The module-1 has one program and is executed almost instantly. The module-2 has four programs and takes about half an hour. The module-3 has fifteen programs and takes about four to five hours depending on the machine configuration. To choose the right pseudorandom bits generator, the algorithms rejected by the first module are not considered by the second module while the third module does consider only those passed by the second module.

Keywords – **NIST test suite, P-value, PRBG, Randomness study, Statistical test.**

Date of Submission: 10 July, 2014 Date of Acceptance: 15 August, 2014

1. INTRODUCTION

Pseudorandom bit generator (PRBG) is required for many applications including stream ciphers in cryptography and thus the understanding of their random property is very important. In 1938, Kendall and Smith [1] put forwarded a classical discussion of a group of four statistical tests, namely (i) Frequency Test, (ii) Serial Test, (iii) Gap and Run Test and (iv) Poker Test, and these were used by Pathria [2 - 4] for testing the degree of randomness of numerical digits appearing in e, π, e^{-1} or π^{-1}. For a long time the group of four tests was an important tool to statisticians to study the random property of a number sequences. Menezes et. al. [5] mentioned five tests as basic statistical tests in which the Autocorrelation test is added over and above the four tests mentioned by Kendall and Smith and all these five tests used to consider bit sequences as an input. However, for testing cryptographic modules, the US-NIST declared in 1994 four tests as the standard statistical tests in which three are from Kendall and Smith and the rest is the long run test replacing the serial test and these four do consider also bit sequences as an input [5, 6]. During 2001, the US-NIST replaced the standard by a more comprehensive Statistical Test Suite [7] considering 16 tests – subsequently the Lempel-Ziv Compression test [8] was deleted and the two revised versions are presented considering 15 tests – one in 2008 [9, 10] and another in 2010 [11].

Since nineties many researchers conceived the idea of battery of statistical tests [5, 10] probably starting from Knuth [12]. During 2007 there was a parallel initiative from University of Montreal to develop a package TestU01 to test randomness of random number generator [13]. A new idea of using Walsh-Hadamard transform is recently coined in testing randomness of bit sequence [14]. The focus of attention is shifting from PRBG towards RBG [15] including its implementation in FPGA [16].

It may be noted that the NIST test suite for studying random property of bit sequences is dependable – its execution time for one bit sequence is tolerable, but for one algorithm is sincerely irritating. In order to have time advantage in testing the randomness of PRBG algorithms, three test modules to be used in succession are considered in this paper. In the first module, the frequency test at the byte level is carried out, in the second module the four tests proposed by Kendall and Smith are considered at the bit level and the latest NIST statistical test suite is the third module. From a group of PRBG algorithms, the second module considers those passed by the first and the third module takes those into account passed by the second.

In this paper the strategy and procedures adopted for the three test modules are described in Sec.2. The three random number generators are presented in Sec.3. Sec.4 discusses the results obtained from each test module. And finally the paper ends with a conclusion, drawn in Sec.5.

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

2. STRATEGY OF THE THREE TEST MODULES FOR STATISTICAL TESTS

The basic strategy is to develop a test tool based on three statistical test modules so that one can study the randomness of a large number of pseudorandom number generators within a reasonable time less than that taken by the NIST tests alone. The test module-1 is a new one which considers the frequency test at byte level and can be executed quickly. The test module-2 considers the frequency test, 2-bit serial test, runs test and 8-bit poker test – all at the bit level and adopts the methodology adopted in FIPS 140-1 to make a bit sequence to pass a particular test. The test module-3 is the NIST Tests Suite coded indigenously [17]. The main purpose of the paper is to use the three test modules in succession and to select a good PRBG(s). For this, six PRBGs namely RC4, PM, BBS-1S, BBS-3S, BBS-1L and BBS-4L are chosen and each one generates 300 random bit files each of 1342400 bits long.

There are two additional purposes in respect of the test module-1. First, to see if one divides a long bit sequence in few smaller blocks and can undertake the block frequency test and the second is to establish a definition of failure of the stipulated formalism of statistical randomness for smaller blocks which are derived from a long bit sequence tested practically random.

2.1 Test Module-1

The underlying idea behind the test module-1 is to note the fact that each of all the 8-bit 256 ASCII characters should appear uniformly all across the random bit sequence with probability of occurrences equal to p_n = 1/256. For a practical bit sequence, the probability of occurrences (p_i) of each of all the 8-bit characters can be calculated, based on which a testing scheme indicating a quantitative measure of randomness can be developed by defining square of the normalized standard deviation (σ) as the mean of the sum of the square of normalized probability deviation of all characters,

$$\sigma = \sqrt{\frac{1}{256}\sum_{i=0}^{255}\left(\frac{p_i - p_n}{p_n}\right)^2} \qquad (1)$$

For a non-random byte sequence, the value of σ^2 is expected to be large. If σ^2 is less than 0.01, one can consider the byte sequence as practically random.

Three tests are undertaken with the test module-1 for a particular bit sequence. The test-1 is developed with a purpose to use it for experiments to choose a suitable algorithm. The test-2 and test-3 are undertaken to see the additional purposes stated above.

