ONE MODIFICATION OF THE ALTERNATING STEP GENERATOR

Borislav Stoyanov

Konstantin Preslavsky University of Shumen, Bulgaria
borislav.stoyanov@shu.bg

Abstract: *In this paper, we propose a research on the modified alternating step generator implementation based on feedback with carry shift registers. The scheme is proposed by Schneier. We calculated the period of the derivative algorithm. The key gamma is statistically tested with NIST test suite. The result of the analysis shows that the output data are such a random physical phenomena generated.*

Keywords: *alternating step generator, pseudorandom bit generator, cryptographic analysis, NIST test suit.*

1. Introduction

Information technology is a vital part of our everyday life. Guarding its secret is the crux of the matter. Despite the variety of encryption algorithms, we are still exposed to glaring omissions in the protection of personal data by some multinational companies. To illustrate my point, here is an example of Ref. [1] where information was extracted so that the passwords of hundred millions of people using Facebook Lite, millions of others Facebook users and hundreds using Instagram, were stored in a text format without any protection.

The question of looking for new cryptographic primitives is always a priority for all of the crypto researchers whose job is the protection of critical data. This way companies will have more to choose from when it comes to protecting their users' data.

Generators of pseudo-random numbers are a crucial primitive to a number of algorithms for information technology protection. In this discourse is explored an option of the interchanging generator of pseudo-random numbers. Its aptness is motivated in the cryptographic systems.

2. One Modification of the Alternating Step Generator

The classical alternating step generator [2] is based on linear feedback shift registers (LFSR) [3]. Because of successive cryptanalysis Schneier propose few modifications of this generator [4], one of them with feedback with carry shift registers FCSRs [5], Fig. 1.

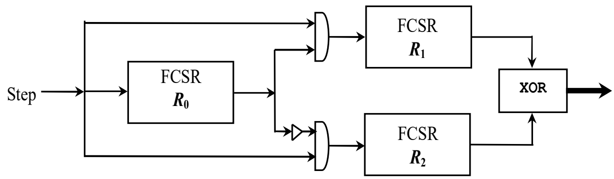

Figure 1. Modification of the Alternating step generator with FCSRs

3. Computer Realization

Let the period lengths of the output sequences of FCSRs are T_0=d_0-1, T_1=d_1-1, and T_2=d_2-1 [5]. Then, if the lengths of R_1 and R_2 are coprime, the period of modified alternating step generator will be $S_0=(T_0.T_1.T_2)/GCD(T_1,T_2)$ and the linear complexity is $\lambda(Z)\geq\log_2(S_0+1)$. The modified alternating step generator is coded by C++ class p_adic [6] with connection coprime integers in interval from 984059 to 2305883.

4. Security Analysis

To evaluate the randomness profile of the output stream generated by the analyzed generator, we used NIST statistical test package [7]. The NIST software includes sixteen tests. The tests fix on a variety of different types of non-randomness that could exist in a sequence. These tests are: monobit, frequency within a block, runs, longest-run-of-ones in a block, binary matrix rank, discrete spectral, non overlapping template matching, overlapping template matching, universal, linear complexity, serial, approximate entropy, cumulative sums, random excursions, random excursions variant.

The testing process consists of the following steps [7], [8]:

(1) State the null hypothesis. Assume that the binary sequence is random.

(2) Compute a sequence test statistic. Testing is carried out at the bit level.

(3) Compute the p-value, $\mathrm{p-value} \in [0, 1]$.

(4) Compare the $\mathrm{p-value\ to}\ \alpha$. Fix α, where $\alpha \in (0.0001, 0.01]$. *Success* is declared whenever $\mathrm{p-value} \geq \alpha$; otherwise, *failure* is declared.

Given the empirical results for a particular statistical test, the NIST suite computes the proportion of sequences that pass. The range of acceptable proportion is determined using the confidence interval defined as, $\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$, where $\hat{p} = 1-\alpha$, and *m* is the number of binary tested sequences. In our two setups *m*=1000. Thus the confidence interval is $0.99 \pm 3\sqrt{\frac{0.99(0.01)}{1000}} = 0.99 \pm 0.0094392$. The proportion should lie above 0.9805607.

The distribution of p-values is examined to ensure uniformity. The interval between 0 and 1 is divided into 10 sub-intervals, and the p-values that lie within each sub-interval are counted. Uniformity may also be specified trough an application of a χ^2 test and the determination of a p-value corresponding to the Goodness-of-Fit Distributional Test on the p-values obtained for an arbitrary statistical test, p-value of the p-values. This is implemented by computing $\chi^2 = \sum_{i=1}^{10} \frac{(F_i - m/10)^2}{m/10}$, where F_i is the number of p-values in sub-interval i, and m is the number of tested sequences. A *p*-value is calculated such that $\text{p-value}_T = igamc(9/2, \chi^2/2)$. If $\text{p-value}_T > 0.0001$, then the streams can be regarded to be uniformly distributed.

