Three Graphical Tests for Quality Analysis of Pseudorandom Numbers

Aye Myat Nyo
University of Computer Studies, Mandalay
ayemyatnyo81@gmail.com

Abstract

Many cryptographic protocols require random or pseudorandom inputs at various points (e.g., for generating digital signatures, or in some authentication protocols). Pseudorandom numbers are critical part of modern computing and they are produced from PRNGs (pseudorandom number generators). The strength of many cryptosystems relies on the quality of pseudorandom number generated by PRNG. As pseudorandom number plays an important role in cryptography, many statistical tests for analyzing the quality of pseudorandom sequence generated by PRNGs have been developed. To analysis the quality of PRNGs' output key-stream, three graphical tests: (i) Groups Comparison Test (ii) Longest Runs of All Test and (iii) Forward Appearance Test are proposed in this paper. And then this paper intended for studying important characteristics of PRNGs) as well as randomness testing of random numbers.

1. Introduction

Cryptography is a physical process that scrambles information by rearrangement and substitution of content. Cryptography enables to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient. The benefit of securing information using cryptographic processes becomes a major goal for many organizations.

A pseudorandom number generator is a cryptographic algorithm used to generate sequence of numbers that must appear randomly. A pseudorandom number generator uses one or more inputs and generates multiple pseudorandom sequences that are approximately independent of each other.

Pseudorandom numbers must be uncorrelated, reproduceable, and portable. Moreover, they can be easily changed by adjusting an initial seed value and then they have a large period of repetition and they must pass all empirical tests for randomness, and they can be generated rapidly using limited computer memory. Many cryptosystems were vulnerable to particular attack due to the weakness of applied PRNGs.

In all these applications, security depends greatly on the randomness of the source. Because security algorithms and protocols rely on the unpredictability of the random bits they use. The NIST statistical test suite package was developed for testing random and pseudorandom number generators [3]. Moreover, there are too many test suits that are intended to test the randomness of the PRNGs [1].

In order to determine whether the pseudorandom sequence is secure or not, we presents Group Comparison Test, Longest Run of All Test and Forward Appearance Test.

graphical tests. These three graphical tests may be useful as an initial step in determining whether or not a random sequence is suitable for cryptographic application.

The purpose of the Group Comparison Test is to analysis the sequence is presence of uniform distribution property to be thought as appropriate randomness. Longest Runs of All Test is to determine this sequence have unpredictability property to be evaluated as a truly random sequence. The task of the Forward Appearance Test is to analysis unpredictability and uniform distribution property.

2. Related Work

As pseudorandom number plays an important role in cryptography, many statistical tests for analyzing the quality of pseudorandom sequence have been invented. The NIST statistical test suite package was developed for testing random and pseudorandom number generators [1]. Moreover, the frequency, serial, gap, poker, coupon collector's, permutation, run, maximum-of-t, collision, birthday spacing, and serial correlation tests were proposed by Donald Knuth [6].

The DIEHARD suite of statistical tests developed by George Marsaglia also consists of fifteen tests [7]. Moreover, the rough set theory can be used for dimension reduction and in [5], Rough Set based Decision Tree (RDT) is constructed based on the predominant attributes.

The Crypt-X suite of statistical tests included the frequency, binary derivative, change point, runs, sequence complexity and linear complexity [2]. All of these test suits were developed with the intention for testing the randomness.

By means of the rough set approach, the input (seeds) can be guessed from databases using the known output sequence. So, the quality analysis of the (PRNGs) and a simple rule based prediction system is presented in paper [8].

3. Pseudorandom Number Generator (PRNG)

A pseudorandom number generator (PRNG) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state. In contrast a true random number generator (TRNG) provides truly random data gathered from undeterministic phenomenon occurring in nature.

Inputs to pseudorandom number generator are called initial seeds. In contexts in which unpredictability is needed, the seed itself must be random and unpredictable [1]. Pseudorandom number generators are widely used in such applications as computer modeling (e.g., Markov chains), statistics, experimental design, cryptographic approach, etc.

