Annales UMCS Informatica AI XIV, 1 (2014) 61–74
DOI: 10.2478/umcsinfo-2014-0003

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Modified Alternating Step Generators with Non-Linear Scrambler

Robert Wicik[1*], Tomasz Rachwalik[1†], Rafał Gliwa[1‡]

[1] *Military Communication Institute, Cryptology Department, Zegrze, Poland*

Abstract – Pseudorandom generators, which produce keystreams for stream ciphers by the exclusive-or sum of outputs of alternately clocked linear feedback shift registers, are vulnerable to cryptanalysis. In order to increase their resistance to attacks, we introduce a non-linear scrambler at the output of these generators. Non-linear feedback shift register plays the role of the scrambler. In addition, we propose Modified Alternating Step Generator with a non-linear scrambler (MASG_{1S}) built with non-linear feedback shift register and regularly or irregularly clocked linear feedback shift registers with non-linear filtering functions.

1 Introduction

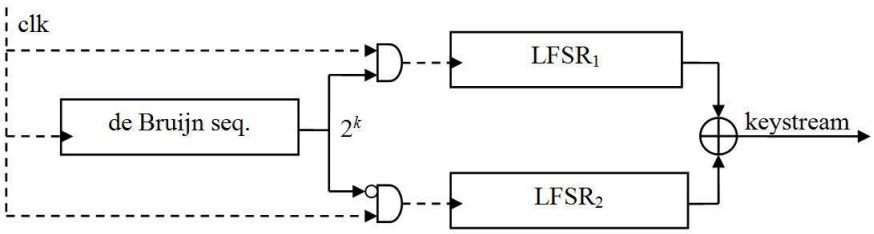
Pseudorandom generators of a keystream composed of linear feedback shift registers (LFSR) are basic components of classical stream ciphers. An LFSR with properly selected feedback gives a sequence of maximal period and good statistical properties but has a low linear complexity. It is vulnerable to the Berlecamp-Massey [**1**] algorithm and can be easily reconstructed having a short output segment. Stop and go or alternating clocking of shift registers are two of the methods to increase linear complexity of the keystream. Other techniques introduce non-linearity to the feedback or to the output of the shift register. All these methods increase resistance of keystream generators to reconstruction of the internal state as well as the member functions from the output sequence.

*r.wicik@wil.waw.pl
†t.rachwalik@wil.waw.pl
‡r.gliwa@wil.waw.pl

In the alternating step generator (ASG) [**3**], the de Bruijn sequence [**2**] controls the irregular clocking of two linear feedback shift registers. The ASG is vulnerable to various attacks [**4, 5, 6, 7, 8, 9, 10, 11, 12, 13**], so there are many modifications of this generator [**14, 15, 16, 17**]. In [**19**] we proposed the next three modifications: MASG, MASG_0 and MASG_1. These modified alternating step generators give sequences with maximal period, good statistical properties and higher linear complexity than the ASG. The introduction of non-linear functions to the generator increases its resistance to the attacks. However, from the analysis of the attacks we conclude that at the output of the alternating step generator a linear function (XOR) should be replaced with a non-linear one. Proposed in [**19**] MASG_2 with non-linear combining function at the output gives non-random sequences. Therefore, we have undertaken further work to improve the MASG family. In this paper, we describe selected attacks on alternating step generators. Then we propose another modification of such generators in order to increase their resistance to these attacks. The modification consists in adding a non-linear scrambler at the output of alternating step generators. We constructed the non-linear scrambler with the maximal period of non-linear feedback shift register. The MASG_{1S} keystream generator, with an implemented non-linear scrambler, non-linear filtering functions and initialization method is particular realization of this idea.

2 Alternating Step Generators

The alternating step generator (ASG) [**3**] is a pseudorandom generator of binary keystream sequences, where the concept of stop-and-go shift registers was adapted. The ASG consists of two linear feedback shift registers, alternately clocked by the de Bruijn sequence [**2**]. The de Bruijn sequence of the period $K = 2^k$ can be easily obtained by adding zero bit after $k-1$ zeros in the sequence with the period $2^k - 1$ from the LFSR (the modified de Bruijn sequence). The exclusive-or sum (XOR) of bits from the irregularly clocked LFSRs produces output bits from the generator, as it is presented in Fig. 1.