2.1.1 Experimental Strategy

In test-1, it is intended to ensure that all the 8-bit characters are present across the entire N-byte sequence almost in equal number. All the characters C_i are counted for the entire N-byte sequence and $p_i = C_i/N$ is computed for i = 0 to 255. Subsequently σ^2 are calculated following eq. (1). The N-byte sequence is declared as practically random if the passing criterion of $\sigma^2 < 0.01$ is satisfied.

In test-2, it is ensured that there is uniformity of distribution of all the characters across the entire N-byte sequence. This is achieved by dividing the entire N-byte sequence into k equal blocks, each of which contains M (= N/k) bytes. For a particular j^{th} block, one counts all the characters C_i in one M-byte sequence, calculates $p_i = C_i/M$ for i = 0 to 255 and computes σ^2_j following eq. (1). Similarly σ^2_j for j = 1 to k are computed for all the k blocks and an average σ^2_{avg} is obtained. The values of k are taken as say 2, 4, 5, 8, 10, 16 and 20. If the passing criterion of $\sigma^2_{avg} < 0.01$ is satisfied for all values of k, the test-2 considers the N-byte sequence as practically random where all characters are uniformly distributed across the entire N-byte sequence.

In test-3 it is intended to derive a byte sequence of smaller length by dividing the long random N-byte sequence in k blocks taking k > 20 where the byte level randomization test is supposed to fail in the sense that at least one character out of 256 characters does not occur in any one of the k blocks. It is observed from experiment that the said byte-level randomization test fails almost for all types of PRBG bit sequences taken for k_{max} = 65 with corresponding M_{min} = 2560. Thus all N-byte sequences are successively divided by k with suitable values of k from 1 to 64. For all the M-byte sequences greater than M_{min}, the interest is to compute σ^2 and to plot σ^2_{min}, σ^2_{max} and σ^2_{avg} in three different colors against the different values of byte-length (M).

2.1.2 Experimental Results

The results and observation of the three tests stated above on fifteen set of data files are described below. The representative 15 data files each with 1342400 bits are taken as input, out of which the first 1331200 bits are selected and converted to 166400 bytes.

The test-1 has been undertaken for each 15 data files in succession and p_i and respective σ^2 are measured. It has been found that 10 data files are found to be random for which their σ^2 is found to be less than 0.01.

For the 10 files passed by test-1, the test-2 performed randomness test for k = 2, 4, 5 and 8 and it is found that for all the 10 files $\sigma^2_{avg} < 0.01$. This indicates that the smallest length of 20800 bytes may be considered as random if the longer file of 166400 bytes exhibits randomness.

Test-3 considered the same 10 files with increasing values of k and it is found that up to k = 64, each of all the blocks of length M-byte ensures the occurrences of all 256 characters with occurrence rate at least one. The values of σ^2_{max}, σ^2_{min} and σ^2_{avg} are computed for k = 1 to 64 with discrete intervals and are plotted in Fig.1.

2.1.3 A notable Observation

The bar chart for k = 1 in Fig.1, corresponds to the frequency test. The block frequency test data for k = 2 to 8 satisfy the stipulated random property, but are not shown in the figure. The data for k = 10 to 64 shown in Fig.1, indicate that σ^2 varies from 0.02 to 0.12 showing non-randomness. For larger values of k beyond 64, the situation

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

would be such that, one character or sometimes more characters are not occurring in each of all the M-byte blocks. The notable observation is the fact that such smaller blocks derived from practically random longer data set, are to be declared as non-random – which is a difficult proposition. It seems that a different statistical approach is needed to be taken for such smaller data files in order to evaluate their random property. There is a need for studies of random property of ciphers of smaller lengths in view of emerging embedded crypto systems.

of occurrences of 00, 01, 10 and 11 in a bit sequence are approximately the same as would be expected for a random bit sequence. The 8-bit Poker test sees if the non-overlapping frequencies of 2^8 types of 8-bit patterns appearing in a long bit sequence are approximately same as would be for a random bit sequence.