Random number generators represent basic cryptographic primitives [3]. They are widely used for example as confidential key generators for symmetric key, stream cipher, public-key cryptosystems (e.g. RSA-moduli) and as password sources. In some algorithms (e.g. DSA) or protocols (e.g. zero-knowledge proof), random numbers are intrinsic to the computation [4].

3.1. Properties of PRNG

A PRNG is its own kind of cryptographic primitive and the better understanding of these primitives will make it easier to design and use PRNGs securely [3]. A PRNG is a single point of failure for many real-world cryptosystems. Many systems use badly-designed PRNGs, or use them in ways that make various attacks easier [2]. So random and pseudorandom numbers generated

for cryptographic applications should be unpredictable.

In the case of PRNGs, if the seed is unknown, the next output number in the sequence should be unpredictable in spite of any knowledge of previous random numbers in the sequence. This property is known as forward unpredictability [1]. It should also not be feasible to determine the seed from knowledge of any generated values (i.e., backward unpredictability is also required) [10].

To ensure forward unpredictability, care must be exercised in obtaining initial seeds. The values produced by a PRNG are completely predictable if the seed and generation algorithm are known. In addition, the seed itself must be unpredictable.

3.2. Random Number Testing

A random bit sequence could be interpreted as the result of the flips of an unbiased "fair" coin with sides that are labeled "0" and "1," with each flip having a probability of exactly ½ of producing a "0" or "1". The flips are independent of each other: the result of any previous coin flip does not affect future coin flips. All elements of the sequence are generated independently of each other, and the value of the next element in the sequence cannot be predicted.

Random and pseudorandom numbers generated for cryptographic applications should be unpredictable. Various statistical tests can be applied to a sequence to attempt to compare and evaluate the sequence to a truly random sequence [10]. Randomness is a probabilistic property. The properties of a random sequence can be characterized and described in terms of probability. Because there are so many tests for judging whether a sequence is random or not .No specific finite set of tests is deemed "complete".

Poker Test starts by dividing the 20,000 bit stream into 5,000 contiguous 4-bit segments. One then counts and stores the number of occurrences of each of the 16 possible 4-bit values. Denoting $f(i)$ as the number of each 4-bit value i where $0 \le i \le 15$, one then evaluate the following:

$$\mathbf{X} = \frac{16}{5000}\left(\sum_{i=1}^{15}[f(t)]\right)\text{-}5000 \qquad (1)$$

The test is passed if $2.16 < \mathbf{X} < 46.17$.The NIST tests and DIEHARD tests, like many statistical tests, are based on hypothesis testing. A hypothesis test is a procedure for determining if an assertion about a characteristic of a population is reasonable. In the present case, the test involves determining whether or not a specific sequence of zeroes and ones is random (it is called null hypothesis H0). For each test, a relevant randomness statistic be chosen and used to determine the acceptance or rejection of the null hypothesis. Under an assumption of randomness, such a statistic has a distribution of possible values.

A theoretical reference distribution of this statistic under the null hypothesis is determined by mathematical methods and corresponding probability value (P-value) is computed, which summarizes the strength of the evidence against the null hypothesis. For each test, the P-value is the probability that a perfect random number generator would have produced a sequence less random than the sequence that was tested, given the kind of non-randomness assessed by the test. If a P-value for a test is determined to be equal to 1, then the sequence appears to have perfect randomness.

A P-value equal to zero indicates that the sequence appears to be completely non-random. A significance level (α) be chosen for the tests and if P-value ≥ α, then the null hypothesis is accepted i.e., the sequence appears to be random. If P-value < α, then the null hypothesis is rejected; i.e., the sequence appears to be non-random. Typically, the significance level (α) is chosen in the interval [0.001, 0.01]. The $\alpha = 0.01$ indicates that one would expect 1 sequence out of 100 sequences to be rejected. A P-value ≥ 0.01 would mean that the sequence would be considered to be random with a confidence of 99%.

4. Proposed Tests

The contributions of this paper are three graphical tests. These are Groups Comparison Test, Longest Runs of All Test and Forward Appearance Test. The basic task of the Groups Comparison Test is to determine a ratio of amount of occurrences of two groups of elements in a tested random byte sequence. The purpose of the Longest Runs of All Test is to determine how much close the occurrence on an output of the pseudorandom number generator of these elements is predicted. The basic task of the Forward Appearance Test is to determine average distance between repeated occurrences of elements of random sequence.