Rysunek 1. The Alternating Step Generator

For properly selected feedback polynomials, the output sequence from the ASG has a large period (1) and a high linear complexity (2):

$$T = M_1 M_2 2^k \tag{1}$$

$$(m_1 + m_2)2^{k-1} < L \leq (m_1 + m_2)2^k \tag{2}$$

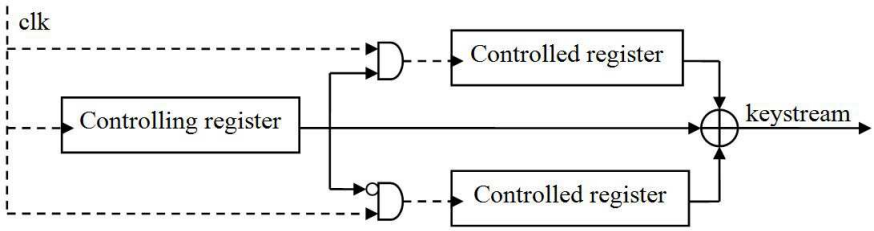
We can observe growth of the linear complexity of the output sequence from the ASG in comparison to the sequence obtained from a simple LFSR (where the linear complexity is equal to its length – m_1 or m_2 in this case).

The ASG is vulnerable to various attacks. There are many variants of correlation and algebraic attacks and the best two are described in [**4**] and [**9**]. Asymptotic time complexity of these attacks is $O(m^2 2^{2m/3})$ and the data complexity is $O(2^{2m/3})$, where m is the length of the shortest register in the ASG. Time complexity of the algebraic attack described in [**12**] is much higher, however this attack can be applied if polynomials of irregularly clocked registers are unknown, while requiring less output bits. These attacks exploit dependencies between the output sequence (for the known plaintext) and the internal states of irregularly controlled registers.

In order to resist the ASG against these attacks, there are proposed many modifications of this generator. In the alternating step generator [**14**] ASG(r, s), two positive integers r and s determine how many times one register (LFSR_1) or the other one (LFSR_2) is clocked. In [**11**] the authors showed, that the ASG(r, s) is as secure as the original ASG. Afterwards, Kanso proposed in [**15**] and [**16**] the MGCCASG and MCCASG constructions based on the ASG(r, s), where the integers r and s are variable – dependent on a key or on a function of a state of the controlling register. Another method of improving the ASG, proposed in [**10**], was to exchange some LFSRs for feedback with carry shift registers (FCSR) and the XOR sum to be added to the (ADD) as an output function. This modification of the ASG does not improve its security substantially.

2.1 Modified Alternating k-Generators

Modified alternating k-generators (MAG_k) were proposed in [**17**]. Output sequence from MAG_k is produced by the XOR sum of binary sequences from all three shift registers, as presented in Fig. 2. Feedback functions of these registers can be linear or non-linear.



Rysunek 2. The Modified Alternating k-Generator

There are more modifications of the MAG_k proposed in [**11**]:

(1) $MAG_k{}^1$ – the function of state bits of the controlling register determines how many times controlled registers are clocked – this generator is similar to MCCASG [**16**];
(2) $MAG_k{}^2$ – the binary output of the function (inner control function) of state bits of the controlling register determines alternating clocking of controlled registers – this generator was analyzed in [**12**], where the authors showed that its security is not better that of the original ASG;
(3) $MAG_k{}^3$ – the output from the generator is produced by the function (*output generating function*) of binary states of all three registers: one controlling and two controlled – this generator is similar to our concept described in [**20**] and to the $MASG_2$ described in 19.

2.2 The MASG Family

In 18 and [**19**] we proposed a family of modified alternating step generators (MASG). We concentrated on selecting proper non-linear functions – as feedback functions and the others as filtering and combining functions. In [**21, 22, 23**] there are described methods for constructing non-linear feedback functions for shift registers. At this time we can achieve registers with the maximal period for the length up to $n = 31$. These registers give sequences with the linear complexity close to the period, maximum 2^{n-2} Our first approach to modification of the ASG was to replace controlled registers ($LFSR_1$ and $LFSR_2$ in Fig. 1) by the non-linear feedback shift registers (NLFSR) and we achieved:

- MASG – the alternating step generator, where the output is produced by the XOR sum of binary sequences from two alternately clocked NLFSRs;
- $MASG_0$ – the alternating step generator, where the output is produced by the XOR sum of binary sequences from all three registers (like in MAG_k).

