Modified Alternating Step Generators with Non-Linear Scrambler

Robert Wicik, Tomasz Rachwalik, Rafał Gliwa
Military Communication Institute, Cryptology Department, Zegrze, Poland
{r.wicik, t.rachwalik, r.gliwa} @ wil.waw.pl

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

Keywords. Stream Cipher, Alternating Step Generator, Linear and Non-Linear Feedback Shift Registers.

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 the output sequence of maximal period and good statistical properties but has low linear complexity. It is vulnerable to Berlecamp-Massey [1] algorithm and can be easy reconstructed having short output segment. Stop and go or alternating clocking of shift registers are two of methods to increase linear complexity of the keystream. Other techniques introduce nonlinearity 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.

In the alternating step generator (ASG) [3], the de Bruijn sequence [2] controls irregular clocking of two linear feedback shift registers. The ASG is vulnerable to various attacks [4..13], so there are many modifications of this generator [14..17]. In [19] we proposed 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 nonlinear 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 nonlinear one. Proposed in [19] $MASG_2$ with nonlinear combining function at the output gives nonrandom sequences. Therefore, we have undertaken further work to improving 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 lies in adding a nonlinear scrambler at the output of alternating step generators. We constructed the nonlinear scrambler with nonlinear feedback shift register of maximal period. Particular realization of this idea is the $MASG_{1S}$ keystream generator, with implemented nonlinear scrambler, nonlinear filtering functions and the initialization method in.

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 period $K=2^k$ can be easily obtained by adding zero bit after k-1 zeros in the sequence with period 2^k-1 from the LFSR (from modified de Bruijn sequence). The exclusive-or sum (XOR) of bits from irregular clocked LFSRs produces output bits from the generator, as it is presented in Fig. 1.

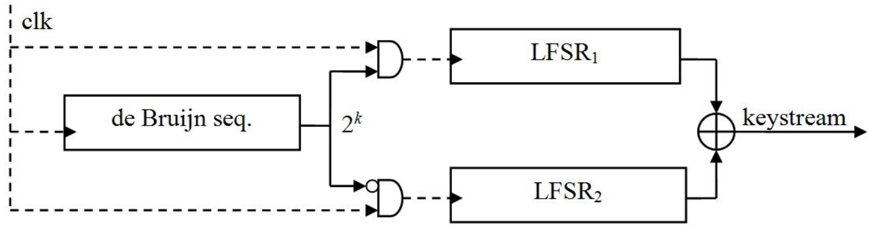

Fig. 1. The Alternating Step Generator

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

$$T= M_1M_22^k \tag{1}$$

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

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 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 irregular clocked registers are unknown, while requiring less output bits. These attacks exploit dependencies between output sequence (for known plaintext) and internal states of clock 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 is clocked one register (LFSR$_1$) or the other one (LFSR$_2$). In [11] authors showed, that the ASG(r,s) is as secure as the original ASG. Afterwards, Kanso proposed in [15] and [16] MGCCASG and MCCASG constructions based on the ASG(r,s), where integers r and s are variable – dependent on the key or on the function of the controlling register state.

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 nonlinear.

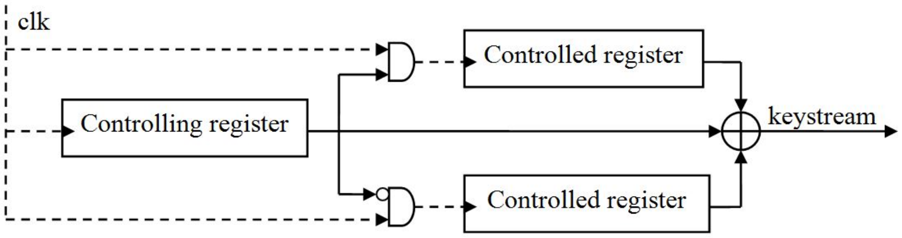

Fig. 2. The Modified Alternating *k*-Generator

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 authors showed that its security is not better than the security 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 registers: one controlling and two controlled ones – 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 nonlinear functions – ones as feedback functions and other ones as filtering and combining functions. In [21..23] there are described methods for constructing nonlinear feedback functions for shift registers. At this time we can achieve registers with maximal period for length up to n=31. These registers give sequences with 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 nonlinear 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 better linear complexity than the ASG, but we should find NLFSRs with greater length than 31 (n>64). These registers should give sequences with maximal period and high linear complexity.

Nonlinear Boolean functions are often used as filtering or combining functions for linear feedback shift registers in order to increase security of keystream generators. Functions proposed in [18] and [19] have high nonlinearity and many nonlinear components in their algebraic normal form. These functions we used 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 nonlinear filtering functions;
- MASG$_2$ – is the MAG$_k$ with nonlinear output function.

