JOURNAL OF THEORETICAL PHYSICS & CRYPTOGRAPHY

www.IJTPC.org/Publications

A Pseudorandom Number Generator using Chaotic Image Processing

Rama Gayakwad, Pankita Ranbir

Department of Computer Science & Informatics, University of Kota, INDIA

Accepted: 14th September, 2012

ABSTRACT

In this paper, after reviewing the main points of random number and logistic map, we introduced a pseudorandom number generator (PRNG) using image processing that the image has the uniform two color distribution similarly evolving along its length and width i.e. it is symmetric with respect to the y=x,that we use simplest image, that is, checkerboard. It is very simple, convenient and fast for PC platforms. To evaluate the randomness of the binary sequences generated by the PRNG, the two basic tests: monobit test, serial test and the most stringent tests of randomness: the NIST suite tests were performed. As a result, no shortcoming have been observed in the binary sequences generated by the proposed PRNG.

Keywords

chaos, Image processing, PRNG.

1. INTRODUCTION

Cryptography is always very important in data origin authentications, entity authentication, data integrity and confidentiality. The good random number generator is intransitive in cryptography for generation of cryptographic keys, allegorically, secret keys utilized in symmetric cryptosystems [11,5] and large numbers is intransitive in asymmetric cryptosystems [10,16], because of unpredictable, should better be generated randomly. In addition, random number generators in many cryptographic protocols, such as to create challenges, padding bytes, blinding values, nonces are employed [1,18,17]. Random number generators can be classified into three classes which are pseudorandom number generators (PRNGs), true random number generators and hybrid random number generators. PRNGs use deterministic processes to generate a series of outputs from an initial seed state [2,3,14]. True random number generators use of non-deterministic source (i.e., the entropy source), along with some processing function (i.e., the entropy distillation process) to generate the random binary sequence [11]. These sources consist of physical phenomena such as atmospheric noise, thermal noise, radioactive decay and even coin-tossing [20]. For running on a personal computer (PC), all the above mentioned sources for true random number generators require additional devices. Considering the above description, PRNGs are usually easier, cheaper and faster than true random number generators for PC applications. Many PRNGs using chaotic maps have been established. Most of them have very complex structures. As we know, complexity of defining equations of the deterministic processes is not a necessary condition for generation of a random binary sequence,as well as, this complexity makes the use of PRNGs be boring. Therefore, in this paper, we propose the new simple pseudorandom number generator (PRNG) based on image processing that the image has the uniform two color distribution similarly evolving along its length and width i.e. it is symmetric with respect to the y=x. The proposed PRNG is more speed, convenient, simple and secure than general PRNGs. To demonstrate these features, we use simplest chaotic map and image, that is, chaotic logistic map and checkerboard. The random sequences produced by the generator is evaluated using the monobit test, the serial test and the 15 statistical tests recommended by U.S. NIST [11]. Experimental results show that this PRNG possesses good uniformity and randomness properties. This paper is arranged as follows. In Section 2, the properties of the logistic map are discussed. In Section 3, we introduce the Proposed pseudorandom number generator. In Section 4, we discuss the uniformity and randomness of the binary sequences generated by the Proposed pseudorandom number generator and finally, in Section 5, we conclude the paper.

2. THE LOGISTIC MAP

The logistic map is one of the most studied discrete chaotic maps. It is well-known as very sensitive to both system variable and control parameter. In addition, other features such as ergodicity, pseudo-randomness and unpredictable behavior. Therefore, it possesses great potential for various cryptographic applications such as image encryption [4,22], publickey cryptography [13], key agreement protocol [17], block cipher [5], and hash function [15,19,21]. It was first proposed as pseudo random number generator by Von Neumann in 1947 partly because it had a "known algebraic distribution and mentioned later, in 1969, by Knuth [8,9]. The simplest form of the logistic map is given by:

$$x_{n+1} = rx_n(1-x_n) \qquad (1)$$

where $x_n \in (0,1)$ and r are the system variable and control parameter, respectively, and n is the number of iterations. Thus, given a control parameter r and a system value x_0; time series of logistic map $\{x_n\}_{n=0}^{\infty}$ is computed. Here, we refer to x_0 and r as the initial state of the logistic map. In the following we use the chaotic logistic map(Fig 1), for the generation of random binary sequences for cryptographic applications, as follows:

$$x_{n+1} = rx_n(1-x_n), \text{ for } x_n \in (0,1), \text{ and } r \in (3.99996,4]$$

As stated in [12],The choice of r in the equation above guarantees the existence of a chaotic orbit that can be shadowed by only one map as stated in . In addition, the above map is supposed to have good qualities as a PRNG when $r \cong 4$ [6].