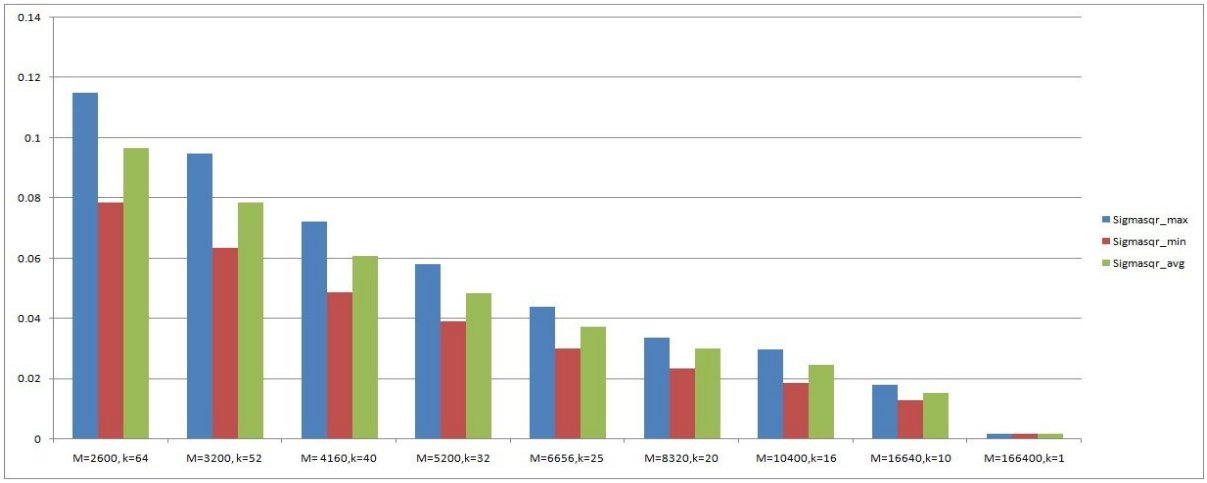

Figure 1. Bar chart for σ^2 vs k and M

2.2 Test Module-2

The underlying idea of the test module-2 is to repeat the four tests of Kendall and Smith [1] considering the input data as sequences of bits instead of decimal numerical digits and imposing in it the idea of chi-square (χ^2) distribution for different degrees of freedom (ν) following FIPS 140-1 [5, 6]. The idea of chi-square (χ^2) distribution function, $P(\chi^2, \nu)$ given in standard literature [5] is essentially a probability function which estimates significance level α for given χ^2 and ν. These values are tabulated in [5]. The meaning of these data χ^2, ν and α can be stated as, $P(x > \chi^2) = \alpha < 1$, if x is a random variable having χ^2 distributions with ν degrees of freedom. Alternatively, one can also state that, if x is a χ^2 distribution with ν degrees of freedom and x exceeds χ^2, then 100 times α samples are rejected for an experiment on 100 samples. In this paper the first 13424 bits have been taken out of the 1342400 bits for one bit sequence – such 300 random samples are taken for each algorithm. The passing criterion for each test on 300 samples is to calculate x-value and to see how many files have x-value $\leq \chi^2$-value given in [5] for $\alpha = 0.05$ and the value ν for a particular test. This means that a particular test should not reject more than 5% of the random data samples obtained from an algorithm supposed to be considered as random.

The four tests are frequency test, 2-bit serial test, 8-bit Poker test and runs test. In frequency test one has to ascertain if the numbers of 0s and 1s are approximately equal as would be for a random sequence. The purpose of the 2-bit serial test is to see if the overlapping frequencies

The runs test considers the number of runs, both of zeros and of ones, of various lengths from 1-bit to 9-bit and sees if these are approximately the same as would be for a random bit sequence.

For each of four tests if the desired situation is not met with, the chi-square deviation (x-value) is computed for all tests. The degrees of freedom of all the four tests are 1, 2, $2^8 - 1 = 255$ and $2\times9 - 2 = 16$ respectively. The significance level α is already set at 0.05 and the corresponding upper value of χ^2 for each of the four tests are 3.8415, 5.9915**,** 293.2478 and 26.2962 respectively following the table given in [5] for respective degrees of freedom.

2.3 Test Module-3

The motivation of the test module-3 is to indigenously code all the NIST statistical tests. It may be noted that the NIST module consists of 15 tests including the Frequency test, Serial Test and Runs test considered in the previous test module-2. The methodology of computing all the NIST tests is reported elsewhere [17]. A new algorithm is presented in NIST Special Publication 800-22 [11] in which the χ^2-value coupled with the degrees of freedom is transformed to a P-value instead of computing the probability value using the χ^2-distribution function. Thereby it sets a passing criterion; P-value $\geq$ 0.01 (significance level). Using all the P-values obtained for a particular test, NIST also mentioned a statistical procedure to compute Proportion of Passing which indicates uniformity or non-

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

uniformity of P-values. The minimum length of a bit-sequence required for any particular test as recommended by NIST is given below.

Name of the fifteen tests are mentioned where the minimum length for each test as recommended by NIST are given within the parenthesis following the name of the test. Test No.1: Frequency Test (100), Test No.2: Block Frequency Test (9000), Test No.3: Runs Test (100), Test No.4: Longest Run of Ones in a Block Test (128), Test No.5: Binary Matrix Rank Test (38912), Test No.6: Discrete Fourier Transform Test (1000), Test No.7: Non-overlapping Template Test (1048576), Test No.8: Overlapping Template Test (1000000), Test No.9: Maurer's "Universal Statistical" Test (1342400), Test No.10: Linear Complexity Test (1000000), Test No.11: Serial Test (1000000), Test No.12: Approximate Entropy Test (100), Test No.13: Cumulative Sums (Cusum) Test (100), Test No.14: Random Excursions Test (1000000), Test No.15: Random Excursions Variant Test (1000000). In test module-3 a bit length of 1342400 is taken for all tests except test nos. 6 and 13 for which said bit length is 13424. Each test considers 300 bit sequences randomly generated by a particular algorithm.