These MASGs produce binary sequences with a better linear complexity than the ASG, but we should find NLFSRs with a greater length than 31 ($n > 64$). These registers should give sequences with the maximal period and the high non-linear order. The non-linear Boolean functions are often used as filtering or combining functions for linear feedback shift registers in order to increase security of keystream generators. The functions proposed in [**18**] and [**19**] have high non-linearity and many non-linear components in their algebraic normal form. We used them in our second approach to modification of the ASG and we achieved:

- $MASG_1$ is the MAG_k, where all linear feedback shift registers are equipped with non-linear filtering functions;

- MASG_2 is the MAG_k with a non-linear output function.

Output sequences from these constructions have better linear complexities than the ASG. MASG_1 gives sequences, which seem to be random, while MASG_2 did not pass randomness tests.

3 Attacks on the Alternating Step Generators

There are many attacks on the alternating step generators. The most of them are divide-and-conquer attacks with a known plaintext. The main goal is to find initial states of shift registers having a portion of the output sequence.

3.1 Divide-And-Conquer Attack

Divide-and-conquer attack was presented by C. G. Günther in [**3**], when describing original ASG. The basis of the attack is that the output sequence may be divided into two parts, derived from regularly clocked registers. Then these subsequences can be tested for a low linear complexity in an easy way using the Berlekamp-Massey algorithm. If a tested sequence with period of 2^k is consistent with a sequence from clock control register, then linear complexity of component sequences for irregularly clocked registers is lower than their periods. The complexity of the divide-and-conquer attack, if one knows only feedbacks of the register, for which initial state is searched, is $O(min^2(m_1, m_2)2^k)$. When one knows feedbacks of all registers, then the complexity of the attack is $O(min(m_1, m_2)2^k)$ and instead of linear complexity – a linear consistency test is applied. In both cases, a guessing clock control register is necessary.

3.2 Edit Distance Correlation Attack

To carry out the edit distance correlation attack [**5**], it must be assumed, that feedbacks of irregularly clocked registers are known and the clocking sequence is characterized by a uniform distribution of bits 0 and 1. The attack involves searching the entire space of initial states of alternately clocked registers with known feedbacks, followed by verifying whether they are appropriate. Verification is based on the Hamming distance between the computed segment of the output sequence (obtained as the output of the generator with fixed states of alternately clocked registers) and the segment obtained as a result of the attack with a known plaintext. This distance is the minimum number of necessary subtractions (edit distance) in the computed segment, which allows obtaining the known output sequence. Minimum is calculated for all $(2k)$ states of clock control register. There exists [**5**] an effective method of calculating the distance and it is possible to determine the probability, that this distance is equal to 0, i.e. when initial sequences give the known output sequence of the generator for the specified clock control sequence. This probability increases with the length of the known segment of the output sequence. The length of required known segment of the output sequence is

linear in relation to the sum of lengths of irregularly clocked registers. The number of multiple solutions is minimized when the available output sequence is 4 times longer than the total length of registers, which are searched. The computational complexity of this attack is $O((m_1 + m_2)^2 2^{m_1+m_2})$. The third register can be restored with the complexity $O(2^{0.27k})$, if only a sufficiently long segment of the sequence is available.

3.3 Edit Probability Correlation Attack

The edit probability correlation attack on individual irregularly clocked registers in the Günther generator was proposed in [**6**]. The attack uses probability (edit probability) that a given segment of the output sequence of the generator has been produced from the sequence derived from the regularly clocked register with the predetermined initial state. Finding the initial state of one of the irregularly clocked registers can be done without knowledge of the other one and without knowledge about the state of clock control register. The edit probability correlation attack requires a known output sequence with the length minimum 4 times longer than that of state of the register, which is searched. The complexity of calculating this probability is the square of the length of output sequence. The computational complexity of this attack, in order to find both initial states of the irregularly clocked registers is $O(max^2(m_1+m_2)2^{max(m_1+m_2)})$. For long registers, the complexity of edit probability correlation attack is much lower than that of the edit distance correlation attack.