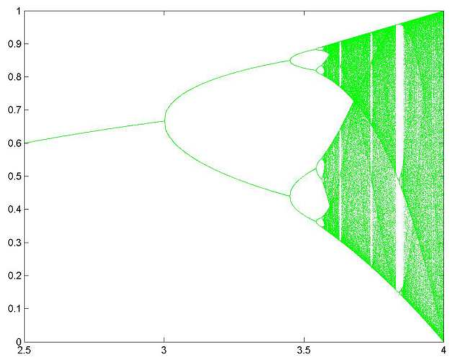

Fig.1. Bifurcation diagram of logistic map.

3. PROPOSED PSEUDORANDOM BIT GENERATOR

To consider a image with the size of 256×256 pixels that the image has the uniform two color distribution imilarly evolving along its length and width i.e. it is symmetric with respect to the y=x,such as checkerboards which are simplest image form this type. Some of checkerboards as shown Fig.2 .

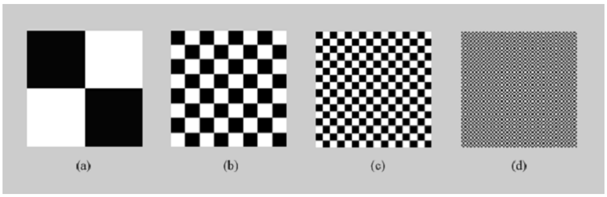

Fig. 2. (a) checkerboard of 2×2 blocks, (b) checkerboard of 8×8 blocks, (c) checkerboard of 16×16 blocks, (d) checkerboard of 64×64 blocks.

Since the image is 256×256 pixels while the output is a randum number containing n bits ($n \geq 10$ there must be a transformation from 2D (2 dimension) signal to 1D signal during the processing. In our experiments, a simple method is used to perform such as transformation, which is composed four steps:

Step 1. For each pixel of the image a coordinate (x,y) is assigned.

Step 2. With using two chaotic maps (in this case, chaotic logistic maps) choose a pixel of image in each iterate.

Step 3. If the pixel is white, the pixel is assigned a value of 1. Otherwise, it is assigned a 0 value.

Step 4. Generate random binary sequence with iterate of step 2 and step 3 (See Fig.3).

One point that should be noted, security issue is the size of the input space. If it is not large enough, the attackers may guess the image with brute-force attack. In our experiments, we use the checkerboard image with 256×256 pixels. Hence, the size of the input space only for checkerboard image is no less than 2^{256} . This size is large enough to defeat brute-force by any super computer today.

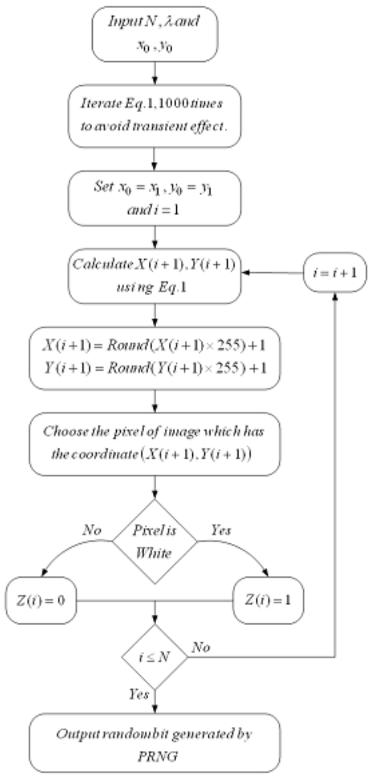
Input N, λ and x_0, y_0
Iterate Eq.1, 1000 times to avoid transient effect.
Set $x_0 = x_1, y_0 = y_1$ and $i = 1$
Calculate $X(i+1), Y(i+1)$ using Eq.1
$X(i+1) = Round(X(i+1)\times255)+1$
$Y(i+1) = Round(Y(i+1)\times255)+1$
Choose the pixel of image which has the coordinate $(X(i+1), Y(i+1))$
No
Pixel is White
Yes
$Z(i) = 0$
$Z(i) = 1$
$i \leq N$
No
Yes
Output random bit generated by PRNG
$i = i+1$