In estimating the degree of randomness of an algorithm, the Threshold value (T_{value}) and P-value of P-values (POP) are considered as two important checking parameters. These are explained in sub-sections 2.3.1 and 2.3.2 respectively.

2.3.1 Computation of Proportion of Passing of a particular test based on its P-values

To estimate the Observed Proportion of Passing (OPOP) of a particular test, it is necessary to consider large number of samples of bit-sequences randomly generated by an algorithm. If *m* samples of bit-sequences obtained from an algorithm are tested by a test producing one P-value, then the statistical average of T_{value} would be,

$$T_{value} = (1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{m}} \qquad (2)$$

Here significance level (α) = 0.01. The size of *m* should be greater than inverse of α. If *m* = 300, T_{value} = 0.972766. This means that such a test is considered statistically successful, if at least 292 P-values out of the 300 P-values do pass the test. If any test produced *n* number of P-values, then to calculate T_{value} in equation (2), one should consider $m \times n$ instead of *m*. With same values of α and *m*, the T_{value} is 0.983907 for *n* = 8 (test no. 14). Such a test is considered statistically successful if at least 2362 P-values out of the total 300×8 = 2400 P-values do pass the test. The status for Proportion of Passing a particular test would be a success if OPOP is greater than the corresponding T_{value}.

2.3.2 Computation of distribution pattern of P-values of a particular test

One can have an understanding about uniform or non-uniform distribution of P-values of a particular test from the series of obtained P-values. The P-values for a particular test are noted in 11 sub-intervals between 0 and 1 in four Tables, Table 3(a) through Table 3(d). For estimating χ^2-deviation of distribution of P-values, the first two P-values are merged in one group and the rest in 9 groups, thereby considering 10 groups of P-values. The χ^2-deviation of distribution of P-values is computed as,

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(s_i - \frac{m}{10}\right)^2}{m/10} \qquad (3)$$

where, S_i is the number of P-values in a group *i*, and *m* is the sample size. If a particular test produces *n* number of P-values, then $m = n\times$(sample size). Here the degrees of freedom ν = 9. The two parameters in the gamma function, $\Gamma(a,x)$ are taken as, a = ν/2 and x = χ^2/2 and the corresponding POP is obtained as,

$$POP = 1 - \frac{\Gamma(a,x)}{\Gamma(a,\infty)} \qquad (4)$$

The P-values are considered as uniformly distributed if eq.(4) provides POP ≥ 0.0001.

3. THREE ALGORITHMS: RC4, PM AND BBS

In reality there is no algorithm to produce pure random numbers. All the algorithms are deterministic in nature. And the output sequences produced by them are not statistically random, rather they are called pseudorandom. There are different types of pseudorandom generators like Linear congruential, Quadratic congruential, etc. A Linear congruential as well as a Quadratic generator along with RC4 is discussed in this Section.

3.1 RC4 algorithm

RC4 is a widely used stream cipher – its algorithm is very simple [18, 19]. Though it was invented by Ron Rivest in 1987, it is still of interest to modern cryptographers [20 – 24] to analyze its weaknesses and to improve its performance. It is intended that it generates random stream of key bytes which are XORed with stream of text bytes producing stream of random cipher bytes.

RC4 design involves an idea coined by Knuth [12] which states that a series of random numbers can suitably be generated if number elements of a reasonably large linear matrix are randomly shuffled for a number of times. Ronald Rivest translated the concept in two stages, KSA (Key Scheduling Algorithm) and PRGA (Pseudo Random Generator Algorithm). In KSA an identity S-Box is chosen with indices as element values. The elements are shuffled 256 times considering a role of the given key. In PRGA two elements are randomly chosen and shuffled and the two together gives one byte – the infinite continuation of the process randomly generates stream of bytes, termed as key stream.

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

3.2 Park and Miller (PM) algorithm

In 1988, S. K. Park and K. W. Miller proposed a linear congruential algorithm [25] to choose a sequence of random decimal integers. In this paper the algorithm is called as PM algorithm. The PM algorithm involves choice of large prime integer D (say $2^{31} - 1$) and another integer A close to $D^{1/2}$ and a choice of an initial seed number ($seed_0$). The seed is continuously upgraded by the remainder obtained by dividing (A×seed) by D.

Problem of the PM algorithm is that once the remainder is zero all remainders become zero and the algorithm fails. The said shortcoming can be handled by choosing a suitable large number as MASK followed by some programming technique so that zero never occurs.

3.3 BBS algorithm

In 1986, L. Blum, M. Blum and M. Shub [26] proposed an unpredictable PRBG which is called as BBS algorithm in present paper. The BBS algorithm is a popular and well known PRBG notionally providing perhaps the highest security. In BBS one has to choose two large prime integers p and q both congruent to 3 modulo 4 and a large number as seed. The seed is continuously upgraded by the remainder obtained by dividing square of the seed by n where n is the product of p and q and the LSB of the upgraded seed is collected –the infinite continuation of the process generates a pseudorandom bit sequence.