Output sequences from these constructions have better linear complexities than the ASG. MASG$_1$ gives sequences, which seems 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 known plaintext. 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 Berlekamp-Massey algorithm. If tested sequence with a period of 2^k is consistent with a sequence from clock control register, then the 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 – linear consistency test is applied. In both cases, 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 known output sequence. Minimum is calculated for all (2^k) states of clock control register. There exists [5] 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 known output sequence of the generator for the specified clock control sequence. This probability increases with the length of 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 four times longer than 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 Günther generator was proposed in [6]. The attack uses probability (edit probability) that given segment of the output sequence of the generator has been produced from the sequence derived from regularly clocked register with 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 length minimum 4 times longer than length 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 the complexity of the edit distance correlation attack.

3.4 Reduced complexity attack

In [4] it was proposed an attack with reduced complexity on generators with irregularly clocked registers. A segment of consecutive zeros (or ones) is searched in the output sequence of the generator. 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 [9] 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 both attacks – on the first and on the second irregularly clocked register.

In another scenario, the segment of some 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 higher quantity of calculations.

3.5 New reduced complexity attack

New reduced complexity attack is based on a low resistance of Günther generator to sampling [9]. The low resistance to sampling indicates the possibility of effective finding all possible register's preimages $A(Z^n)$ of a generator, for a given segment of the output sequence (Z^n). Generally, this resistance is defined as 2^{-n}, where n is the maximum available length of the 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 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.

Average number of initial states of Günther generator for a given segment of output sequence is 2^{3m-n} for $n \leq 3m$, where m is length of registers, n – 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 size of 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. But, in this case, the question is how big should be this set to include correct initial state of one of 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 Günther generator among a certain set of possible initial states. The most likely solutions are being found using the edit probability calculated for each possible initial state and given segment of output sequence. The complexity of this attack for a random segment of output sequence with weight of w and length of n is $O(m^2 2^{m\gamma})$, where m is the length of 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 the attack [4] and 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 attach is more flexible in terms of useful output sequences, which 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 $O((m_1^3 + m_2^3)2^k)$. In [11..13] there were described further similar attacks on modified generators with alternately clocked registers, as MAG_k^2 or ASG (r,s). Such attacks have higher computational complexity than Johansson's attack, but they need shorter, known output sequence. They use a linear relationship (XOR) between sequences from registers at the output of the generator and they find sequences of individual clocked irregularly 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: 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 $O(2^{k+1})$. When feedbacks of registers are not known, it must be 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 Nonlinear feedback shift register as a scrambler

In previous chapter, we described attacks on alternating step generators. These attacks explore linearity of the transformation at the output and low linear complexity of shift registers. Hence, in the MASG family we proposed some nonlinear functions – ones as nonlinear feedbacks of shift registers, others as filtering or combining functions of linear feedback shift registers.

Known nonlinear feedback shift registers, which give maximal length output sequences, are too short for practical applications. Therefore, MASG and MASG$_0$ are constructions, which do not ensure sufficient resistance to the attacks. As well as MASG$_2$ with nonlinear combining function gives sequence, which is not random. Hence, the MASG$_1$ built with linear feedback shift registers with nonlinear 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 a nonlinear transformation at this point. So we propose a nonlinear multiplicative scrambler as an output function of the alternating step generators. As the scrambler, we use the nonlinear feedback shift register with maximal period and linear complexity close to the period. The general scheme of the generator with the scrambler is presented in Fig. 3.

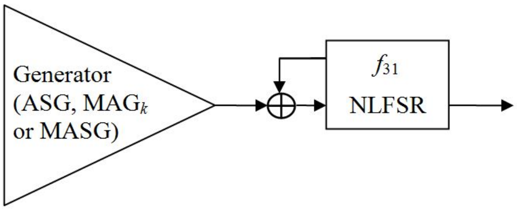

Fig. 3. Generator with the nonlinear scrambler

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 nonlinear feedback. The example of nonlinear feedback function of the scrambler might look like this [23] f_{31}:

$$\begin{aligned} &s_0+s_2+s_6+s_7+s_8+s_9+s_{10}+s_{14}+s_{15}+s_{16}+s_{20}+s_{21}+s_{26}+s_{29}+\\ &s_4s_5s_7s_9s_{12}\bar{s}_{19}\bar{s}_{21}s_{22}s_{24}s_{25}\bar{s}_{26}\bar{s}_{29}+s_4s_5s_7s_9s_{12}s_{19}s_{21}s_{22}s_{24}s_{25}s_{26}s_{29} \end{aligned} \tag{3}$$

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

4.1 MASG$_1$ with the scrambler

The scheme of the MASG$_1$ with the nonlinear 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 nonlinear filtering functions gb_0, gb_1 and gb_2 [19]. Controlling register and nonlinear feedback shift register are clocked regularly. 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.

