Ann. Acad. Rom. Sci.
Ser. Math. Appl.
ISSN 2066-6594 Vol. 12, No. 1-2/2020

SELF-SHRINKING CHAOS BASED PSEUDO-RANDOM ALGORITHM*

Borislav Stoyanov†

Dedicated to Dr. Vasile Drăgan on the occasion of his 70th anniversary

Abstract

We propose a novel self-shrinking chaos based pseudo-random number output algorithm. The result of the analysis shows that the presented generator ensures a secure way for sending electronic information with critical applications in data encryption.
MSC: 41A50, 11K45, 11B83

keywords: Chebyshev polynomial, self-shrinking rule, pseudo-random byte generator, statistical suite.

1 Introduction

Random number generators are physical sources (atmospheric noise, electrical noise, radioactive decay, etc.) that return uniformly distributed and completely unpredictable values. Truly random numbers are applicable for a kind of tasks, such as encrypting data, gaming, and experimental design. Generating random numbers is particularly hard. Pseudo-random generators are software alternative algorithms to truly random generators. They

*Accepted for publication on April 21, 2020

†borislav.stoyanov@shu.bg University of Shumen, Bulgaria; Paper written is partially supported of the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science.

are deterministic algorithms that use mathematical formulas to produce pseudo-random sequences of numbers. In the last fifteen years the chaotic functions and self-shrinking schemes have been used actively in the area of pseudo-random generation. The recursive chaotic functions are characterized by "unpredictable" behaviour which can be used in the production of pseudo-random values. To recognize pseudo-random number stream from true ones there are key space analysis [2], correlation analysis [12], speed analysis, and tests with statistical test packages ENT [17] and NIST [13].

Chebyshev maps are extensively included in chaotic cryptographic derivatives. Geisel and Fairen [7] presented a number of chaos based statistical characteristics of Chebyshev polynomials. New pseudo-random generation algorithm designed from the two Chebyshev polynomials, combined with threshold function is proposed in [15]. To extend the ability of anti reconstruction of chaotic function through reverse iteration a fast stochastic middle multi-bits quantification generator based on Chebyshev map is modelled in [6]. The correlation characteristics of the output sequence are analysed. In [14], design of pseudo-random bit generation algorithm combining Chebyshev polynomial and Tinkerbell map, is presented.

The self-shrinking generator [11] is a deterministic random bit generator. It is based on the shrinking principle [4]. The self-shrinking generator uses a single linear feedback shift register (LFSR) [10] to control its gamma. Pairs of output bits are processed. If the first bit is 1 the second bit produces part of the keystream. In [8] pseudo-random algorithm, so-called new self-shrinking generator, has been introduced by Kanso for possible use in spread spectrum communications. One bit is processed. If it is 1, the next generated bit produces part of the output pseudo-random bits.

The aim of the study is to present a novel chaos based pseudo-random number generation algorithm. It can be alternative of algorithms proposed in [1], [6], [8], [12], and [16]. The algorithm is based on simple decimation of Chebyshev polynomial. We evaluate the security of the proposed steps as a binary pseudo-random generator in comparison with some other closely related schemes. The key space of 2^{106} bits of the proposed scheme is enough to defeat brute-force attacks. We show that the correlation coefficients of the output sequences of the novel algorithm are very close to zero. The speed of 28.57 MBit/s. is larger than the other chaos based generators. In addition, the output pseudo-random byte stream is verified with ENT and with NIST test packages.

2 Self-Shrinking Chaos Based Pseudo-Random Algorithm

2.1 Chebyshev Polynomial

The Chebyshev recursive polynomial of the first kind [9] $T_n(x) : R \to R$ of degree n is defined as

$$T_0(x) = 1 \quad (1)$$
$$T_1(x) = x \quad (2)$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad (3)$$

where $n \geq 2$, $x \in [-1, 1]$. Restricted in the interval $[-1, 1]$ the recursive polynomial is a commonly used chaotic function $T_n(x) : [-1, 1] \to [-1, 1]$ for all $n > 1$. First four Chebyshev polynomials of the first kind are plotted in Fig.1.

