Extended Security Analysis of the Zaslavsky Maps Based Pseudorandom Byte Generator

Borislav Stoyanov[1] [0000-0002-7307-5914], Tsvetelina Ivanova[1] [0000-0002-0775-6551], Mihaela Todorova[1] [0000-0002-8622-9732], Miroslav Cholakov[2] [0000-0002-5352-6263] and Hyudyaim Ahmed[3] [0000-0003-0189-9719]

University of Shumen, Universitetska Str. 115, Shumen, Bulgaria

[1]{borislav.stoyanov, ts.r.ivanova, mihaela.todorova}@shu.bg, [2]moltenhardrock@abv.bg, [3]bytess91@gmail.com

Abstract. The Zaslavsky maps based pseudorandom byte generator is a recent and secure cryptographic primitive. Extended security analysis of the generator is presented. We evaluated the output byte properties by period and linear complexity calculation and DieHarder and PractRand statistical packages. This gives the motivation to consider the Zaslavsky systems based pseudorandom byte generator as reasonable for basic cryptographic applications in encryption process.

Keywords: Zaslavsky Map, Pseudorandom Generator, Security Analysis.

1 Introduction

Cryptanalysis is the study of analyzing cipher text, ciphers and cryptographic primitives with the aim of understanding how they operate and finding and improving methods for defeating or destroying them. Pseudorandom byte generators are distinct algorithms that use mathematical formulas to output bytes with random like properties.

The use of chaos maps as a secure cryptographic system in the last thirty years has been at the prominence of dynamical equations. In [1], a new scheme for generating pseudorandom numbers based on Duffing map is presented. In [2], a security evaluation of the pseudorandom bit out algorithm based on multi-modal maps is proposed. Novel cryptanalysis of the number for video encryption is designed in [3]. In [4], a chaos game based pseudorandom number generator is presented. Number of pseudorandom bit output algorithms based on chaotic systems are designed in [5], [6], and [7].

Zaslavsky maps [8] based pseudorandom byte generator is presented in [9]. The generator is described in details, and the cryptographic analysis, including initial key evaluation and statistical analysis, is carried out.

The aim of the article is to present new analysis of some cryptographic properties of Zaslavsky maps based pseudorandom byte generator. In Section 2, the steps of the generator are described. Section 3 presents calculations of the period and the linear complexity, and DieHarder [10] and PractRand [11] statistical results are given. Finally, the last section concludes the article.

2 Description of the Zaslavsky Maps based Pseudorandom Byte Generator

The mathematical expression of the Zaslavsky equations is given by:

$$y_{n+1} = \mathrm{mod}(y_n+\nu(1+\mu z_n)+\varepsilon\nu\mu\cos(2\pi y_n),1) \tag{1}$$

$$z_{n+1}=e^{-r}(z_n+\varepsilon\cos(2\pi y_n) \tag{2}$$

where

$$\mu=1-e^{-r}/r, \tag{3}$$

$r = 3.0$, $\nu = 400/3$, and $\varepsilon = 0.3$. The generator under present study is based on two Zaslavsky chaotic systems. The initial parameters $y_{1,0}$, $y_{2,0}$, $z_{1,0}$, and $z_{2,0}$ are real numbers. With each iteration, four real values $y_{1,i}$, $y_{2,j}$, $z_{1,i}$, and $z_{2,j}$, are generated, then converted to 256 bit values, and XOR-ed. Pseudorandom byte m is outputted.

3 Extended Security Analysis of the Zaslavsky Maps based Pseudorandom Byte Generator

3.1 Period and Linear Complexity

The period and linear complexity of two hundred sequences of length M=200,000 of the generator were computed using SAGE [12]. The values obtained are comparable to those reported in [13]. Each tested binary sequence had huge period of M and linear complexity value of $(M/2)\pm1$.

3.2 Experimental Testing with Statistical Packages

The DieHarder package (version 3.31.1) is a random number generator-testing suite. This testing and benchmarking software program consists tests presented in Table 1 *and* Table 2.

The DieHarder output results are presented in Table 1 and Table 2.

Table 1. DieHarder test results, I part.

Test name	p-value	Assessment
diehard birthdays	0.56483962	passed
diehard operm5	0.26303032	passed
diehard rank 32x32	0.52810343	passed

diehard rank 6x8	0.95957035	passed
diehard bitstream	0.53527613	passed
diehard opso	0.91909082	passed
diehard oqso	0.86167844	passed
diehard dna	0.93016479	passed
diehard count 1s stream	0.22990683	passed
diehard parking lot	0.88193571	passed
diehard 2dsphere	0.47770491	passed
diehard 3dsphere	0.97957902	passed
diehard squeeze	0.97235302	passed
diehard sums	0.02026052	passed
diehard runs	0.39874109, 0.90351885	passed
diehard craps	0.36983582, 0.86498446	passed

Table 2. DieHarder test results, II part.

Test name	p-value	Assessment
count the 1s byte	0.24207567	passed
marsaglia tsang gcd	0.50132978, 0.67555797	passed
sts monobit	0.72146044	passed
sts runs	0.40218825	passed
sts serial	0.01146032--0.98601086	passed
rgb bitdist	0.01123223--0.95692540	passed
rgb minimum distance	0.11964788--0.90910943	passed
rgb permutations	0.37055571--0.96669890	passed
rgb lagged sum	0.08486053--0.97133190	passed
rgb kstest test	0.84077327	passed
dab bytedistrib	0.76852095	passed
dab dct	0.53079707	passed
dab filltree	0.22119502, 0.70442029	passed
dab filltree2	0.90468519, 0.94557881	passed
dab monobit2	0.94581666	passed

The second package is PractRand. We tested our pseudorandom scheme for bytes up to 2^{24} bytes in length, Table 3 *and* Table *4*.

