Advanced Studies in Theoretical Physics
Vol. 9, 2015, no. 9, 411 - 421
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/astp.2015.5342

Novel Secure Pseudo-Random Number Generation Scheme Based on Two Tinkerbell Maps

Borislav Stoyanov

Department of Computer Informatics
Faculty of Mathematics and Informatics
Konstantin Preslavski University of Shumen, 9712 Shumen, Bulgaria

Krasimir Kordov

Department of Computer Informatics
Faculty of Mathematics and Informatics
Konstantin Preslavski University of Shumen, 9712 Shumen, Bulgaria

Abstract

The chaotic map based pseudo-random number generators with good statistical properties have been widely used in modern cryptography algorithms. This paper proposes a Tinkerbell map as a novel pseudo-random number generator. We evaluated the proposed approach with various statistical packages: NIST, DIEHARD and ENT. The results of the analysis demonstrate that the new derivative bit stream scheme is very suitable for embedding in critical cryptographic applications.

PACS: 03.67.Dd, 07.05.Pj, 07.05.Kf

Keywords: Tinkerbell map, pseudo-random number generator

1 Introduction

In the last two decades there has been a fast-growing interest in chaotic maps as a pseudo-random number generators in communications and information sciences. Cryptographic protocols for key exchange, identification, symmetric encryption or authentication have embedded pseudo-random number generators.

In [15], a pseudo-random algorithm based on a second-order chaotic digital filter is proposed. A logistic map as a pseudo-random number scheme is described in [4]. In [17], a cross-coupled tent map based bit generator is developed. A pseudo-random bit generator based on the combination of the logistic map and middle square method is presented in [19]. In [10], a pseudo-random generator based on the exact solution to the logistic map is proposed. A new technique for generating random-looking binary digits based on logistic map is presented in [13].

Reference [11] proposed a one-dimensional iterative chaotic map with infinite collapses within symmetrical region $[-1, 0) \cup (0, 1]$. In [6], a novel chaotic system, built on trigonometric functions, is proposed. The new chaotic system is used, in conjunction with a binary operation, in the designing of a new pseudo-random bit generator algorithm.

A method for chaos based encryption of data items by an arithmetic operation with example is provided in [14]. The examples of chaos equations include: Lorenz attractor, Tinkerbell map, Gumowski/Mira map, etc.

Inpired from [7] and [8] and with respect of [14] and [9], the aim of the paper is to propose a new chaos-based pseudo-random bit generator based only of two Tinkerbell maps, which has suitable features for embedding in various cryptographic applications.

2 Tinkerbell Map as a Pseudo-Random Number Generator

The Tinkerbell map [2] is a two-dimensional discrete-time dynamical system given by:

$$\begin{aligned} x_{n+1} &= x_n^2 - y_n^2 + ax_n + by_n \\ y_{n+1} &= 2x_n y_n + cx_n + dy_n \ , \end{aligned} \tag{1}$$

where $a = 0.9$, $b = -0.6013$, $c = 2.0$ and $d = 0.50$. The Tinkerbell map is illustrated in Figure 1.

We construct a new pseudo-random number algorithm which modifies the solutions of two Tinkerbell maps. The proposed pseudo-random bit generator is based on the following equations:

$$\begin{aligned} x_{1,n+1} &= x_{1,n}^2 - y_{1,n}^2 + ax_{1,n} + by_{1,n} \\ y_{1,n+1} &= 2x_{1,n} y_{1,n} + cx_{1,n} + dy_{1,n} \\ x_{2,m+1} &= x_{2,m}^2 - y_{2,m}^2 + ax_{2,m} + by_{2,m} \\ y_{2,m+1} &= 2x_{2,m} y_{2,m} + cx_{2,m} + dy_{2,m} \ , \end{aligned} \tag{2}$$