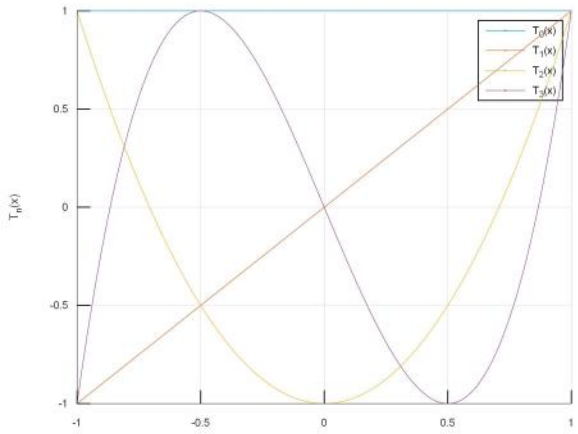

Figure 1: Chebyshev polynomials of the first kind.

2.2 Proposed Algorithm

Here we design a novel self-shrinking Chebyshev polynomial based pseudo-random byte generated algorithm. It includes the following steps:

1. The initial value of $T_0(x)$ is defined. The Chebyshev polynomial function is iterated for L times.

2. The function is iterated one time. The floating-value number $T_i(x)$ is post-processed as integer value $i = abs(mod(integer(T_i(x) \times 10^{16}), 256)$.

3. If $i > 127$ then the Chebyshev function is iterated one time. Then we post-process T_{i+1} as byte value $j = abs(mod(integer(T_{i+1}(x) \times 10^{16}), 256)$. The byte j produces part of the output keystream.

4. Return to Step 2 until a byte output length is reached.

The novel self-shrinking number generator is implemented in C++ language, using the initial key A with values $T_1(x) = -0.16029381194009314$ and $L = 320$.

2.3 Key size evaluation

A requirement condition for an encryption algorithm to be secure is that the key size is high enough so as to disrupt brute-force method.

The secret key size is composed by the two 64 bit secret values T_1 and L. As reported in [18], the computational precision of the 64-bit double-precision number is about 10^{-15}. Then the proposed key size is more than 2^{106} bits. The key space is totally secure against brute-force attacks [2].

2.4 Analysis of correlation coefficient

We tested the byte sequence from novel self-shrinking generator with correlation coefficient test, before and after minimal alteration of the key space.

In the experiments, 3 sequences are generated with initial keys A, B: $T_1(x) = -0.16029381194009315$ and $L = 320$, and C: $L = 319$ and $T_1(x) = -0.16029381194009314$. We have tested sequences with 1,000,000 bytes.

The correlation coefficient r between two adjacent bytes (x_i, y_i) is evaluated by the next formulas [3]:

$$r = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{4}$$

where

$$D(x) = \frac{1}{M}\sum_{i=1}^{M}(x_i - \bar{x})^2, \tag{5}$$

$$D(y) = \frac{1}{M}\sum_{i=1}^{M}(y_i - \bar{y})^2, \tag{6}$$

$$cov(x, y) = \sum_{i=1}^{M} (x_i - \bar{x})(y_i - \bar{y}), \tag{7}$$

M is the total number of couples (x_i, y_i), obtained from the bytes sequences, and $\bar{x}$, $\bar{y}$ are the mean values of x_i and y_i, respectively. The correlation coefficient can range in the interval $[-1.00; +1.00]$.

Table 1 shows the results of adjacent bytes correlation coefficients calculations of the first and the second generated sequences.