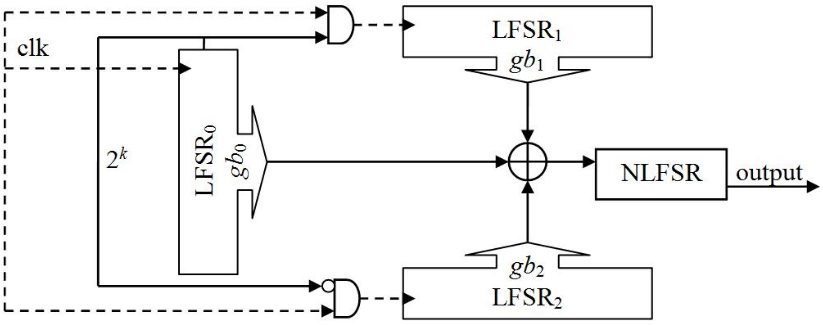

Fig. 4. MASG$_{1S}$

4.2 Initializing the MASG$_{1S}$

MASG$_{1S}$ requires 426 bits for initial states of registers. The key for contemporary stream ciphers should be in range 160-256 bits. So let us assume that 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).

Table 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

After initial filling, according to the rules described above, 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, 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.

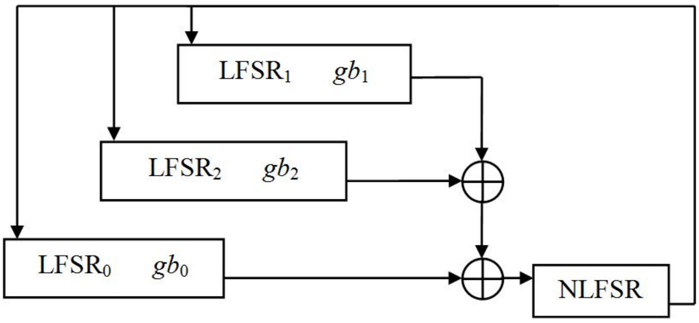

Fig. 5. Initializing MASG$_{1S}$

4.3 Randomness properties

We experimentally checked randomness of keystreams produced by alternating step generators: ASG, MAG$_k$ and MASG$_1$ with the nonlinear scrambler (3): ASG$_S$, MAG$_{kS}$ and MASG$_{1S}$. We tested the randomness using seven basic statistical tests [24], [25]: frequency test, serial test, two bit test, 8-bit poker test, 16-bit poker test, runs test (for max 22 consecutive zeros or ones), autocorrelation test (for shifted sequences by 1, 2, …, 8 bits).

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

Table 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

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

Table 3. Classes of statistics for 10GB sequences

Test no.	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ASG$_S$	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

Table 4. Percentages of classes for 1MB subsequences

Class	ABC	A	B	C	D	E	F	G	H
ASG$_S$	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
SGCL	95.05	79.48	10.33	5.14	2.57	1.42	0.44	0.41	0.11

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

- all sequences passed tests with significance level α=0.05,
- percentages of classes are close to those expected (Table 2),
- results for alternating step generators with the nonlinear scrambler are similar to results 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. 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 nonlinear 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 nonlinear feedback shift register i.e. minimum by 2^{n-1} for register of length n.

We propose nonlinear feedback register as the scrambler. We constructed maximal period nonlinear 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 initial state of the NLFSR is proper. That requires constructing the test. This will not be easy due to the random properties of 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 nonlinear filtering function to each linear feedback shift register. The functions increases linear complexity of the sequences from these registers and protect them against Berlecamp-Massey algorithm. Nonlinear scrambler, nonlinear filtering functions and initialization method strengthen resistance of MASG$_{1S}$ to the attacks dedicated to alternating step generators.

5 Summary

In this paper, we have analyzed attacks on 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 nonlinear scrambler at the output of the alternating step generators. Such scrambler can be built with the nonlinear feedback shift register, which gives sequence of full period and linear complexity close to the period.

We also used the nonlinear scrambler at the output of the modified alternating step generator MASG$_1$. The resulting keystream generator MASG$_{1S}$ is built of one nonlinear feedback shift register and three linear feedback shift registers, each with nonlinear filtering function. We proposed initializing method for the MASG$_{1S}$. This method produces initial states of shift registers from a key, before starting keystream generation. Adding nonlinear filtering functions, the nonlinear scrambler and the initialization phase to the generator substantially increases its resistance to the divide-and-conquer attacks with known plaintext.

In general, the complexity of the attacks on alternating step generators will increase by the factor, which determines the difficulty of finding proper initial state of the nonlinear 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 sequences before and after scrambler have good random properties. Hence, it will be not easy to find, 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 the nonlinear scrambler. The ASG$_S$, MAG$_{kS}$ and MASG$_{1S}$ give keystreams, which pass randomness testes. Test results are similar to the results for the true random number generator. Thus, generators with nonlinear scrambler can be used as keystream generators in stream ciphers.