Fig. 3. Block diagram of PRNG

4. ANALYSIS OF RANDOMNESS OF BIT SEQUENCES

In this section, We have survey the randomness and uniformity of the several binary sequences of large size, generated by the PRNG for different sets of control parameter and initial conditions of chaotic maps and images. Here, we show the results for 10000 and 15000 sized binary sequences corresponding to the following parameter values of the four sets: (4,0.2,0.6,a), (3.99999,0.3,0.4,b), (3.99998,0.5, 0.8, c), (3.99997,0.7, 0.6, d), where the first parameter value represents the control parameter value which is same for both logistic maps, and the second and third one as the initial condition for the two logistic maps, and fourth one as the number of checkerboard images, as shown in Fig.2. For convenience, these four sets are designated as $(r_1, x_1, y_1, a), (r_2, x_2, y_2, b)$, $(r_3, x_3, y_3, c), (r_4, x_4, y_4, d)$. We discuss in the following paragraph of this Section the result and conclusions of our study of the different statistical tests to observe the randomness and uniformity of the binary sequences generated by the proposed PRNG.

Monobit Test: The goal of this test is to determine whether the frequency of 0's and 1's in binary sequences generated by the PRNG are approximately same [7]. Let n_0, n_1 denote the number of 0's and 1's in binary sequences respectively. We calculate χ^2 by using the formula [7]:

$$\chi^2 = \frac{(n_0 - n_1)^2}{n}$$

which approximately follow a χ^2 distribution with one degree of freedom. The computed results are shown in Table 1.The calculated values of χ^2 are less as compared to the critical value of χ^2 at $\alpha = 0.05$ (5% level of significance) and 1df (one degree of freedom). It means that these binary sequences pass the monobit test and can be said to be satisfactorily random with respect to this test [7].

Table. 1. Shows the calculated χ^2 values for monobit test for two different large sized binary sequences having N=10000 and 15000 corresponding to four different sets of parameter values.

Size	Parameter	Calculated χ^2 value	Critical χ^2 value at $\alpha = 0.05$
10000	(r_1, x_1, y_1, a)	0.0036	3.8415
10000	(r_2, x_2, y_2, b)	0.0016	3.8415
10000	(r_3, x_3, y_3, c)	0.9216	3.8415
10000	(r_4, x_4, y_4, d)	0.0324	3.8415
15000	(r_1, x_1, y_1, a)	0.3651	3.8415
15000	(r_2, x_2, y_2, b)	0.1563	3.8415
15000	(r_3, x_3, y_3, c)	0.6403	3.8415
15000	(r_4, x_4, y_4, d)	0.0096	3.8415

Serial Test: The goal of this test is to determine whether the number of occurrence of pairs 00, 01, 10 and 11 in the bit streams generated by PRNG is approximately same [7]. Let n_{00}, n_{10}, n_{01} and n_{11} denote the number of occurrence of pairs 00, 01, 10 and 11 respectively in the binary sequences. We calculate χ^2 by using the formula [7]:

$$\chi^2 = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1,$$

and the computed values are found to follow approximately the χ^2 distribution with 2 degrees of freedom. The results are shown in Table 2. The calculated values of χ^2 are less than critical value of χ^2 at $\alpha = 0.05$ (5% level of significance) and 2df (two degrees of freedom). It means that binary sequences pass the serial test and are satisfactorily random with respect to this test.