It is widely believed that there is no polynomial time Monte Carlo algorithm for Composite Quadratic Residues with small error probability [27]. The BBS generator follows the Composite Quadratic Residue property and this provides some evidence that BBS generator has provable security. But it is considered as a slow algorithm as it produces only single bit at a time. To make faster execution of the BBS generator it is proved that $r \leq \log_2\log_2(n)$ number of bits can be extracted at a time where n is the modulus used in the algorithm [27]. The generator will be faster if $r \geq 2$. Now if one intends to extract 4 bits, that is, r = 4 then $n \geq 2^{2^r} = 2^{16} = 65536$. In this paper two sets of p and q are considered for statistical tests. In first set n value is small with p = 131, q = 499 and the second set has large n value with p=42839, q=50123. With small n value and taking 1 bit in single iteration in BBS, BBS-1S algorithm is defined. Similarly taking 3 bits together one can define BBS-3S algorithm with small n value. With large n > 65536 one can extract 1 bit and also 4 bits in single iteration and can define BBS-1L and BBS-4L respectively as two more algorithms.

4. RESULTS AND DISCUSSIONS

Six pseudorandom algorithms are taken into account – those are RC4, PM, BBS-1S, BBS-3S, BBS-1L and BBS-4L. The results obtained by the three test modules are discussed respectively in sections 4.1, 4.2 and 4.3.

4.1 Results of Test Module-1

Six algorithms each having 300 bit sequences are tested by test module-1. For a particular algorithm the best and worst values of σ calculated by eq. (1) are shown in Table 1 along with the Proportion of Passing obtained from the probability of occurrence of 8-bit value.

Table 1: Value of σ in test module-1

Data set	Best value	Worst value	Proportion of Passing
RC4	0.03391956	0.04429594	100.00
PM	0.03399225	0.04461352	100.00
BBS-1S	1.273882	15.96872	0.00
BBS-3S	0.7244467	9.183318	0.00
BBS-1L	0.03419726	0.04453781	100.00
BBS-4L	0.03499842	0.04455483	100.00

Table 1 indicates that values of σ for data set BBS-1S and BBS-3S are very high. The other data sets are satisfactory.

4.2 Results of Test Module-2

The BBS-1S and BBS-3S are considered here with a view to see their results with the test module-2. The passing criterion for each test is set at 90% while all tests together it is set at 80%. The test-wise values of percentage of passing are presented in Table 2 for the six algorithms considered for study. It is seen that BBS-1S and BBS-3S exhibit poor performance. The passing criteria for other tests are seen to be satisfactory.

Table 2: Percentage of passing in test module-2

Data set	Frequency test	Serial test	Poker test	Runs test	All (4) tests
RC4	94.333	93.667	94.667	92.677	81.333
PM	96.000	95.667	97.333	95.000	88.000
BBS-1S	22.333	4.333	0	0	0
BBS-3S	50.333	35.000	0	0	0
BBS-1L	94.667	94.667	95.333	91.333	80.667
BBS-4L	96.667	95.667	96.333	91.667	85.333

4.3 Results of Test Module-3

In test module-3 four algorithms, namely, RC4, PM, BBS-1L and BBS-4L each having 300 bit sequences are tested. For all the four algorithms, the P-value data of all the 15 tests are presented in Tables 3(a) through 3(d). In Tables 3

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

the P-value data for a particular test are shown being divided in 11 groups between 0 and 1. For each test, OPOP of all the 300 sequences is estimated and presented in Table 4 and if OPOP $\geq T_{value}$, a particular test is considered as being passed. The distribution pattern of P-values is estimated through POP, following the mathematical equations given in the NIST document [11] and the result is presented in Table 5. The P-values are uniformly distributed if POP ≥ 0.0001. The OPOP and POP are the two checking parameters measuring the degree of randomness of an algorithm. The T_{value} differs from test to test since it depends on the number of P-values generated by a test program. The each of the test 11 and 13 has two P-values, the test 14 has eight, test 15 has eighteen and each of the rest eleven tests has one P-value.

If P-value < 0.01, then it will be considered as unsuccessful. Tables 3(a) through 3(d) show the distribution of P-values within specific 11 sub-ranges. The Proportion of Passing for a particular test is the ratio of sum of the last ten columns to the total sum of eleven columns. It is compared with the T_{value} mentioned in eq.(2) to see whether the data set is statistically random or not.

The POP, mentioned in eq.(4), is also derived from Table 3 for any test. Based on this POP one can conclude about the distribution pattern of a data set. The POP ≥ 0.0001 indicates that the distribution is uniform. Table 4 depicts the OPOP, where in columns 3rd through 6th the Y/N indicates the successful/unsuccessful in Proportion of Passing. The POP is depicted in Table 5, where in columns 2nd through 5th the Y/N indicates the uniformity/non-uniformity of distribution of P-values. From various observations on test results, it is also understood that P-values are more uniformly distributed if POP is larger.