3.4 Johansson's Attack

In [**4**] the attack with reduced complexity on generators with irregularly clocked registers was proposed. In the output sequence of the generator, a segment of consecutive zeros (or ones) is searched. It is assumed that half of them come from one of the irregularly clocked registers. This occurs with a certain probability. The remaining bits are obtained by exhaustive search. The optimal computational complexity of this attack is $O(m^2 2^{2/3m})$ and requires $O(2^{2/3m})$ bits of sequence, where m is the length of the register, which is searched: m_1 or m_2. These complexities apply to the attack both the first and the second irregularly clocked registers.

In another scenario, the segment of a number of ones (or zeros) in the output sequence is searched and it is assumed that half of ones (or zeros) of that segment has originated from one register, and the rest (ones and zeros) from the other. This occurs with a certain probability. The complexities of the attack according to this scenario are similar to these mentioned above for one register. Finding the initial state of the second register may require better quantity of calculations.

3.5 New Reduced Complexity Attack

New reduced complexity attack is based on low resistance of Günther generator to sampling [**9**]. The low resistance to sampling indicates the possibility of effective finding all possible register preimages $A(Z^n)$ of a generator, for a given segment of

output sequence (Z^n). Generally, this resistance is defined as 2^{-n}, where n is the maximum available length of output sequence.

In order to execute the attack, first, the set of all possible states for a given segment of output sequence of length n is searched. Algorithm for finding this set is based on the divide-and-conquer attack with the parity test. For all states of the initial clock control register, the output segment is divided into bits, originated from particular irregularly clocked registers. Then all states of irregularly clocked registers are checked if they can generate separate bits – if so, the possible states of three registers are added to the set of $A(Z^n)$, which is searched.

The average number of initial states of Günther generator for a given segment of output sequence is 2^{3m-n} or $n \leq 3m$ where m is the length of registers, n – the length of segment of output sequence. The computational complexity of the algorithm is $O(max(2^m, 2^{3m-n}))$. The complexity is determined by the factor 2^m when the size of a set of possible initial states is $\leq 2^m$, otherwise it is determined by the value 2^{3m-n}.

This algorithm can be effective and it shows low resistance of generator to sampling, where $n \leq 2m$, that is, when resistance to sampling is about 2^{-2m}, where $2m$ is the total length of irregularly clocked registers. In a modified version of the algorithm, T random elements of set $A(Z^n)$ can be found. However, in this case the question is how big this set should be to include a correct initial state of one of the registers. In [**9**], formulas (2) and (3) determine the probability and the conditional entropy of solutions. Generally, the reduced complexity attack is to find initial states of the Günther generator among a set of possible initial states. The most likely solutions are found using the edit probability calculated for each possible initial state and a given segment of a output sequence. The complexity of this attack for a random segment of output sequence with the weight of w and the length n is $O(m^2 2^{m\gamma})$, where m is the length of the register, which is searched, γ depends on m and $\gamma < 1$. For $\gamma < 1$ and for $h(w/n) = 2/3$ the attack is similar to that in [**4**] and the asymptotic complexities are as follows: computational $O(m^2 2^{2/3m})$, memory $O(2^{2/3m})$. In comparison with the Johansson attack, the reduced complexity attack is more flexible in terms of useful output sequences, whose weights can be freely chosen.

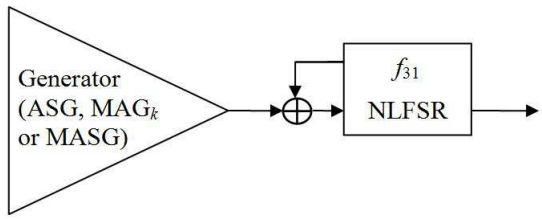
3.6 Algebraic Attack

The algebraic attack on stream ciphers with irregularly clocked registers was presented in [**8**]. The complexity of the attack on the original Günther generator is $O((m_1{}^3 + m_2{}^3))$. In [**11, 12, 13**] there were described further similar attacks on the modified generators with the alternately clocked registers, as $\mathrm{MAG}_k{}^2$ or $\mathrm{ASG}(r, s)$. Such attacks have higher computational complexity than the Johansson's attack, but they need a shorter, known output sequence. They use a linear relationship (XOR) between the sequences from registers at the output of the generator and they find sequences of individual irregularly clocked registers by searching among all possible initial states of clock control register. In the case of the attack on the modified k-generator of the second type: $\mathrm{MAG}_k{}^2$, for known feedbacks of registers, the attack needs $k+m_1+m_2$