Table 1: Correlation coefficients of three pairs of pseudo-random sequences

Sequences	Corr. coef.
(A, B)	0.0018083
(A, C)	0.0023617
(B, C)	-0.00131397

It is clear that the proposed self-shrinking generator not retain any linear dependencies between observed bytes. The inspected correlation coefficients are very close to zero. Overall, the correlation coefficients of the novel generator are similar with results of two other pseudo-random generation schemes [1] and [5].

2.5 Speed analysis

We have measured the output byte generation time by using novel chaos based self-shrinking scheme. Speed test has been done on a 2.4 GHz Intel Core i7-3630QM Dell Inspiron laptop. The results are presented in Table 2. The speed of 28.57 MBit/s. is larger than the other pointed pseudo-random generation algorithms.

Table 2: Speeds of the proposed algorithm and some other algorithms

Algorithm	Speed/MBit.s^{-1}
Proposed	28.57
Reference [12]	1.7
Reference [16]	0.48

2.6 Statistical package testing

The statistical package ENT [17], is used in order to measure the random characteristics of the novel generator. The software suite is based on 6 tests to pseudo-random streams: entropy, χ-square test, arithmetic mean, Monte Carlo value for π, and serial correlation coefficient. We tested output of 2000000000 bits of the novel self-shrinking scheme. The following results are obtained:

- zeros: 999994339 occurrences with fraction 0.499997; ones: 1000005661 with fraction 0.500003.
- entropy is 1.000000 bits per bit.
- optimum compression would reduce the size of this 2000000000 bit file by 0%.
- χ^2 square distribution for 2000000000 samples is 0.060, and randomly would exceed this value 80.01% of the times.
- arithmetic mean value of data bytes is 0.5000 (0.5 is random).
- Monte Carlo value for π is 3.141255602 (error 0.01%).
- serial correlation coefficient is 0.000052 (totally uncorrelated is 0.0).

The byte output values passed successfully ENT suite.

The NIST software [13] includes 15 statistical tests: monobit, block frequency, cumulative sums, runs, longest run of ones, rank, spectral, non-overlapping templates, overlapping templates, Universal statistical, approximate entropy, serial, linear complexity, random excursions, and random excursion variant. 2000×125000 bytes were processed using the novel self-shrinking generator. The output values of the first 13 test are presented in Table 3. The minimum pass rate for each test with the exception of the random excursion variant test is approximately equal to 1966 for a sample size of 2000 byte sequences.

The random excursion test outputs 8 P-values which are tabulated in Table 4.

The random excursion variant test outputs 18 randomness probability values and they are recorded in Table 5. The minimum pass rate for the random excursion variant test is approximately equal to 1196 for a sample size of 1219 binary sequences.

Table 3: NIST software results

NIST test	P-value	Pass rate
Monobit	0.897763	1977/2000
Block frequency	0.909427	1976/2000
Cumulative sums forward	0.470189	1975/2000
Cumulative sums reverse	0.034712	1976/2000
Runs	0.869278	1980/2000
Longest run of ones	0.828164	1983/2000
Rank	0.540204	1982/2000
Spectral	0.829047	1986/2000
Non-overlapping templates	0.299009	1976/2000
Overlapping templates	0.715679	1976/2000
Universal	0.081261	1973/2000
Approximate entropy	0.942198	1981/2000
Serial 1	0.402080	1974/2000
Serial 2	0.134558	1973/2000
Linear complexity	0.866097	1976/2000

Table 4: Random excursion results

States	P-value	Pass rate
-4	0.799990	1207/1219
-3	0.385671	1200/1219
-2	0.845679	1206/1219
-1	0.010437	1205/1219
+1	0.301460	1208/1219
+2	0.747339	1204/1219
+3	0.821152	1208/1219
+4	0.228141	1209/1219

The novel self-shrinking chaos based pseudo-random generator passed successfully NIST tests.

Based on the good statistical records, it is clear that the proposed chaos based self-shrinking byte generator has good statistical characteristics and provide acceptable level of security.