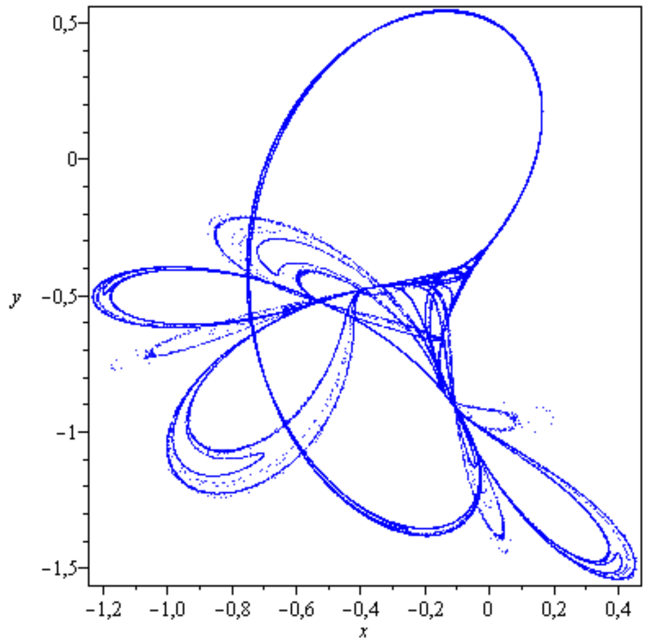

Figure 1: Tinkerbell map

where initial values $x_{1,0}, y_{1,0}, x_{2,0}$ and $y_{2,0}$ are used as a key.

Step 1: The initial values $x_{1,0}, y_{1,0}, x_{2,0}$ and $y_{2,0}$ of the two Tinkerbell maps from Eqs. (2) are determined.

Step 2: The first and the second Tinkerbell maps from Eqs. (2) are iterated for M and N times, respectively, to avoid the harmful effects of transitional procedures, where M and N are different constants.

Step 3: The iteration of the Eqs. (2) continues, and as a result, two real fractions $y_{1,n}$ and $y_{2,m}$ are generated and preprocessed as follows:

$$\begin{aligned} y_{1,n} &= abs(mod(integer(y_{1,n} \times 10^9), 2) \\ y_{2,m} &= abs(mod(integer(y_{2,m} \times 10^9), 2), \end{aligned} \tag{3}$$

where $abs(x)$ returns the absolute value of x, $integer(x)$ returns the the integer part of x, truncating the value at the decimal point, $mod(x, y)$ returns the reminder after division.

Two output bits are obtained.

Step 4: Return to Step 3 until the bit stream limit is reached.

The proposed generator is implemented by software simulation in C++ language, using the following initial seed: $x_{1,0} = -0.145622309652631$, $y_{1,0} = -0.742799703451115$, $M = 730$, $x_{2,0} = -0.634155080322761$, $N = 830$, and $y_{2,0} = -0.332344590085382$, stated as a key K1.

3 Analysis of the Tinkerbell Map based Pseudo-Random Number Generator

3.1 Key Space

The key space is a set of all possible keys that can be used in the initial seed of the pseudo-random scheme. The novel algorithm has six secret keys $x_{1,0}$, $y_{1,0}$, M, $x_{2,0}$, $y_{2,0}$, and N. According to the IEEE floating-point standard [12], the computational precision of the 64-bit double-precision number is about 10^{-15}. If we assume the precision of 10^{-9}, the secret key's space is more than 2^{183}.

In Table 1, we have compared the key space of our method with references [7],[14], and [17]. The key size of 2^{183} is larger than the other pointed pseudo-random schemes and sufficient enough to defeat brute-force attacks [3].

Generator	Key Space (Bin.)
Proposed	2^{183}
Reference [7]	2^{173}
Reference [17]	2^{64}
Reference [14]	2^{56}

Table 1: Key spaces of the proposed algorithm and some other algorithms.

3.2 Key Sensitivity Test

A typical property of the chaotic maps and pseudo-random number generators is to be sensitive to small changes in the initial conditions. We have performed the correlation coefficients test [1],[5], before and after slight modification of the key space of the novel Tinkerbell map based pseudo-random number generator, as follows:

(1) $x_{1,0} = -0.145622309652631$; a sequence with 1,000,000 bytes is generated, then a new sequence by slight modification of the initial condition $x'_{1,0} = -0.145622309652632$ is generated.

(2) $y_{1,0} = -0.742799703451115$; a sequence with 1,000,000 bytes is generated, then a new sequence by slight modification of the initial condition $y'_{1,0} = -0.742799703451116$ is generated.

(3) $x_{2,0} = -0.634155080322761$; a sequence with 1,000,000 bytes is generated, then a new sequence by slight modification of the initial condition $x'_{2,0} = -0.634155080322762$ is generated.

(4) $y_{2,0} = -0.332344590085382$; a sequence with 1,000,000 bytes is generated, then a new sequence by slight modification of the initial condition $y'_{2,0} = -0.332344590085383$ is generated.