1, 000, 000, 000 bits were generated using the modification of the Alternating Step Generator. The results are tabulated in Table 1.

Table 1: NIST test results

NIST statistical test	***p-value***	**pass rate**
monobit	0.877083	991/1000
mlock-frequency	0.415422	992/1000
cumulative sums (Forward)	0.346443	1000/1000
cumulative sums (Reverse)	0.616305	994/1000
runs	0.360287	993/1000
longest run of ones	0.373625	990/1000
rank	0.481479	990/1000
spectral	0.490483	993/1000
nNon overlapping templates	0.496351	992/1000
overlapping templates	0.007584	994/1000
universal	0.953089	997/1000
approximate entropy	0.1455750	989/1000
random-excursions	0.026588	598/601
random-excursions variant	0.915549	598/601
serial 1	0.270849	995/1000
serial 2	0.596371	993/1000
linear-complexity	0.877083	990/1000

The minimum pass rate is more than 980 and the *p_values* are in range > 0.007583. Compared with similar pseudo-random algorithms [8–13] the analyzed one has good statistical results, the output data are such a random physical phenomena generated.

Conclusion

We have presented theoretical and statistical results of the modified alternating step generator based on feedback with carry shift registers. Based on the analysis the modified alternating scheme has suitable characteristics for cryptographic primitives.

Acknowledgements

This work is partially supported by the Scientific research fund of Konstantin Preslavsky University of Shumen under the grant No. RD-08-74/30.01.2019.

References

1. Canahuati, P. (2019) Keeping Passwords Secure, https://newsroom.fb.com/news/2019/03/keeping-passwords-secure/.
2. Gunther C.G. (1987) Alternating Step Generators Controlled by De Brujin Sequences, Proc. EUROCRYPT'87, LNCS, 304, pp. 5–14, https://link.springer.com/chapter/10.1007%2F3-540-39118-5_2.
3. Klein, A. (2013) Linear Feedback Shift Registers. In: Stream Ciphers, Springer, London.
4. Schneier, B. (1996) Applied Cryptography. New York: John Wiley & Sons.
5. Klapper, A., Goresky, M. (1997) Feedback Shift Registers, 2-Adic Span, and Combiners With Memory, Journal of Cryptology, 10 (2), 111–147, https://doi.org/10.1007/s001459900024.
6. Stoyanov, B., Bedzhev, B., Zhekov, Zh. (2004) Computation Model of p-adic Arithmetic, XXXIX International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST, 341-344, http://www.icestconf.org/wp-content/uploads/2016/proceedings/icest_2004_01.pdf.
7. L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, N. Heckert, and Dray J. (2010) A Statistical test suite for random and pseudorandom number generators for cryptographic application, NIST Special Publication 800-22, Revision 1a (Revised: April 2010), http://doi.org/10.6028/NIST.SP.800-22r1a.
8. Tasheva Z, Bedzhev B, Stoyanov B. N-adic Summation-Shrinking Generator. Basic properties and empirical evidences. IACR Cryptology ePrint Archive. 2005;2005:68.
9. Tasheva, Z., Bedzhev, B., Stoyanov, B. (2005) P-adic shrinking-multiplexing generator, 2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 443-448, https://doi.org/10.1109/IDAACS.2005.283020.
10. Stoyanov, B., Bedzhev, B., Tashev, T., Vasileva, S. (2007) Application and statistic testing of the 5-adic summation-shrinking generator, CompSysTech '07 Proceedings of the 2007 international conference on Computer systems and technologies. Article No. 39, https://doi.org/10.1145/1330598.1330641.

11. Stoyanov, B., Kordov, K. (2014) Pseudorandom Bit Generator with Parallel Implementation, In: Lirkov I., Margenov S., Waśniewski J. (eds) Large-Scale Scientific Computing. LSSC 2013. Lecture Notes in Computer Science, vol 8353. Springer, Berlin, Heidelberg, 557-564, https://link.springer.com/chapter/10.1007/978-3-662-43880-0_64.
12. Wicik; R., Rachwalik, T. (2013) Modified Alternating Step Generators, 2013 Military Communications and Information Systems Conference, https://ieeexplore.ieee.org/document/6695519.
13. Wicik; R., Rachwalik, T., Gliwa, R. (2014) Modified Alternating Step Generators with Non-Linear Scrambler, Annales UMCS Informatica AI XIV, 1 (2014), 61–74, https://journals.umcs.pl/ai/article/view/3391/2585.