Table 3(a): Frequency distribution of P-values of RC4

Test No.	0.00-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	6	24	29	33	38	26	36	32	24	25	27
2	1	27	32	31	33	32	31	26	26	25	36
3	6	24	29	31	32	25	17	35	32	38	31
4	1	36	39	30	23	29	28	30	21	34	29
5	3	27	26	33	40	32	27	25	27	31	29
6	4	19	43	29	26	28	27	35	37	25	27
7	4	25	28	29	28	37	27	34	32	32	24
8	4	29	30	28	28	24	37	38	29	25	28
9	2	23	30	24	25	43	31	29	28	33	32
10	4	29	26	34	29	39	21	28	24	32	34
11	5	60	70	62	56	49	51	65	63	62	57
12	3	26	38	31	30	27	26	31	31	29	28
13	7	54	61	61	44	57	71	55	70	58	62
14	29	216	236	248	246	263	252	233	254	207	216
15	72	474	543	573	558	561	530	568	503	522	496

Table 3(b): Frequency distribution of P-values of PM

Test No.	0.00-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	2	20	32	28	25	32	37	27	31	46	20
2	1	26	29	32	30	31	30	29	28	27	37
3	1	26	41	29	33	31	30	32	23	25	29
4	5	23	22	35	25	33	27	31	42	27	30
5	3	28	30	29	37	28	27	28	28	30	32
6	6	26	31	27	29	38	24	36	34	27	22
7	3	26	32	21	30	23	28	30	39	30	38
8	6	32	29	29	22	32	25	27	33	33	32
9	6	16	42	30	26	33	39	33	26	30	19
10	2	31	32	22	40	31	30	21	25	34	32
11	7	61	49	63	80	58	54	46	70	51	61
12	3	30	21	30	47	29	23	28	35	24	30
13	7	51	64	71	63	61	54	69	68	52	40
14	42	260	264	220	231	213	252	259	216	236	207
15	93	487	558	557	568	501	544	545	502	544	501

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

Table 3(c): Frequency distribution of P-values of BBS-1L

Test No.	0.00-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	4	21	19	27	24	33	27	37	36	32	40
2	1	25	23	29	27	34	35	30	28	36	32
3	2	20	27	30	31	31	44	28	27	28	32
4	8	16	39	25	28	26	33	40	26	28	31
5	2	25	37	30	25	31	21	33	23	36	37
6	1	23	38	28	27	35	18	36	37	32	25
7	2	29	34	31	28	27	33	26	27	29	34
8	4	28	24	28	39	22	36	24	34	30	31
9	6	32	34	27	35	27	23	23	26	35	32
10	1	33	29	24	27	29	33	28	32	36	28
11	4	25	53	64	58	66	47	73	69	60	81
12	2	12	24	32	31	29	21	36	38	37	38
13	12	54	77	56	51	65	73	59	53	45	55
14	34	245	241	242	227	215	262	251	220	236	227
15	65	427	492	485	555	573	569	546	568	566	554

Table 3(d): Frequency distribution of P-values of BBS-4L

Test No.	0.00-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	5	17	33	33	30	33	41	24	29	21	34
2	1	17	30	31	33	25	26	28	23	46	40
3	3	33	35	30	27	30	35	21	37	22	27
4	3	21	34	30	38	26	20	32	30	35	31
5	1	31	31	25	28	24	43	34	33	24	26
6	4	33	36	28	36	34	24	29	32	18	26
7	4	23	35	32	34	25	23	34	40	27	23
8	1	28	27	32	28	32	35	23	28	37	29
9	2	27	30	38	23	36	25	26	28	26	39
10	2	27	28	30	35	42	32	25	28	27	24
11	9	51	38	69	54	61	66	60	53	69	70
12	4	24	20	35	28	34	40	25	27	30	33
13	4	38	52	67	51	66	68	65	72	55	62
14	34	233	220	234	228	269	240	238	219	252	233
15	64	479	517	525	508	507	458	545	596	568	633

5. CONCLUSION

The BBS-1S and BBS-3S have been discarded by test module-1 as well as by test module-2. From the results shown in Table 4, one can say that PM has two "N" values for Proportion of Passing criteria, for tests 14 and 15 but very close to the success result, and in Table 5, it has uniform distribution of P-values for all the fifteen tests. In Table 5, for BBS-4L the distribution of P-values in test 15 is not uniform, although in Table 4, it is a success following Proportion of Passing criteria. Considering the overall result, BBS-4L can also be treated as practically random. Both RC4 and BBS-1L have very good result. From the observations, we can conclude that BBS would give better pseudorandom bit sequences if large modulus is used instead of a smaller one.

In test module-1, byte level frequency test is executed, so one can avoid the bit level frequency tests in test module-2 and in test module-3 to save the execution time. The serial test in module-3 can be executed for variable length of bit, while the same test in module-2 is considers only 2-bit values. However, as test module-2 consist runs test and serial test; one can also avoid the respective tests 3 and 11 in module-3 to have efficient result.