Table 5: Random excursion variant results

States	P-value	Pass rate
-9	0.766873	1204/1219
-8	0.201911	1206/1219
-7	0.740745	1203/1219
-6	0.630236	1208/1219
-5	0.283723	1209/1219
-4	0.587342	1213/1219
-3	0.208233	1212/1219
-2	0.466754	1213/1219
-1	0.001462	1210/1219
+1	0.032851	1205/1219
+2	0.604458	1207/1219
+3	0.018587	1210/1219
+4	0.426663	1206/1219
+5	0.747339	1206/1219
+6	0.710655	1202/1219
+7	0.295462	1207/1219
+8	0.909773	1208/1219
+9	0.906449	1205/1219

3 Conclusions

In summary, the novel self-shrinking chaos based pseudo-random byte generation scheme is presented in this paper. The designed algorithm decimates Chebyshev recursive polynomial. Detailed security analysis on the proposed steps is given. Based on the results, we can conclude that the novel self-shrinking scheme provides valuable level of data security.

References

[1] A. Akhshani, A.Akhavan, A.Mobaraki, S.-C.Lima, Z.Hassan. Pseudo random number generator based on quantum chaotic map. *Commun. Nonlinear Sci.* 19:101-111, 2014.

[2] G. Alvarez, S. Li. Some Basic Cryptographic Requirements for Chaos Based Cryptosystems. *Int. J. Bifurcat. Chaos* 16:2129-2151, 2006.

[3] G. Chen, Y. Mao, C. K. Chui. A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps. *Chaos Solitons Fractals* 21:749-761, 2004.

[4] D. Coppersmith, H. Krawczyk, Y. Mansour. The shrinking generator. In *Lecture Notes in Computer Science* 773:22-39, 1993.

[5] M. Francois, T. Grosges, D. Barchiesi, R. Erra. *Informatica* 24:181-197, 2013.

[6] C. Fu, P. Wang, X. Ma, Z. Xu, W. Zhu. A Fast Pseudo Stochastic Sequence Quantification Algorithm Based on Chebyshev Map and Its Application in Data Encryption. In *Lecture Notes in Computer Science* 3991:826-829, 2006.

[7] T. Geisel, V. Fairen. Statistical properties of chaos in Chebyshev maps. *Phys. Lett. A* 105(6):263-266, 1984.

[8] A. Kanso. New Self-Shrinking Generator. In *Proceedings of the Security and Protection of Information Conference* 3:69-74, 2003.

[9] L. Kocarev, J. Makraduli, P. Amato. Public-key encryption based on Chebyshev polynomials. *Circ. Syst. Sign. Pr.* 24:495-517, 2005.

[10] J. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory* 15:122-127, 1969.

[11] W. Meier, O. Staffelbach. The self-shrinking generator. In *Lecture Notes in Computer Science* 950:205-214, 1994.

[12] M.A. Murillo-Escobar, C. Cruz-Hernandez, L. Cardoza-Avendano, and R. Mendez-Ramirez. A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dyn.* 87:407-425, 2017.

[13] A. Rukhin, J. 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 Application. NIST Special Publication 800-22, Revision 1a (Revised: April 2010), Lawrence E. Bassham III, 2010.

[14] B. Stoyanov. Pseudo-random bit generation algorithm based on Chebyshev polynomial and Tinkerbell map. *Appl. Math. Sci.* 8:6205-6210, 2014.

[15] B. Stoyanov. Pseudo-random bit generator based on Chebyshev map. In *AIP Conference Proceedings* 1561:369-372, 2013.

[16] L. Yang, T. Xiao-Jun. A new pseudorandom number generator based on a complex number chaotic equation. *Chin. Phys. B* 9:090506, 2012.

[17] J. Walker. ENT: A Pseudorandom Number Sequence Test Program, http://www.fourmilab.ch/random/, 2004.

[18] IEEE Computer Society. *IEEE standard for binary floating-point arithmetic*. ANSI/IEEE Std. 754, 1985.