bits of output sequence to find the initial states of registers. The complexity of the attack is then . When feedbacks of registers are not known, it must be the Berlekamp-Massey algorithm additionally used, hence it is required to know $k+2m_1+2m_2$ bits of output sequence to execute the attack and the complexity is $O((m_1{}^2+m_2{}^2)2^{k+1})$. To avoid such attack, the output of the generator should not be defined either by a linear function or by a function that will approximately describe linear relationships between the output of the generator and the outputs of individual registers.

4 Non-linear Feedback Shift Register as a Scrambler

In the previous chapter, we described attacks on the alternating step generators. These attacks explore linearity of the transformation at the output and a low linear complexity of shift registers. Hence, in the MASG family we proposed non-linear functions – some as non-linear feedbacks of shift registers, others as filtering or combining functions of linear feedback shift registers. The known non-linear feedback shift registers, which give maximal length output sequences, are too short for practical applications. Therefore, MASG and MASG_0 are the constructions, which do not ensure sufficient resistance to the attacks. Also MASG_2 with the non-linear combining function gives a sequence, which is not random. Hence, the MASG_1 built with the linear feedback shift registers with the non-linear filtering functions is the best choice from the MASG family. MASG_1 can be the basis for the construction of secure generator for stream cipher. The ASG, MAG_k and MASG_1 have linear functions at the output. The analysis, given in section 3, suggests non-linear transformation at this point. So we propose non-linear multiplicative scrambler as an output function of the alternating step generators. As the scrambler, we use a non-linear feedback shift register with the maximal period and the linear complexity close to the period. The general scheme of the generator with the scrambler is presented in Fig. 3.



Rysunek 3. Generator with the non-linear scrambler

The output sequence from the alternating step generator is applied to the input of the scrambler, where bit after bit is added (mod 2) to its non-linear feedback. The example of the non-linear feedback function of the scrambler might look like this [**23**] f_{31}:

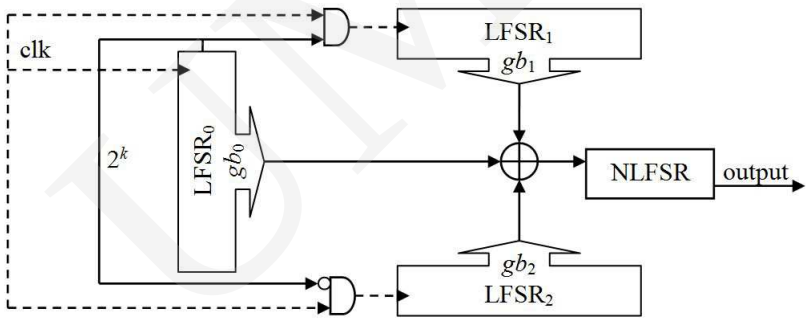
$$s_0 + s_2 + s_6 + s_7 + s_8 + s_9 + s_{10} + s_{14} + s_{15} + s_{16} + s_{20} + s_{26} + s_{29} + \\ + s_4 s_5 s_7 s_9 s_{12} \bar{s}_{19} \bar{s}_{21} s_{22} s_{24} s_{25} \bar{s}_{26} \bar{s}_{29} + s_4 s_5 s_7 s_9 s_{12} s_{19} s_{21} s_{22} s_{24} s_{25} s_{26} s_{29} \quad (3)$$

where:

$s_0, s_1, \ldots, s_{30}$ are the bits of the NLFSR state register;
$\bar{s}_i = s_i + 1$ for $i = 0, 1, \ldots, 30$; the addition and multiplication are performed using modulo 2 operation.