4.1. Group Comparison Test

The purpose of the test is to determine how these elements for all blocks of a tested sequence are close. To be true random sequence, occurrences of values of these two groups should not strongly differ from each other. Otherwise, at presence of obvious differences between two groups, the tested sequence can be concluded as not true random.

A sequence of bytes generated by the RNG or PRNG exists such as $\varepsilon = \varepsilon_1, \varepsilon_2 \ldots \varepsilon_n$. The first group is the elements between interval [0-127] and the second group is the elements between intervals [128-255].

The test can be summarized as follow:

n - The length of input random sequence in bytes

m - Number of testing blocks

- Calculate the length of each block $k = \left[\frac{n}{m}\right]$ (2)
- Calculate the frequency of occurrences of each element of two groups in a tested sequence

  Group 1: 0, 1… 127

  Group 2: 128, 129… 255

Block 1: $b_{11} = \sum_{i=1}^{k} a_i \varepsilon_i$, (3)

$$b_{12} = \sum_{i=1}^{k} b_i \varepsilon_i, \quad (4)$$

Block 2: $b_{21} = \sum_{i=k+1}^{2k} a_i \varepsilon_i$, (5)

$$b_{22} = \sum_{i=k+1}^{2k} b_i \varepsilon_i, \quad (6)$$

…

Block m: $b_{m1} = \sum_{i=(m-1)k+1}^{mk} a_i \varepsilon_i$, (7)

$$b_{m2} = \sum_{i=(m-1)k+1}^{mk} b_i \varepsilon_i, \quad (8)$$

Where

a_i=1, if $\varepsilon_i \in$[0,127] or ai=0, if $\varepsilon_i \in$[0,127]

b_i=1, if $\varepsilon i \in$[128,255] or b_i=0, if $\varepsilon_i \in$[128,255]

With the purpose of simplicity, in the following example we assume that all possible elements (values) set on an interval [0- 9], not on an interval [0-255].

Group 1: 0...4 and Group 2: 5...9.

Tested input sequence $\mathcal{E}$ = 4, 8, 1, 7, 0, 3, 5, 1, 5, 6, 1, 2, 9, 9, 6, 4, 5, 8, 0, 7, 3, 6, 5, 9, 2, 2, 7, 8, 0, 1, 2, 0, 4, 5, 2, 1, 0, 3, 6, 4, 5, 7, 9, 6, 8, 7, 9, 5, 8, 0, 8, 9, 7

The length of the random tested byte string

n = 53

Number of testing blocks **m** = 5

- Calculate the length of each block k = 10
- Calculate the frequency of occurrences of elements (values) for all groups 1 – 5

  b_{11}=5, b_{21}=4, b_{31}=5, b_{41}=8, b_{51}=1

  b_{12}=5, b_{22}=6, b_{32}=5, b_{42}=2, b_{52}=9

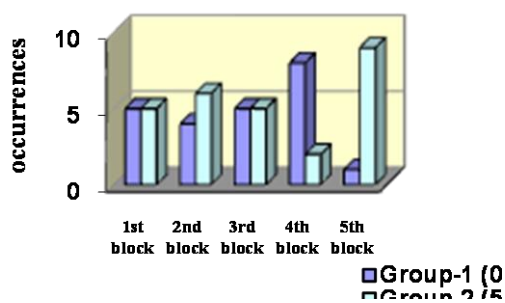

Figure 1. Sample result of Group Comparison Test

It can be clearly seen that in the fourth block, the number of occurrences of elements of group 1 are significantly greater than the number of occurrences of elements of group 2. And then in the fifth block, the occurrences of group 2 are sharply greater than the occurrences of group 1. Therefore, we can conclude that this sequence absents uniform distribution property to be thought as appropriate randomness.

