Modified Alternating Step Generators

Robert Wicik, Tomasz Rachwalik

Military Communication Institute
Warszawska 22A, 05-130 Zegrze, Poland
{r.wicik, t.rachwalik}@wil.waw.pl

Abstract. Irregular clocking of feedback shift registers is a popular technique to improve parameters of keystream generators in stream ciphers. Another technique is to implement nonlinear functions. We join these techniques and propose Modified Alternating Step Generators built with linear and nonlinear feedback shift registers. Adequate nonlinear Boolean functions are used as feedbacks or as filtering functions of shift registers in order to increase complexity of sequences produced by individual registers and the whole generator. We investigate basic parameters of proposed keystream generators, such as period, linear complexity and randomness.

Key words: stream ciphers, alternating step generator, feedback shift registers, period and linear complexity, randomness.

1 Introduction

The basic component of stream ciphers is a pseudorandom generator of a keystream. A sender combines the keystream with a plaintext during ciphering. Subsequently, a receiver combines the same keystream with a cryptogram to retrieve the plaintext. Parameters of keystreams are very important due to various methods of cryptanalysis. A generator of keystream in a stream cipher should produce pseudorandom sequences of large period, high linear complexity and very good statistical properties. It should be resistant to reconstruct the internal state and member functions (registers) from the output sequence.

One class of pseudorandom generators bases on feedback shift registers (FSR) with linear or nonlinear functions as their feedbacks. Such generators contain one or frequently more than one feedback shift registers regularly or irregularly clocked. Irregular clocking of one FSR controlled by the output of another FSR is a principle of stop-and-go generator [2]. Such generator (under suitable conditions) gives pseudorandom sequence of high linear complexity.

In the alternating step generator (ASG), the de Bruijn sequence controls the irregular clocking of two linear feedback shift registers [3]. In the alternating step(r,s) generator (ASG(r,s)), two integers r and s determine how many times one register or the other is clocked by one bit of the de Bruijn sequence [7]. These generators produce binary sequences with maximum period, high linear complexity and good statistical properties, but they are vulnerable to cryptanalytic attacks [6], [12], [25], [26].

Modified alternating generators (MAG) base on the ASG or ASG(r,s), but there are some modifications in controlling of clocking or in producing of an output [8] [9] [11]. The most interesting one is the third class of modified alternating k-generators – MAG_k^3, where the function of binary states of all three registers of the generator gives an output. We use this construction as the base of our proposals of modified alternating step generators – MASG – designed to enhance the resistance of the ASG to known attacks.

The basic technique for protection of the interior of the keystream generator is to introduce nonlinearity of its output. We described in [20] the concept of utilizing wide nonlinear transformation as an output of the ASG. In this paper, in the MASG and MASG_0 we use nonlinear Boolean functions as feedbacks of shift registers [22] [27]. In the MASG_1 and MASG_2 we use nonlinear Boolean functions [19] as filters and combiners of states of registers to produce output binary sequences of the generators. We evaluate periods and linear complexities of output keystreams of MASGs and asses their randomness.

2 The Alternating Step Generator

The alternating step generator (ASG) is a pseudorandom generator of binary sequences, where the concept of the stop-and-go generator was developed [3]. The ASG consists of two linear feedback shift registers (LFSR) alternately clocked by the de Bruijn sequence [4]. 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 produce output bits from the generator.

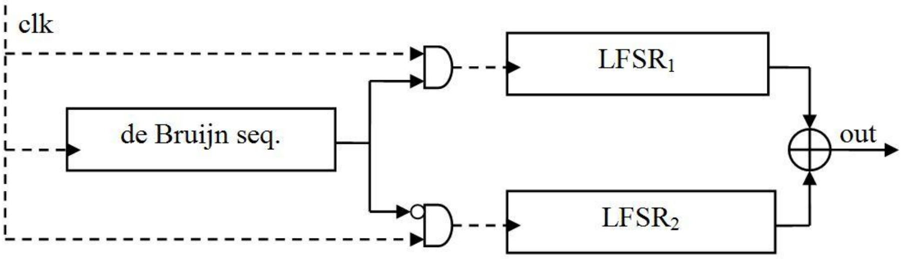

Figure 1. The Alternating Step Generator

The output sequence from the ASG has large period T and high linear complexity L:

$$T= M_1 M_2 2^k \quad (1)$$

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

where:

- the period of de Bruijn sequence is $K=2^k$,
- different and irreducible polynomials defining feedbacks of LFSRs of degrees m_1 and m_2 have periods: $M_1,M_2> 1$; $\gcd(M_1, M_2) = 1$ (in a binary case it is enough to feedback polynomials be relatively prime).