4.1 MASG$_1$ with the Scrambler

The scheme of the MASG$_1$ with the non-linear scrambler (MASG$_{1S}$) is presented in Fig. 4. Controlling (LFSR$_0$) and controlled (LFSR$_1$ and LFSR$_2$) shift registers have linear feedbacks and are equipped with the non-linear filtering functions gb_0, gb_1 and gb_2 [**19**]. The controlling register and non-linear feedback shift register are clocked regularly. The controlled registers are clocked alternately. Lengths of LFSR$_0$, LFSR$_1$, LFSR$_2$ and NLFSR are $k = 127$, $m_1 = 131$, $m_2 = 137$ and $n = 31$ respectively.



Rysunek 4. $MASG_{1S}$

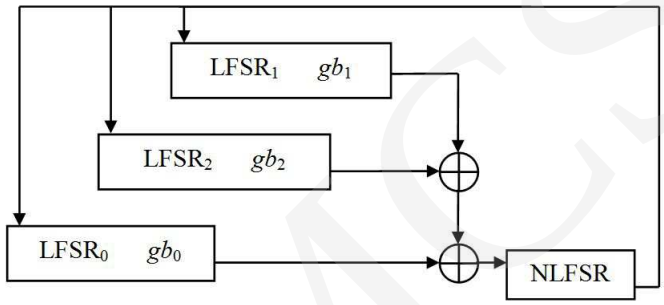
4.2 Initializing the $MASG_{1S}$

MASG$_{1S}$ requires 426 bits for the initial states of registers. The key for the contemporary stream ciphers should be in the range 160-256 bits. So let us assume that the key bits are assigned to the state registers of LFSR$_0$, LFSR$_1$, LFSR$_2$ and NLFSR, according to the Table 1. The remaining cells of the state registers are filled with one-bits (this protects them from filling only by zeros).

After initial filling, according to the rules described above, the generator is clocked 853 times. The output of the MASG$_{1S}$ is added (mod 2) to all linear registers (controlling and controlled ones). During this process, the generator does not produce output bits, LFSR$_0$ and NLFSR are clocked regularly, LFSR$_0$ and NLFSR are clocked alternately. The scheme of the MASG$_{1S}$ initializing process is presented in Fig. 5.

TABLICA 1. Distribution of the key to the registers

Key length	**160**	**192**	**224**	**256**
LFSR$_0$	43	53	64	75
LFSR$_1$	43	54	64	75
LFSR$_2$	43	54	65	75
NLFSR	31	31	31	31



RYSUNEK 5. Initializing MASG$_{1S}$

4.3 Randomness Properties

We checked experimentally randomness of keystreams produced by alternating step generators: ASG, MAG$_k$ and MASG$_1$ with the non-linear scrambler (3): ASGS, MAG$_{kS}$ and MASG$_{1S}$. We tested the randomness using seven basic statistical tests [**24**], [**25**]:

(1) frequency test
(2) serial test
(3) two bit test
(4) 8-bit poker test
(5) 16-bit poker test
(6) runs test (for max 22 consecutive zeros or ones)
(7) autocorrelation test (for shifted sequences by 1, 2, ..., 8 bits)

We tested 10 GB sequences produced by the ASGS, MAG$_{kS}$ and MASG$_{1S}$ starting with the randomly selected initial states. Additionally, we took 10 GB sequences from the random number generator SGCL-100M [**26**]. As the reference distributions the tests use the chi-square distributions and the standard normal distribution. The resulting statistics were split into 8 classes according to the range of significance level as it is shown in Table 2. For the popular level of significance =0.05, sequences pass tests if their statistics are a class A, B or C.

Tablica 2. Classes of statistics

Class	A B C	A	B	C	D	E	F	G	H
%	95	80	10	5	2.5	1.5	0.5	0.4	0.1

The obtained results of experiments for overall sequences are given in Table 3. Table 4 contains the percentages of classes of statistics for 1 MB subsequences of the examined sequences.