Let us consider next example with another input sequence $\mathcal{E}$ = 1, 9, 3, 6, 0, 5, 7, 8, 2, 4, 7, 5, 0, 9, 2, 8, 1, 3, 4, 6, 9, 1, 7, 6, 5, 0, 2, 3, 1, 8, 3, 6 ,2, 7, 9, 8, 5, 4, 5, 7, 2, 5,8, 1

The length of the random tested byte string

$$n = 44$$

Number of testing blocks m = 4

- Calculate the length of each block k = 10
- Calculate the frequency of occurrences of elements (values) for all groups 1 – 4

  b_{11}=5, b_{21}=5, b_{31}=5, b_{41}=4

  b_{12}=5, b_{22}=5, b_{32}=5, b_{42}=6

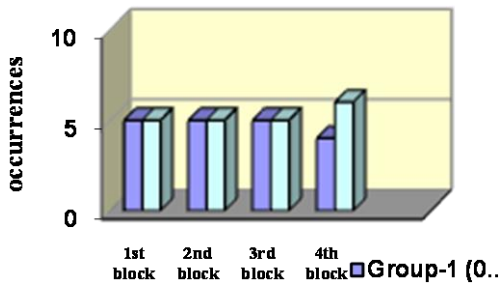

Figure 2. Sample result of Group Comparison Test

It is obviously seen that in first three blocks, the number of occurrences of elements of group 1 are equal to the number of occurrences of elements of group 2. In the fourth block, the occurrences of group 2 are a little more than the occurrences of group 1. This sequence presents uniform distribution property. Therefore, we can conclude this sequence has appropriate randomness. And then we assign threshold value β.

β=A measure of how the observed sequence of numbers for an assumption of randomness

$$\beta = \frac{\sum_{i=1}^{m} \mu}{m} \qquad (9)$$

μ=the difference between the number of occurrences of elements of group 1 and the number of occurrences of elements of group 2

If the computed β is less than or equal to assigned threshold value β, then conclude that the sequence has uniform distribution to be considered as appropriate randomness. Otherwise, we conclude that the sequence doesn't have uniform distribution property.

4.2. Longest Run of All Test

The purpose of the test is to determine the longest series for each element of input sequence and to determine whether these series of each element are too long or not. At presence of too long series, the tested sequence can be concluded as not true random. Sequence of bytes generated by the RNG or PRNG exists such as ε =ε_1, ε_2… ε_n. The test can be summarized as follow:

n - The length of input random sequence in bytes

- Find maximal length of a continuous series for each possible value in input sequence.
- Let maximal lengths of series are

$$\text{max\_run\_len}_i.i = \overline{0,\ 255} \qquad (10)$$

Tested input sequence $\mathcal{E}$ = 4, 1, 1, 7, 0, 0, 5, **1, 1**, 6, 1, 2, 9, 9, 6, 4, **5, 5**, 0, 7, 3, 6, 5, **9, 9, 9, 9, 9, 9**, 1, 2, 0, 4, **3, 3, 3, 3, 3, 3, 3**, 5, 2, 1, **0, 0, 0**, 4, **2, 2**, 3, 7, **4, 4, 4**, **6, 6**, 8, 6, 5, **8, 8, 8**, 9

The length of input random sequence in bytes:

$$\mathbf{n} = 63$$

- Find maximal length of a continuous series for each possible value in input sequence

max_run_len$_0$= 2,
max_run_len$_1$= 1,
max_run_len$_2$= 1,
max_run_len$_3$= 6,
max_run_len$_4$= 2,
max_run_len$_5$= 1,
max_run_len$_6$= 1,

max_run_len$_7$= 0,
max_run_len$_8$= 2,
max_run_len$_9$= 5

- Construct histogram depending on results

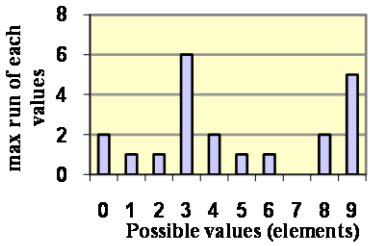

Figure 3. Sample result of Longest Runs of All Test

From this figure we can know that the series for an element (value) 3 is the longest. Therefore, we can prove that sequence can be predicted. In other words, this sequence is lack of unpredictability property to be evaluated as a truly random sequence.