Table 3. PractRand test results, I part.

Test name	Raw	Processed
BCFN(2,13):!	R= +0.0	“pass”
BCFN(2+0,13-3)	R= -0.6	p = 0.588
BCFN(2+1,13-4)	R= -2.2	p = 0.814
BCFN(2+2,13-5)	R= +1.9	p = 0.202
BCFN(2+3,13-5)	R= -0.3	p = 0.526
BCFN(2+4,13-6)	R= +1.6	p = 0.236
BCFN(2+5,13-6)	R= -2.5	p = 0.860
BCFN(2+6,13-7)	R= -3.0	p = 0.921
BCFN(2+7,13-8)	R= +5.0	p = 0.033
DC6-9x1Bytes-1	R= +3.0	p = 0.121

Table 4. PractRand test results, II part.

Test name	Raw	Processed
Gap-16:!	R= +0.0	“pass”
Gap-16:A	R= +2.1	p = 0.147
Gap-16:B	R= +0.8	p = 0.284
[Low1/8]BCFN(2,13):!	R= +0.0	“pass”
[Low1/8]BCFN(2+0,13-5)	R= -2.3	p = 0.829
[Low1/8]BCFN(2+1,13-6)	R= -4.6	p = 0.988
[Low1/8]BCFN(2+2,13-6)	R= -1.3	p = 0.683
[Low1/8]BCFN(2+3,13-7)	R= -1.2	p = 0.662
[Low1/8]BCFN(2+4,13-8)	R= -0.2	p = 0.468
[Low1/8]DC6-9x1Bytes-1	R= -1.3	p = 0.829
[Low1/8]Gap-16:!	R= +0.0	“pass”
[Low1/8]Gap-16:A	R= +1.9	p = 0.159
[Low1/8]Gap-16:B	R= -0.5	p = 0.634

The size of initial values can guarantee long period and good linear complexity of the output stream. All probability values, calculated from DieHarder and PractRand tests are in acceptable range of [0, 1). Based on these results we can conclude that all of statistical tests are passing successfully and the generator under present study is very suitable for critical cryptographic applications.

4 Conclusions

We have presented new security analysis of the Zaslavsky functions based pseudorandom byte generator. The results from the period, linear complexity, and DieHarder and PractRand statistical packages evaluation, indicate that the studied algorithm is reasonable for basic cryptographic applications in derivative encryption schemes.

5 Acknowledgement

This work is partially supported by the Bulgarian Ministry of Education and Science under the National Program for Research "Young Scientists and Postdoctoral Students". This work is partially supported by the Scientific research fund of Shumen University under the grant No. RD-08-107/02.02.2021.

References

1. Riaz, M., Ahmed, J., Shah, R., Hussain, A.: Novel Secure Pseudorandom Number Generator Based on Duffing Map. Wireless Personal Communications 99, 85–93 (2018).
2. Lambić, D.: Security analysis of the pseudo-random bit generator based on multi-modal maps. Nonlinear Dynamics 91, 505–513 (2018).
3. Lambić, D., Janković, A., Ahmad, M.: Security Analysis of the Efficient Chaos Pseudo-random Number Generator Applied to Video Encryption. Journal of Electronic Testing 34, 709–715 (2018).
4. Aybi, P., Setayeshi, S., Rahmani, A.: Deterministic chaos game: A new fractal based pseudo-random number generator and its cryptographic application. Journal of Information Security and Applications 52, 102472 (2020).
5. Wang, L., Cheng, H.: Pseudo-Random Number Generator Based on Logistic Chaotic System. Entropy 21(10), 960 (2019).
6. Lv, X., Liao, X., Yang, Bo.: A novel pseudo-random number generator from coupled map lattice with time-varying delay. Nonlinear Dynamics 94, 325–341 (2018).
7. Barani, M., Ayubi, P., Valandar, M., Irani, B.: A new Pseudo random number generator based on generalized Newton complex map with dynamic key. Journal of Information Security and Applications 53, 102509 (2020).
8. Zaslavsky, G.: The simplest case of a strange attractor. Physics Letters A 69(3), 145–147 (1978).
9. Stoyanov, B., Todorova, M., Ivanova, T., Borboryan, G., Hasanov, A.: Two Zaslavsky maps in pseudorandom byte generation. In: Todorov, M. (ed.) AMiTaNS'19, AIP Conference Proceedings, vol. 2164, pp. 120013. American Institute of Physics (2019).
10. Brown, R., Eddelbuettel, G., Bauer, D.: DieHarder: A random number test suite, https://webhome.phy.duke.edu/~rgb/General/dieharder.php, last accessed 2021/03/28.
11. Doty-Humphrey, C.: Practrand: C++ library of pseudo-random number generators and statistical tests for rngs, http://pracrand.sourceforge.net/, last accessed 2021/03/28.
12. Stein, W.: Sage mathematics software (v.9.2), https://www.sagemath.org/, last accessed 2021/03/28.
13. Stoyanov, B., Kordov, K.: Novel Secure Pseudo-Random Number Generation Scheme Based on Two Tinkerbell Maps. Advanced Studies in Theoretical Physics 9(9), 411–421 (2015).