Tablica 3. Classes of statistics for 10GB sequences

Test no.	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ASGS	A	A	A	A	A	A	A A A A A A A A
MAG$_{kS}$	A	A	A	A	A	A	A A A C A B A A
MASG$_{1S}$	A	B	A	B	A	B	C A A A A A A C
SGCL-100M	A	A	A	A	A	A	B A A A C A A B

Tablica 4. Percentages of classes for 1MB subsequences

Class	**ABC**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
ASGS	94.97	79.61	10.39	4.97	2.61	1.55	0.42	0.36	0.09
MAG$_{kS}$	94.96	79.48	10.59	4.89	2.60	1.44	0.47	0.42	0.12
MASG$_{1S}$	94.98	79.82	10.52	4.91	2.59	1.46	0.48	0.38	0.11

Randomness tests results for ASGS, MAG$_{kS}$ and MASG$_{1S}$ are what we expected for random sequences:

- all sequences passed tests with the significance level $\alpha = 0.05$,
- percentages of classes are close to those expected (Table 2),
- results for alternating step generators with the non-linear scrambler are similar to those for the random number generator.

4.4 Resistance of Alternating Step Generators with Scrambler to Attacks

Attacks to the alternating step generators explore linearity of the XOR transformation at the output and linearity of feedback functions of shift registers. These known plaintext divide-and-conquer attacks rely on matching the sequence fragments to the registers in order to guess their initial states and the key. To protect alternating step generators from these types of attacks, we propose to add a non-linear scrambler to their outputs. We assume the worst case when the plaintext and feedback functions are known. Then, an attacker will have access to the output of the generator, but not to the XOR sum of the sequences of alternating step registers. Complexity of the attacks will increase by the factor that determines guessing the initial state of the non-linear feedback shift register i.e. minimum by 2^{n-1} for register of length n. We

propose non-linear feedback register as the scrambler. We constructed maximal period non-linear feedback shift registers up to $n = 31$. Currently known registers of such type have length $n = 34$. These are short registers and it seems, that complexity of the attacks will not increase significantly. However, the attacker should check if the initial state of the NLFSR is proper. That requires constructing the test. This will not be easy due to the random properties of the sequences before and after the scrambler. Presumedly, high complexity and low efficiency of the test leads to increased resistance of the generators to the attacks. But it requires further work.

Additionally, in the MASG_{1S} we used non-linear filtering function to each linear feedback shift register. The functions increase linear complexity of the sequences from these registers and protect them against the Berlecamp-Massey algorithm. non-linear scrambler, non-linear filtering functions and initialization method causes that the MASG_{1S} is resistant to the attacks dedicated to the alternating step generators.

5 Summary

In this paper, we have analyzed the attacks on the alternating step generators. These attacks explore linearity of shift registers and linearity of the output XOR function. In order to increase resistance to the attacks we proposed the non-linear scrambler at the output of the alternating step generators. Such scrambler can be built with the non-linear feedback shift register, which gives a sequence of full period and linear complexity close to the period.

We also used the non-linear scrambler at the output of the modified alternating step generator MASG_1. The resulting keystream generator MASG_{1S} is built of one non-linear feedback shift register and three linear feedback shift registers, each with a non-linear filtering function. We proposed the initializing method for the MASG_{1S}. This method produces initial states of shift registers from a key, before starting keystream generation. Adding non-linear filtering functions, non-linear scrambler and initialization to the generator substantially increases its resistance to the divide-and-conquer attacks. In general, the complexity of the attacks on alternating step generators will increase by the factor, which determines the difficulty of finding a proper initial state of the non-linear feedback shift register at the output (we assume that the plaintext and feedback functions are known). It does not seem to be a complex problem for short NLFSR, but the sequences before and after scrambler have good random properties. Hence, it will be not easy to find out, that the initial state of the NLFSR is correct. The complexity of the appropriate test will be the subject for further work. We checked randomness of the alternating step generators with a scrambler. The ASGS, MAG_{kS} and MASG_{1S} give keystreams, which pass randomness testes. The test results are similar to those for the true random number generator. Thus, generators with a non-linear scrambler can be used as keystream generators in stream ciphers.