Let us consider with next input sequence:

$\mathcal{E}$ = **2, 2,** 0, 6,1,4,1,7, 7, 8,1, 2,7, 3, **1, 1,** 2, 9, **6, 6,** 4, **5, 5,**6, 5, 1, **0, 0,** 2, 0, 4, **3, 3,3,** 1,4,1,0,5, 3**, 7, 7**,**4, 4,** 8,5,3, **9, 9,** 6, 5, **8, 8**, 9,9,0,5,1,7,2,7

The length of input random sequence in bytes:

n = 60

- Find maximal length of a continuous series for each possible value in input sequence

max_run_len$_0$= 1,
max_run_len$_1$= 1,
max_run_len$_2$= 1,
max_run_len$_3$= 2,
max_run_len$_4$= 1,
max_run_len$_5$= 1,
max_run_len$_6$= 1,
max_run_len$_7$= 1,
max_run_len$_8$= 1,
max_run_len$_9$= 1

- Construct histogram depending on calculated result

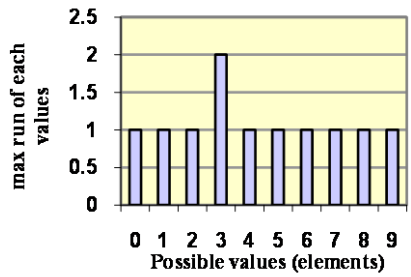

Figure 4. Sample result of Longest Runs of All Test

From previous figure we can see that almost of the series for each element are same. Therefore, we can conclude that this sequence presents of unpredictability property. And then we assign threshold value β.

β =A measure of how the observed sequence of numbers for an assumption of randomness. And then we compute the differences between the numbers of occurrences of elements. If the computed β is less than or equal to assigned threshold value β, then conclude that the sequence presents of unpredictability property to be evaluated as a truly random. Otherwise, conclude that the sequence absents unpredictability property.

4.3. Forward Appearance Test

To be true random sequence, average distance between repeated occurrences of elements should not strongly differ from each other. Otherwise, at presence of obvious differences even for one block, the tested sequence admit not true random. A sequence of bytes generated by the RNG or PRNG exists such as $\varepsilon = \varepsilon_1, \varepsilon_2 \ldots \varepsilon_n$. The test can be summarized as follow:

n - The length of input random sequence in bytes.

- Find the number of recurrences N_i for every **i**-th element in input sequence.

- Find the distances between repeated occurrences for every **i**-th element in input sequence.

$$\text{next\_appear\_len}_{1i}, \text{next\_appear\_len}_{2i} \ldots i = \overline{0,\ 255} \quad (11)$$

- Calculate the mean value for every **i**-th element in input sequence

$$mean_i,\ i = \overline{0,\ 255} \quad (12)$$

$$mean_i = \frac{\sum_{j=1}^{N_i - 1} next\_appear\_len_{ji}}{N_i - 1} \quad (13)$$

$$\text{max\_run\_len}_i.i = \overline{0,\ 255} \quad (14)$$

Tested input sequence $\mathcal{E} = 4, 1, 1, 7, 0, 1, 5, 1, 6, 3, 1, 2, 9, 9, 6, 4, 5, 5, 0, 7, 3, 6, 5, 9, 8, 1, 9, 0, 9, 1, 2, 0, 4, 5, 2, 1, 0, 8, 8, 4, 2, 2, 3, 7, 4, 2, 4, 6, 6, 0, 6, 5, 7, 7, 1$

The length of random sequence in bytes:

n = 80

For every **i**-th element in input sequence, calculate the mean values.

$$\text{mean}_0 = \frac{14 + 9 + 4 + 5 + 13}{6 - 1} = 9$$

$$\text{mean}_1 = \frac{1 + 3 + 2 + 3 + 15 + 4 + 6 + 19}{9 - 1} = 6.625$$

$$\text{mean}_2 = \frac{19 + 4 + 6 + 1 + 4}{6 - 1} = 6.8$$

$\text{mean}_3 = 16.5$, $\text{mean}_4 = 9.2$, $\text{mean}_5 = 9$

$\text{mean}_6 = 8.4$, $\text{mean}_7 = 12.5$, $\text{mean}_8 = 7$,

$\text{mean}_9 = 8.6$

- Construct histogram depending on calculated results

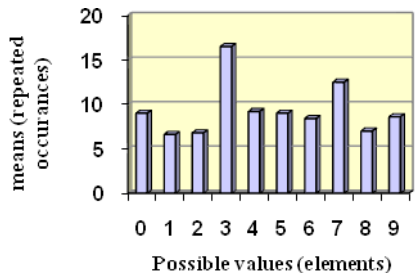

Figure 5. Sample result of Forward Appearance Test

From histogram, it can be clearly seen that average distance between occurrences of element (value) 3 is more than others. Therefore, this sequence absents unpredictability and uniform distribution property to be concluded as appropriate randomness.