Shows the calculated χ^2 values for serial test for two different large sized binary sequences having N=10000 and 15000 corresponding to four different sets of parameter values.

Size	Parameter	Calculated χ^2 value	Critical χ^2 value at $\alpha = 0.05$
10000	(r_1, x_1, y_1, a)	1.0161	5.9915
10000	(r_2, x_2, y_2, b)	1.2720	5.9915
10000	(r_3, x_3, y_3, c)	3.1832	5.9915
10000	(r_4, x_4, y_4, d)	1.0932	5.9915
15000	(r_1, x_1, y_1, a)	2.9798	5.9915
15000	(r_2, x_2, y_2, b)	3.1754	5.9915
15000	(r_3, x_3, y_3, c)	4.2791	5.9915
15000	(r_4, x_4, y_4, d)	2.0199	5.9915

In addition to the statistical tests discussed above, the US NIST statistical test suite provides 15 statistical tests to detect deviations of a binary sequence from randomness. A statistical test is formulated to test a null hypothesis which states that the sequence being tested is random. There is also an alternative hypothesis which states that the sequence is not random. For each test, there is an associated reference distribution (typically normal distribution or χ^2 distribution), based on which a P-value is computed from the binary sequence. If the P-value is greater than a predefined threshold α which is also called significance level, then the sequence would be considered to be random with a confidence of $1-\alpha$, and the sequence passes the test successfully. Otherwise, the sequence fails this test. A P-value of zero indicates that the sequence appears to be completely non-random, and the larger the P-value is, the closer a sequence to a perfect random sequence. In our experiment, we set α to its default value 0.01, which means a sequence passed the test is considered as random with 99% confidence. Before presenting the test results of our proposed three approaches, we would first introduce all 16 statistical tests briefly as follows. A more detailed description for those tests could be found in [11].

Frequency test (FT): The goal of this test is to determine whether the number of ones and zeros in a sequence are approximately the same. It is recommended that each sequence to be tested consist of a minimum of 100 bits (i.e., $n \geq 100$).

Frequency Test within a Block (FTB): The goal of this test is to determine whether the frequency of ones is an M-bit block is approximately $\frac{m}{2}$.

Runs test (RT): The goal of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence, where a run is an uninterrupted sequence of identical bits. It is recommended that each sequence to be tested consist of a minimum of 100 bits (i.e., $n \geq 100$).

Test for the longest run of ones in a block (LROBT): The goal of this test is to determine whether the length of the longest run of ones within the tested sequence is as expected for a random sequence. It is recommended that each sequence to be tested consist of a minimum of 6272 bits for M=128.

Binary matrix rank test (BMRT): The goal of this test is to check for linear dependence among fixed length substrings of the original sequence. It is recommended that each sequence to be tested consist of a minimum of 10^5 bits (i.e., $n \geq 10^5$).

Discrete fourier transform test (DFTT): The goal of this test is to detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness. It is recommended that each sequence to be tested consist of a minimum of 1000 bits (i.e., $n \geq 1000$).

Non-overlapping template matching test (NTMT): The goal of this test is to reject sequences that exhibit too many occurrences of a given aperiodic pattern. The length in bits of each template is set to 9. It is recommended that each sequence to be tested consist of a minimum of 2^{20} bits (i.e., $n \geq 2^{20}$).

Overlapping template matching test (OTMT): The goal of this test is to reject sequences that show deviations from the expected number of runs of ones of a given length. The default length of the run of ones is set to 9. It is recommended that each sequence to be tested consist of a minimum of 10^6 bits (i.e., $n \geq 10^6$).

Maurer's universal statistical test (MUST): The goal of the test is to detect whether or not the sequence can be significantly compressed without loss of information. The default length of each block is set to 1280 and the default number of blocks in the initialization sequence is set to 7. It is recommended that each sequence to be tested consist of a minimum of 904,960 bits (i.e., $n \geq 904{,}960$).

Linear complexity test (LCT): The goal of this test is to determine whether or not the sequence is complex enough to be considered random, where the linear complexity is determined by the Berlekamp–Massey algorithm. The default length of each block is set to 500 bits. It is recommended that each sequence to be tested consist of a minimum of 10^6 bits (i.e., $n \geq 10^6$).