Literatura

[1] Berlekamp E. R., Algebraic Coding Theory, Aegean Park Press (1984).
[2] Bruijn N. G. de, A combinatorial problem. Indag. Math., 8 (1946): 461–467.
[3] Günther C. G., Alternating step generator controlled by de Bruijn sequences, Advances in Cryptology – Eurocrypt'87, LNCS, vol. 304 (1988): 5–14.
[4] Johansson T., Reduced complexity correlation attacks on two clock-controlled generators, In Proceedings of Asiacrypt (1998): 342–356.
[5] Golic J., Menicocci R., Edit Distance Correlation Attack on the Alternating Step Generator, Advances in Cryptology – Crypto'97, LNCS, vol. 1294, (1997): 499–512.
[6] Golic J., Menicocci R., Edit Probability Correlation Attacks on the Alternating Step Generator. Sequences and Their Applications - SETA (1998).
[7] Golic J., Menicocci R., Correlation analysis of the Alternating Step Generator. Design, Codes and Cryptography, 31, Kluwer Academic Publishers (2004): 51–74.
[8] Al-Hinai S., Batten L., Colbert B., Wong K., Algebraic Attacks on Clock-Controlled Stream Ciphers, LNCS, vol. 4058, Springer (2006): 1–16.
[9] Khazaei S., Fisher S., Meier W., Reduced complexity attacks on the alternating step generator, Proceedings of SAC'07, Springer-Verlag (2007): 1–16.
[10] Su S., Chiu K., Wuu L., The Cryptanalysis of LFSR/FCSR based alternating step generator, ICCES (2006).
[11] Hassanzadeh M. M., Helleseth T., Algebraic attack on the alternating step(r,s) generator, Proceedings of the IEEE International Symposium on Information Theory, IEEE (2010): 2493–2497.
[12] Hassanzadeh M. M., Helleseth T., Algebraic attack on the second class of modified alternating k-generators, NISK Conference (2010).
[13] Hassanzadeh M. M., Helleseth T., Algebraic attack on the more generalized clock-controlled alternating step generators, Proceeding of SPCOM (2010): 1–5.
[14] Kanso A. A., The alternating step(r,s) generator, SECI, Tunis (2002).
[15] Kanso A. A., More generalized clock-controlled alternating step generator, Proceedings of ACNS'04, LNCS, vol. 3089 (2004): 326–338.
[16] Kanso A. A., Modified clock-controlled alternating step generator, Computer Communications, 32, Elsevier (2009): 787–799.
[17] Białota R., Kawa G., Modified alternating k-generators, Design, Codes and Cryptography, 35, Kluwer Academic Publishers (2005): 159–174.
[18] Wicik R., Rachwalik T., Modyfikacje generatora z naprzemiennym taktowaniem rejestrów, KSTiT, Gdańsk, Przegląd telekomunikacyjny, nr 8-9/2013 (2013): 1225–1230.
[19] Wicik R., Rachwalik T., Modified Alternating Step Generators, MCC, Saint-Malo, France (2013), Military Communications and Information Technology: Recent Advances in Selected Areas, WAT, Warszawa (2013), Cryptology ePrint Arch., 728 (2013).
[20] Borowski M., Wicik R., How to speed up a stream cipher, RCMCIS (2002), Biuletyn WIŁ, Zegrze, (2003).
[21] Rachwalik T., Szmidt J., Wicik R., Zabłocki J., Generation of nonlinear feedback shift registers with special purpose hardware, MCC, Gdańsk (2012), Cryptology ePrint Archive, 314 (2012).
[22] Szmidt J., Dąbrowski P., Łabuzek G., Rachwalik T., Searching for nonlinear feedback shift registers with parallel computing, MCC, Saint Malo, France (2013), Cryptology ePrint Arch., 542 (2013).
[23] Dąbrowski P., Łabuzek G., Rachwalik T., Szmidt J., Searching for Nonlinear Feedback Shift Registers with Parallel Computing, Information Processing Letters, 114 (2014): 268–272.
[24] Menezes A. J., Oorschot P. C. van, Vanstone S. A., Handbook of applied cryptography, CRC Press (1997).
[25] Wicik R., Borowski M., Randomness testing of some random and pseudorandom sequences, Military Commu-nication Conference, Prague (2008).

[26] Leśniewicz M., Sprzętowa generacja ciągów losowych z przepływnością 100 Mbit/s, Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, nr 11/2011 (2011): 1608–1613.