Let us prove with next input sequence: 9,7,4,2,7,1,5,0,5,9,2,5,8,3,1,4,8,4,0,3,7,0,6,9,8,3, 6,07,2,5,1,8,4,2,5,0,2,1,7,2,6,3,1,9,3,7,2,5,4,1,3,7 ,2,5,1,53,4,3,6,4,1,9,6,9,2,9,4,5,8,6, 2,8, 8,6

The length of input random sequence in bytes:

n = 77

For every **i**-th element in input sequence, calculate the mean values.

mean_0=9.5,

mean1=9.142

mean2=9.75,

mean_3=8.714,

mean4=9,

mean5=9.12,

mean_6=11.43,

mean7=9,

mean8=10,

mean_9=10.142

- Construct histogram depending on calculated results

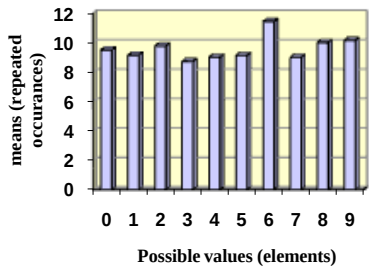

Figure 6. Sample result of Forward Appearance Test

From histogram we can conclude that those average distances between occurrences of element are almost equal. We can prove that this sequence presents unpredictability and uniform distribution property to be evaluated as truly randomness.

5. Conclusions

Developing a good and efficient algorithm for generating pseudorandom numbers on a computer is a very difficult problem. No amount of empirical testing can ever determine how well a given generator will perform for a new application. It is crucial that the implementation and testing of random number generators keeps pace with changes.

In order to determine whether the generator is secure or not, three graphical tests are applied. We determine the generator is suitable for use in cryptographic applications which may need to meet stronger requirements than for other applications. In particular, outputs of generators must be unpredictable in the absence of knowledge of the inputs. These three graphical tests may be useful as an initial step in determining whether or not a generator is suitable for a particular cryptographic application.

References

[1] A Statistical Test Suites for Random and Pseudorandom Number Generators For Cryptographic Applications, NIST Special Publication 800-22 (revision) -2008.

[2] Crypt-X test suit http://www.isrc.qut.edu.au/cryptx/index.html.

[3] Cryptanalytic Attacks on Pseudorandom Number Generators, John Kelsey, Bruce Schneier , David Wagner , Chris Hally.

[4] Chaos–based True Random Number Generator Embedded in A Mixed–Signal Reconfigurable Hardware, Miloˇs Drutarovsk´y — Pavol Galajda.

[5] Decision Tree Indution Using Rough Set Theory, Ramadevi Yellasiri, C.R.Rao, Vivekchan Reddy, Dept. of CSE, JATIT, 2007.

[6] Donald Knuth, The Art of Computer Programming, Seminumerical Algorithms, Volume 2, 3rd edition, Addison Wesley, Reading, Massachusetts, 1998.

[7] Marsaglia G., DIEHARD: a battery of tests for random number generators. http://stat.fsu.edu/~geo/diehard.html.

[8] Quality Analysis of Pseudorandom Number Generator Using Rough Sets, Than Naing Soe, Aye Thida, Thein Than Thwin, Aye Myat Nyo, Wai Wai Zin

[9] Ueli Maurer, “A Universal Statistical Test for Random Bit Generators,” Journal of Cryptology, Vol. 5, No. 2, 1992, pp. 89-105.

[10] William Stallings, “Cryptography and Network Security”, Fourth Edition, ISBN 81-7758-774-9, Copyright 2006 by Pearson Education, Inc.