6. ACKNOWLDGEMENTS

We express our gratitude towards UGC, New Delhi and TCS, Kolkata for providing financial support respectively to the first author and the second author. We are also indeed thankful to the Department of Radio Physics and Electronics, University of Calcutta for providing necessary infrastructural facilities to undertake research activities. The cooperation and supports extended by Dr. Amlan Chakrabarti, the Head, A. K. Choudhury School of Information Technology, University of Calcutta is greatly appreciated.

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

Table 4: Results of the Observed Proportion of Passing (OPOP) under test module-3

Test No.	T_{value}	Observed Proportion of Passing (OPOP)			
		RC4	PM	BBS-1L	BBS-4L
1	0.97277	0.98000 Y	0.99333 Y	0.98667 Y	0.98333 Y
2	0.97277	0.99667 Y	0.99667 Y	0.99667 Y	0.99667 Y
3	0.97277	0.98000 Y	0.99667 Y	0.99333 Y	0.99000 Y
4	0.97277	0.99667 Y	0.98333 Y	0.97333 Y	0.99000 Y
5	0.97277	0.99000 Y	0.99000 Y	0.99333 Y	0.99667 Y
6	0.97277	0.98667 Y	0.98000 Y	0.99667 Y	0.98667 Y
7	0.97277	0.98667 Y	0.99000 Y	0.99333 Y	0.98667 Y
8	0.97277	0.98667 Y	0.98000 Y	0.98667 Y	0.99667 Y
9	0.97277	0.99333 Y	0.98000 Y	0.98000 Y	0.99333 Y
10	0.97277	0.98667 Y	0.99333 Y	0.99667 Y	0.99333 Y
11	0.97781	0.99167 Y	0.98833 Y	0.99333 Y	0.98500 Y
12	0.97277	0.99000 Y	0.99000 Y	0.99333 Y	0.98667 Y
13	0.97781	0.98833 Y	0.98833 Y	0.98000 Y	0.99333 Y
14	0.98391	0.98797 Y	0.98250 N	0.98583 Y	0.98583 Y
15	0.98594	0.98667 Y	0.98278 N	0.98796 Y	0.98815 Y

Table 5: Results of the P-value of P-values (POP) under test module-3

Test No.	P-value of P-values (POP)			
	RC4	PM	BBS-1L	BBS-4L
1	6.71779e-1 Y	4.56746e-2 Y	1.50907e-1 Y	2.40914e-1 Y
2	9.19967e-1 Y	9.78072e-1 Y	8.04337e-1 Y	1.71071e-2 Y
3	3.66918e-1 Y	6.02458e-1 Y	3.72502e-1 Y	3.55909e-1 Y
4	4.01199e-1 Y	3.78138e-1 Y	3.72502e-1 Y	4.62245e-1 Y
5	7.59756e-1 Y	9.79974e-1 Y	3.29332e-1 Y	3.09056e-1 Y
6	2.20931e-1 Y	5.34146e-1 Y	1.53763e-1 Y	2.49284e-1 Y
7	8.93001e-1 Y	3.95358e-1 Y	9.73936e-1 Y	3.45115e-1 Y
8	6.85579e-1 Y	7.06149e-1 Y	4.13032e-1 Y	8.34308e-1 Y
9	4.25059e-1 Y	8.21774e-2 Y	4.49672e-1 Y	3.72502e-1 Y
10	4.55937e-1 Y	3.39799e-1 Y	9.11413e-1 Y	4.94392e-1 Y
11	6.82134e-1 Y	5.08454e-2 Y	1.91356e-4 Y	1.06666e-1 Y
12	9.52778e-1 Y	5.77531e-2 Y	1.96313e-2 Y	3.72502e-1 Y
13	4.65415e-1 Y	1.50907e-1 Y	8.38668e-2 Y	1.37282e-1 Y
14	2.52479e-1 Y	1.58847e-4 Y	1.01765e-1 Y	2.35792e-1 Y
15	2.20502e-1 Y	1.23649e-1 Y	1.15477e-2 Y	4.79661e-6 N

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

REFERENCES

[1] M. G. Kendall, B. B. Smith, Randomness and Random Sampling Numbers. *J. Royal Statistical Soc., 101*, 1938, 147-66.

[2] R. K. Pathria, A Statistical Analysis of the first 2,500 Decimal places of e and 1/e, 1961. http://www.dli.gov.in/rawdataupload/upload/insa/INSA_1/20005aca_270.pdf

[3] R. K. Pathria, A Statistical Study of Randomness Among the First 10,000 Digits of π, 1961. http://www.ams.org/journals/mcom/1962-16-078/S0025-5718-1962-0144443-7/

[4] R. K. Pathria, A Statistical Study of Randomness Among the First60,000Digitsof e, 1963. http://www.dli.gov.in/rawdataupload/upload/insa/INSA_1/20005ab8_663.pdf

[5] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography* (CRC Press, 1996) 169-190.