The correlation coefficient r between two adjacent bytes (a_i, b_i) is computed in accordance with the way described in [5].

$$r = \frac{cov(a,b)}{\sqrt{D(a)}\sqrt{D(b)}}, \tag{4}$$

where

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

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

$$cov(a,b) = \sum_{i=1}^{M}(a_i - \bar{a})(b_i - \bar{b}), \tag{7}$$

M is the total number of couples (a_i, b_i), obtained from the byte sequences, and $\bar{a}$, $\bar{b}$ are the mean values of a_i and b_i, respectively. The correlation coefficient can range in the interval $[-1.00; +1.00]$.

Table 2 shows the results of adjacent bytes correlation coefficients calculations of the first and the second generated sequences.

Key Part 1	**Key Part 2**	**Corr. Coeff.** r
$x_{1,0} = -0.145622309652631$	$x'_{1,0} = -0.145622309652632$	-0.0014
$y_{1,0} = -0.742799703451115$	$y'_{1,0} = -0.742799703451116$	0.00043
$x_{2,0} = -0.634155080322761$	$x'_{2,0} = -0.634155080322762$	-0.000031
$y_{2,0} = -0.332344590085382$	$y'_{2,0} = -0.332344590085383$	-0.0002

Table 2: Correlation coefficients of four pairs of pseudo-random sequences.

It is clear that the proposed pseudo-random number 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 scheme are similar with results of three other pseudo-random number schemes [1], [8] and [24].

3.3 Speed Test

The novel pseudo-random number generator is measured on 2.8 GHz Pentium IV personal computer. In Table 3, we have compared the speed of our method with references [22],[25], and [26]. The data show that the novel pseudo-random number scheme has a satisfactory speed.

Generator	**Speed (Mbit/s)**
Proposed	0.4901
Reference [26]	0.4844
Reference [25]	0.3798
Reference [22]	0.2375

Table 3: Speeds of the proposed algorithm and some other algorithms.

3.4 Period and Linear Complexity

The period length and linear complexity of one hundred sequences of length L=100,000 of the novel scheme were computed using SAGE [21]. Our results are analogous to those reported by Kanso and Smoui [13]. Each tested binary sequence had large period length of L and linear complexity value of $(L/2)\pm 1$.

3.5 Experimental Statistical Tests

In order to measure randomness of the bits sequences generated by the new pseudo-random number scheme, we used NIST [20], DIEHARD [16] and ENT [23] statistical packages.

The NIST statistical test suite (version 2.1.1) includes 15 tests, which focus on the randomness of binary sequences produced by either hardware or software based number generators. These tests are: frequency (monobit), block-frequency, cumulative sums, runs, longest run of ones, rank, Fast Fourier Transform (spectral), non-overlapping templates, overlapping templates, Maurer's "Universal Statistical", approximate entropy, random excursions, random-excursion variant, serial, and linear complexity.

1000 sequences of 1000000 bits were produced using the new scheme. The results from all statistical tests are given in Table 4.

NIST statistical test	**Proposed Generator**	
	P-value	**Pass rate**
Frequency (monobit)	0.500279	994/1000
Block-frequency	0.576961	990/1000
Cumulative sums (Forward)	0.668321	993/1000
Cumulative sums (Reverse)	0.624627	991/1000
Runs	0.029205	990/1000
Longest run of Ones	0.686955	990/1000
Rank	0.587274	987/1000
FFT	0.896345	989/1000
Non-overlapping templates	0.528107	990/1000
Overlapping templates	0.883171	989/1000
Universal	0.538182	989/1000
Approximate entropy	0.751866	992/1000
Random-excursions	0.548333	643/649
Random-excursions Variant	0.527817	642/649
Serial 1	0.922855	988/1000
Serial 2	0.607993	984/1000
Linear complexity	0.841226	993/1000

Table 4: NIST Statistical test suite results for 1000 sequences of size 10^6-bit each generated by the proposed generator

The entire NIST test is passed successfully: all the $P-values$ from all 1000 sequences are distributed uniformly in the 10 subintervals and the pass rate is also in acceptable range. The minimum pass rate for each statistical test with the exception of the random-excursion (variant) test is approximately 980 for a sample size of 1000 binary sequences for both pseudo-random generators. The minimum pass rate for the random excursion (variant) test is approximately 634 for a sample size of 649 binary sequences for the proposed random algorithm.