Serial test (ST): The goal of this test is to determine whether the number of occurrences of the 2m m-bit overlapping patterns is approximately the same as would be expected for a random sequence. The default length of each block is set to 16 bits. It is recommended that each sequence to be tested consist of a minimum of 10^6 bits (i.e., $n \geq 10^6$).

Approximate entropy test (AET): The goal of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m + 1) against the expected result for a random sequence. The default length of each block is set to 10 bits. It is recommended that each sequence to be tested consist of a minimum of 2^{12} bits (i.e., $n \geq 2^{12}$).

Cumulative sum test (CST): The goal of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences. It is recommended that each sequence to be tested consist of a minimum of 100 bits (i.e., $n \geq 100$).

Random excursions test (RET): The goal of this test is to determine if the number of visits to a state within a random alk exceeds what one would expect for a random sequence. It is recommended that each sequence to be tested consist of a minimum of 10^6 bits (i.e., $n \geq 10^6$).

Random excursions variant test (REVT): The goal of this test is to detect deviations from the expected number of occurrences of various states in the random walk. It is recommended that each sequence to be tested consist of a minimum of 10^6 bits (i.e., $n \geq 10^6$).

The NIST suite tests were performed on four binary sequences, each containing 2^{20} bits. The P-value as well as final results obtained from the NIST suite for four different sets are given in Table 3. The PRNG successfully passes all randomness tests of NIST suite.

Table 3. Shows the P-values obtained from NIST suite for fifteen different tests. The P-values are obtained for four different sets of parameters for each test.

NIST Tests	$(r_1, x_1, y_1, a$	$(r_2, x_2, y_2, b$	$(r_3, x_3, y_3, c$	$(r_4, x_4, y_4, d$
FT	0.701859	0.789024	0.945500	0.964170
FBT	0.523046	0.216087	0.117780	0.393870
RT	0.158536	0.851214	0.276645	0.606118
LROBT	0.602752	0.730143	0.790260	0.155951
RBMRT	0.561422	0.394378	0.591566	0.510643
DFTT	0.231903	0.630986	0.276625	0.023156
ATMT	SUCCESS	SUCCESS	SUCCESS	SUCCESS
PTMT	0.665345	0.093392	0.764690	0.399512
MUST	0.975385	0.673747	0.684394	0.642245
LCT	0. 91286	0. 91698	0. 87837	0. 72932
ST (P1) (P2)	0.105117 0.046667	0.131160 0.067172	0.250763 0.152698	0.319768 0.169570
AET	0.143994	0.035079	0.541637	0.672321
CST (FORWARD) (REVERSE)	0.931691 0.600018	0.939282 0.857153	0.831316 0.792883	0.651691 0.693584
RET	SUCCESS	SUCCESS	SUCCESS	SUCCESS
REVT	SUCCESS	SUCCESS	SUCCESS	SUCCESS

5. CONCLUSIONS

We have proposed a new PRNG which is more speed, convenient, simple and secure than general PRNGs. Many PRNGs using chaotic maps have been established. Most of them have very complex structures. As we know, complexity of defining equations of the deterministic processes is not a necessary condition for generation of a random binary sequence, as well as, this complexity makes the use of PRNGs be boring. Therefore, we have proposed the new simple PRNG based on image processing that the image has the uniform two color distribution similarly evolving along its length and width i.e. it is symmetric with respect to the y=x. To demonstrate these features, we use simplest chaotic map and image, that is, chaotic logistic map and checkerboard. To evaluate the randomness and uniformity, we have employed two different statistical tests i.e. monobit test and serial test on several large sized binary sequences, generated by the PRNG. These binary sequences pass all two tests successfully. Further, the most stringent tests of randomness, the NIST suite tests have also been performed to evaluate the randomness of the binary sequences generated by the PRNG. The PRNG successfully passes all the randomness tests of NIST suite. We suggest the use of the random binary sequences generated by the proposed PRNG to design new secure cryptosystems.