[6] FIPS Pub 140-1, Security Requirements for Cryptographic Modules, *Archived Publication*, 1994. http://csrc.nist.gov/publications/fips/fips140-1/fips1401.pdf

[7] A. Rukhin, J. Soto, et al, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, *NIST Special Publication 800-22*, 2001. http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22b.pdf

[8] Song-Ju Kim, Ken Umeno, Akio Hasegawa, Corrections of the NIST Statistical Test Suite for Randomness. http://eprint.iacr.org/2004/018.pdf

[9] A. Rukhin, J. Soto, et al, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, *NIST Special Publication 800-22*, Revision 1, 2008. http://csrc.nist.gov/publications/PubsSPArch.html

[10] J. Soto, Statistical Testing of Random Number Generators, *National Institute of Standards & Technology*. http://csrc.nist.gov/groups/ST/toolkit/rng/documents/nissc-paper.pdf

[11] A. Rukhin, J. Soto, et al, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, *NIST Special Publication 800-22*, Revision 1a, 2010. http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf

[12] D. E. Knuth, *The Art of Computer Programming Seminumerical Algorithms Volume 2* (3rd edn, Addison Wesley, Reading, Massachusetts, 1998).

[13] Pierre L'Ecuyer, Richard Simard, A C-Library for Empirical Testing of Random Number Generators, *ACM Transactions on Mathematical Software, 33*(4), 2007.

[14] A. G. Oprina, A. Popescu, E. Simion, G. Simoin, Walsh-Hadamard Randomness Test and New Methods of Test Results Integration, *Bulletin of the Transilvania University of Brasov, 2*(51), Series III: Mathematics, Informatics, Physics, 2009, 93-106.

[15] E. Barker, J. Kelsey, Recommendation for Random Bit Generator (RBG) Constructions, *NIST Special Publication 800-90C (Draft)*, Computer Security Division, Information Technology Laboratory, August 2012. http://csrc.nist.gov/publications/drafts/800-90/draft-sp800-90c.pdf

[16] D. Hotoleanu, O. Cret, A. Suciu, T. Gyorfi, L. Văcariu, Real-time Testing of True Random Number Generators through Dynamic Reconfiguration, *Proc. 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, 2010 Doi 10.1109/DSD.2010.56 (© 2010 IEEE)

[17] JKM. S. Zaman, R. Ghosh, Review on fifteen Statistical Tests proposed by NIST, *J. Theoretical Physics and Cryptography, 1*, 2012, 18-31.

[18] W. Stallings, *Cryptography and Network Security Principles and Practices* (4th edn, Delhi, Pearson Education, 2008).

[19] B.A. Forouzan, D. Mukhopadhyay, *Cryptography and Network Security* (2nd edn, New Delhi, TMH, 2011).

[20] G. Paul, S. Maitra, *RC4 Stream Cipher and its Variants* (Boca Raton, Chapman & Hall, CRC, 2012).

[21] S. Maitra, G. Paul, Analysis of RC4 and Proposal of Additional Layers for Better Security Margin, *Proc. Indocrypt*, IIT Kharagpur, LNCS 5365, 2008, 27-39.

[22] S. Paul, B. Preneel, A New Weakness in the RC4 Keystream. Generator and an Approach to Improve the Security of the Cipher, *Proc. Fast Software Encryption*, Berlin, LNCS 3017, 2004, 245-259.

[23] S. Fluhrer, I. Mantin, A. Shamir, Weakness in the Key Scheduling Algorithm of RC4, *Proc. Int. Workshop on Selected Areas in Cryptography*, Toronto, LNCS 2259, 2001, 1-24.

[24] A. Roos, A class of weak keys in the RC4 stream cipher, 1995. http://groups.googl.com/group/sci.crypt.research/msg/078aa9249d76eacc?dmode=source

[25] S.K. Park, K.W. Miller, Random Number Generators: Good ones are hard to find, *Communications of the ACM, 31*(10), 1988, 1192 – 1201.

[26] L. Blum, M. Blum, M. Shub, A Simple Unpredictable Pseudo-Random Number Generator, *SIAM Journal on Computing, 15*(2), 1986, 364-383

[27] D. R. Stinson, *Cryptography Theory and Practice* (3rd edn, Boca Raton, Chapman & Hall, CRC, 2006).

Author's Biography

J K M Sadique Uz Zaman received his MCA and M. Tech. degree respectively from T. M. Bhagalpur University and University of Calcutta and pursuing Ph.D. (Tech) at University of Calcutta as a UGC selected SRF in Engineering and Technology. He has 4 research papers in

Volume: 6 Issue: 1 Pages: 2141-2150 (2014) ISSN : 0975-0290

International Journals and 1 book chapter in Springer LNCS. He has qualified the GATE-2012 in Computer Science and Information Technology and also qualified the UGC-NET December 2012 for Lectureship in Computer Science and Applications.

Sangeet Saha received his M. Tech. degree from University of Calcutta in Computer Science and Engineering and pursuing Ph.D. (Tech) at the same university. He is enjoying scholarship provided by the TCS (Tata Consultancy Services), Kolkata.

Ranjan Ghosh received his M. Tech. and Ph.D. (Tech) from University of Calcutta. He is an Associate Professor at Department of Radio Physics and Electronics, University of Calcutta. Topic of his present research area is Cryptography and Network Security.