Based on the results from the NIST tests the Tinkerbell map based random generator is suitable for cryptographic applications.

The DIEHARD suite consists of a number of different statistical tests: birthday spacings, overlapping 5-permutations, binary rank (31 x 31), binary rank (32 x 32), binary rank (6 x 8), bitstream, Overlapping-Pairs-Sparse-Occupancy, Overlapping-Quadruples-Sparse-Occupancy, DNA, stream count-the-ones, byte-count-the-ones, 3D spheres, squeeze, overlapping sums, runs up, runs down, craps. For the DIEHARD tests, we generated two files with 80 million bits each, from the proposed chaotic pseudo-random bit generators. The results are given in Table 5.

All of the DIEHARD *P-values* are in acceptable range of $[0, 1)$, hence the Tinkerbell map based streams are with highly unpredictable zeros and ones.

The ENT package performs 6 tests (Entropy, Optimum compression, χ^2 distribution, Arithmetic mean value, Monte Carlo π estimation, and Serial correlation coefficient) to sequences of bytes stored in files and outputs the

DIEHARD statistical test	**Proposed Generator *P-value***
Birthday spacings	0.444124
Overlapping 5-permutation	0.455073
Binary rank (31 x 31)	0.679847
Binary rank (32 x 32)	0.383671
Binary rank (6 x 8)	0.510800
Bitstream	0.412834
OPSO	0.432322
OQSO	0.401211
DNA	0.495023
Stream count-the-ones	0.648846
Byte count-the-ones	0.492608
Parking lot	0.501457
Minimum distance	0.505244
3D spheres	0.536202
Squeeze	0.945790
Overlapping sums	0.362025
Runs up	0.303442
Runs down	0.543410
Craps	0.925022

Table 5: DIEHARD statistical test results for two 80 million bits sequences generated by the proposed generator

results of those tests. We tested output of the two strings of 125000000 bytes of the proposed pseudo-random number generator. The results are summarized in Table 6.

ENT statistical test	**Proposed Generator results**
Entropy	7.999999 bits per byte
Optimum compression	OC would reduce the size of this 125000000 byte file by 0 %.
χ^2 distribution	For 125000000 samples is 240.50, and randomly would exceed this value 73.40 % of the time.
Arithmetic mean value	127.5140 (127.5 = random)
Monte Carlo π estim.	3.141558194 (error 0.00 %)
Serial correl. coeff.	-0.000077 (totally uncorrelated = 0.0)

Table 6: ENT statistical test results for two 80 million bits sequences generated by the proposed generator.

The proposed number generator passed all the tests of ENT. This demonstrate that the novel scheme is suitable for encryption/decryption and statistical sampling applications.

Conclusions

We have presented a strongly pseudo-random number derivative scheme constructed by the solutions of only two Tinkerbell maps. The key space, key sensitivity, speed, period, linear complexity and package statistical tests analysis results demonstrate that the new algorithm can assure high level of pseudo-randomness in critical cryptographic applications.

Acknowledgements. This work is partially supported by the Scientific research fund of Konstantin Preslavski University of Shumen under the grant No. RD-08-305/12.03.2015.

References

[1] A. Akhshani, A. Akhavan, A. Mobaraki, S. C. Lim, Z. Hassan, Pseudo Random Number Generator Based on Quantum Chaotic Map, *Communications in Nonlinear Science and Numerical Simulation,* **19** (2014), 101 - 111. http://dx.doi.org/10.1016/j.cnsns.2013.06.017

[2] K. T. Alligood, T. D. Sauer, J. A. Yorke, *CHAOS: An Introduction to Dynamical Systems,* Springer-Verlag, Berlin, 1996. http://dx.doi.org/10.1007/978-3-642-59281-2

[3] G. Alvarez, S. Li, Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, *International Journal of Bifurcation and Chaos,* **16** (2006), 2129 - 2151. http://dx.doi.org/10.1142/s0218127406015970

[4] M. Andrecut, Logistic Map as a Random Number Generator, *International Journal of Modern Physics B,* **12** (1999), 921 - 930. http://dx.doi.org/10.1142/s021797929800051x

[5] G. Chen, Y. Mao, C. K. Chui, A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps, *Chaos Solitons Fractals* **21** (2004), 749-761. http://dx.doi.org/10.1016/j.chaos.2003.12.022