References

1. E. R. Berlekamp, Algebraic Coding Theory (Revised ed.), Aegean Park Press, 1984. J. L. Massey, “Shift-register synthesis and BCH decoding”, IEEE Trans. Information Theory, IT-15 (1): 122–127, 1969.
2. N. G. de Bruijn. A combinatorial problem. Indagationes Mathematicae, 8(1946), pp. 461-467.
3. C. G. Günther. Alternating step generator controlled by de Bruijn sequences, Advances in Cryptology – Eurocrypt’87, LNCS 304, pp. 5-14, 1988.
4. T. Johansson. Reduced complexity correlation attacks on two clock-controlled generators. Asiacrypt’98, LNCS 1514, pp. 342-356, 1998.
5. J. Golic, R. Menicocci, Edit Distance Correlation Attack on the Alternating Step Generator. Advances in Cryptology – Crypto’97, 1294 LNCS pp. 499-512, 1997.
6. J. Golic, R. Menicocci, Edit Probability Correlation Attacks on the Alternating Step Generator. Sequences and Their Applications - SETA, 1998.
7. J. Golic, R. Menicocci, Correlation analysis of the Alternating Step Generator. Design, Codes and Cryptography, 31, pp. 51-74, Kluwer Academic Publishers, 2004.
8. S. Al-Hinai, L. Batten, B. Colbert, and K. Wong, Algebraic Attacks on Clock-Controlled Stream Ciphers, LNCS 4058, pp 1-16, Springer, 2006.
9. S. Khazaei, S. Fisher, W. Meier, Reduced complexity attacks on the alternating step generator. Proceedings of SAC’07, Springer-Verlag, pp. 1-16, 2007.
10. S. Su, K. Chiu, L. Wuu. The Cryptanalysis of LFSR/FCSR based alternating step generator. ICCES. 2006.
11. M. M. Hassanzadeh, T. Helleseth. Algebraic attack on the alternating step(r,s) generator. Proceedings of the IEEE International Symposium on Information Theory, pp. 2493-2497, IEEE, 2010.
12. M. M. Hassanzadeh, T. Helleseth. Algebraic attack on the second class of modified alternating k-generators. NISK conference, 2010.
13. M. M. Hassanzadeh, T. Helleseth. Algebraic attack on the more generalized clock-controlled alternating step generators. Proceeding of SPCOM 2010, pp. 1-5, 2010.
14. A. A. Kanso. The alternating step(r,s) generator. SECI, Tunis, 2002.
15. A. A. Kanso. More generalized clock-controlled alternating step generator. Proc of ACNS’04, LNCS 3089, pp. 326-338, 2004.
16. A. A. Kanso. Modified clock-controlled alternating step generator. Computer Communications 32, Elsevier, pp. 787-799, 2009.
17. R. Białota, G. Kawa. Modified alternating k-generators. Design, Codes and Cryptography, 35, pp. 159-174, Kluwer Academic Publishers, 2005.
18. R. Wicik, T. Rachwalik. Modyfikacje generatora z naprzemiennym taktowaniem rejestrów. KSTiT, Gdańsk, Przegląd telekomunikacyjny nr 8-9/2013 s. 1225-1230, 2013.
19. R. Wicik, T. Rachwalik, Modified Alternating Step Generators, MCC, Saint-Malo, France, 2013. Cryptology ePrint Arch., 2013/728. eprint.iacr.org/2013/.
20. M. Borowski R. Wicik. How to speed up a stream cipher. RCMCIS 2002, Biuletyn WIŁ, Zegrze, 2003.
21. T. Rachwalik, J. Szmidt, R. Wicik, J. Zabłocki. Generation of nonlinear feedback shift registers with special purpose hardware. MCC, Gdańsk, 2012. Cryptology ePrint Archive, 2012/314. eprint.iacr.org/2012/
22. J. Szmidt, P. Dąbrowski, G. Łabuzek, T. Rachwalik. Searching for nonlinear feedback shift registers with parallel computing. MCC, Saint Malo, France 2013. Cryptology ePrint Arch., 2013/542. eprint.iacr.org/2013/
23. P. Dąbrowski, G. Łabuzek, T. Rachwalik, J. Szmidt, Searching for Nonlinear Feedback Shift Registers with Parallel Computing, Information Processing Letters 114 (2014) pp.268-272.
24. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. Handbook of applied cryptography. CRC Press, 1997.
25. R. Wicik, M. Borowski. Randomness testing of some random and pseudorandom sequences. Military Communication Conference, Prague, 2008.
26. M. Leśniewicz, Sprzętowa generacja losowych ciągów binarnych. WAT, Warszawa, 2011.