6. REFERENCES

[1] F. Cao, Z. Cao, A secure identity-based proxy multi-signature scheme, Information Sciences, vol. 3, no. 3, pp. 292-302, 2009.

[2] R. M. DSouza, Y. Bar-Yam, M. Kardar, Sensitivity of ballistic deposition to pseudorandom number generators, Phys. Rev. E, vol. 57, no. 5 , pp. 5044-5052, 1998.

[3] J. F. Fernandez, C. Criado, Algorithm for normal random numbers, Phys. Rev. E, vol. 60, no. 3, pp. 3361-3365 , 1999.

[4] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, International Journal of Bifurcation and Chaos, vol. 8, no. 6, pp. 1259-1264, 1998.

[5] G. Jakimoski, L. Kocarev, Block encryption ciphers based on chaotic maps, IEEE Transaction on Circuits System -I, vol. 48, no. 2, pp. 163-169, 2002.

[6] A. Kanso, N. Smaoui, Logistic chaotic maps for binary numbers generations, Chaos, Solitons and Fractals, vol. 40, no. 5, pp. 2557-2568, 2009.

[7] N. K Pareek, V. Patidar, K. K Sud, A Random Bit Generator Using Chaotic Maps, International Journal of Network Security, Vol. 10, no. 1, pp. 32-38, 2010.

[8] C. Peng, S. Prakash, H. J. Herrmann, H. E. Stanley, Randomness versus deterministic chaos: Effect on invasion percolation clusters, Phys. Rev. A, vol.42, no. 8, pp. 4537-4542 , 1990.

[9] S. C. Phatak S. Suresh Rao, Logistic map: A possible random-number generator, Phys. Rev. E, vol. 51, no. 4, pp. 3670-3678 , 1995.

[10] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120-126 , 1978.

[11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel,D. Banks, A. Heckert, J. Dray, S. VoA, statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST special publication, pp. 800-22 , 2010.

[12] N. Smaoui, E. Kostelich Using chaos to shadow the quadratic map for all time, Int J Comput Math, vol. 70, no. 1, pp. 117-129, 1998.

[13] R. Tenny, L.S. Tsimring, Additive mixing modulation for public key encryption based on distributed dynamics, IEEE Transactions on Circuits and Systems-I, vol. 52, no. 3, pp. 672-679, 2005.

[14] I. Vattulainen, T. Ala-Nissila, K. Kankaala, Physical models as tests of randomness, Phys. Rev. E,vol. 52, no. 3, pp. 3205-3214 , 1995.

[15] Y. Wang, X. Liao, K. Wong, One-way hash function construction based on 2D coupled map lattices, Information Sciences, vol. 178, no. 5, pp. 1391-1406, 2008.

[16] B. Wang, Q. Wu, Y. Hu, A knapsack-based probabilistic encryption scheme, Information Sciences, vol. 177, no. 19, pp. 3981-3994 , 2007.

[17] D. Xiao, X. Liao, S. Deng, Using time-stamp to improve the security of a chaotic maps-based key agreement protocol, Information Sciences, vol. 178, no. 6, pp. 1598-1602 , 2008.

[18] D. Xiao, X. Liao, S. Deng, A novel key agreement protocol based on chaotic maps, Information Sciences, vol. 177, no. 4, pp. 1136-1142 , 2007.

[19] X. Yi, Hash function based on chaotic tent maps, IEEE Transactions on Circuits and Systems -II, vol. 52, no. 6, pp. 354-357,2005.

[20] Q. Zhou, X. Liao, K.W. Wong, Y. Hu, D. Xiao, True random number generator based on mouse movement and chaotic hash function, Information Sciences, vol. 179, no. 19, pp. 3442-3450, 2009.

[21] J. Zhang, X. Wang, W. Zhang, Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter, Physics Letters A, vol. 362, no. 5-6, pp. 439-448, 2007.

[22] Q. Zhou, K.W. Wong, X. Liao, T. Xiang, Y. Hu, Parallel image encryption algorithm based on discretized chaotic map,'' Chaos Solitons Fractals, vol. 38, no. 4, p. 1081-1092, 2008.