[6] A. Dăscălescu, R. E. Boriga, A. Diaconu, Study of a New Chaotic Dynamical System and Its Usage in a Novel Pseudorandom Bit Generator, *Mathematical Problems in Engineering,* **2013** (2013), Article ID 769108, 1-10. http://dx.doi.org/10.1155/2013/769108

[7] M. François, D. Defour, P. Berthomé, A Pseudo-Random Bit Generator Based on Three Chaotic Logistic Maps and IEEE 754-2008 Floating-Point Arithmetic, *Lecture Notes in Computer Science,* **8402** (2014), 229 - 247. http://dx.doi.org/10.1007/978-3-319-06089-7_16

[8] M. François, T. Grosges, D. Barchiesi, R. Erra, Pseudo-Random Number Generator Based on Mixing of Three Chaotic Maps, *Communications in Nonlinear Science and Numerical Simulation,* **19** (2014), 887 - 895. http://dx.doi.org/10.1016/j.cnsns.2013.08.032

[9] A. Goldsztejn, W. Hayes, P. Collins, Tinkerbell Is Chaotic, *SIAM Journal on Applied Dynamical Systems,* **10** (2011), 1480 - 1501. http://dx.doi.org/10.1137/100819011

[10] J. A. González, R. Pino, A Random Number Generator Based on Unpredictable Chaotic Functions, *Computer Physics Communications,* **120** (1999), 109 - 114. http://dx.doi.org/10.1016/s0010-4655(99)00233-7

[11] D. He, C. He, L. Jiang, H. Zhu, G. Hu, Chaotic characteristics of a one-dimensional iterative map with infinite collapses, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* **48** (2001), 900 - 906. http://dx.doi.org/10.1109/81.933333

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

[13] A. Kanso, N. Smaoui, Irregularly Decimated Chaotic Map(s) for Binary Digits Generations, *International Journal of Bifurcation and Chaos,* **19** (2009), 1169 - 1183. http://dx.doi.org/10.1142/S0218127409023573

[14] H. S. Lambert, *Method and Apparatus for Encryption of Data,* US Patent 7133522 B2, Nov. 7, 2006.

[15] T. Lin, L. O. Chua, A New Class of Pseudo-Random Number Generator Based on Chaos in Digital Filters, *International Journal of Circuit Theory and Applications,* **21** (1993), 473 - 480. http://dx.doi.org/10.1002/cta.4490210506

[16] G. Marsaglia, *Diehard: a Battery of Tests of Randomness.* http://www.fsu.edu/pub/diehard/

[17] N. K. Pareek, V. Patidar, K. K. Sud, A Random Bit Generator Using Chaotic Maps, *International Journal of Network Security,* **10** (2010), 32 - 38.

[18] V. Patidar, K. K. Sud, N. K. Pareek, A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing, *Informatica,* **33** (2009), 441 - 452.

[19] H. Rahimov, M. Babaie, H. Hassanabadi, Improving Middle Square Method RNG Using Chaotic Map, *Applied Mathematics,* **2** (2011), 482 - 486. http://dx.doi.org/10.4236/am.2011.24062

[20] 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. http://csrc.nist.gov/rng/

[21] W. A. Stein et al. Sage Mathematics Software (Version 6.1.1), *The Sage Development Team,* 2014. http://www.sagemath.org

[22] X. Tong, M. Cui, Feedback Image Encryption Algorithm with Compound Chaotic Stream Cipher Based on Perturbation, *Science in China Series F: Information Sciences,* **53** (2010), 191 - 202. http://dx.doi.org/10.1007/s11432-010-0010-3

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

[24] X. Wang, W. Zhang, W. Guo, J. Zhang, Secure Chaotic System with Application to Chaotic Cipher, *Information Sciences,* **221** (2013), 555 - 570. http://dx.doi.org/10.1016/j.ins.2012.09.037

[25] M. J. Werter, An Improved Chaotic Digital Encoder, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* **45** (1998), 227 - 229. http://dx.doi.org/10.1109/82.661656

[26] L. Yang, T. Xiao-Jun, A New Pseudorandom Number Generator Based on a Complex Number Chaotic Equation, *Chinese Physics B,* **21** (2012), 1 - 7. http://dx.doi.org/10.1088/1674-1056/21/9/090506

Received: April 1, 2015; Published: May 18, 2015