We can observe growth of the linear complexity of the output sequence from the ASG (2) 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 [6] and [25]. Asymptotic time complexity of these attacks is $O(m^2 2^{2m/3})$ and data complexity $O(2^{2m/3})$, where m is the length of the shortest register in the ASG. Time complexity of the algebraic attack described in [13] 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 irregularly controlled registers.

2.1 *The Alternating Step(r,s) Generator*

The alternating step(r,s) generator, ASG(r,s), was proposed in [7]. Two positive integers: r and s are parameters of this generator, where r determines how many times one register (LFSR$_1$) is clocked and s – the other one (LFSR$_2$). The original construction of ASG is a special case of ASG(r,s) for r=s=1, where LFSRs are alternately clocked only ones at a given time.

The controlling sequence in ASG(r,s) is the de Bruijn sequence, where there are 2^{k-1} ‘zeros’ and ‘ones’, thus after full period K=2^k, the LFSR$_1$ is clocked $r2^{k-1}$ times and the LFSR$_2$ is clocked $s2^{k-1}$ times. The output sequence of ASG(r,s) has the same period T (1) and linear complexity L (2) as the original ASG under extra conditions:

$$\gcd(r, M_1) = 1, \quad \gcd(s, M_2) = 1.$$

The aim of the introduction of two integers r and s to the ASG was to increase its resistance to correlation attacks. But in [12] authors showed, that the ASG(r,s) is as secure as the original ASG. Afterwards, Kanso proposed in [8] and [9] 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 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 XOR sum for addition with carry (ADD) as an output function. This modification of the ASG does not improve its security substantially.

3 Modified Alternating *k*-Generators

Modified alternating k-generators (MAG$_k$) were proposed in [11]. In the first approach, the main change introduced to the ASG was in the output function – sequence is produced by the XOR sum of binary sequences from all three registers of the MAG$_k$, as it is presented in Fig. 2. Feedback shift registers in the MAG$_k$ can be linear on nonlinear.

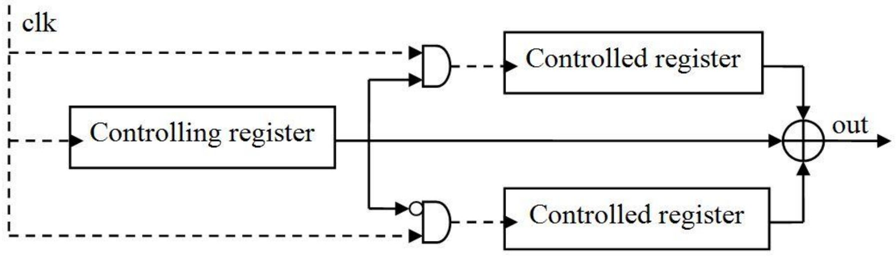

Figure 2. The Modified Alternating k-Generator

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;
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;
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 ones.

The first class of MAG$_k$ is similar to the MCCASG [9]. The second class was analyzed in [13], where authors showed that its security is not better than the security of the original alternating step generator. The third class is similar to our concept described in [20], where the output transformation combines bit sequences from all registers of the ASG. The MAG$_k^3$ produces one bit at a given time and the generator from [20] produces the wide vector of bits.

We concentrated on the third class of modified alternating *k*-generators, the MAG$_k^3$, because we suppose that properly selected nonlinear output function can avoid attacks on the generator. But, first we replaced two controlled linear feedback shift registers with nonlinear feedback shift registers (NLFSR).

4 NonLinear Feedback Shift Registers

Linear feedback shift registers (LFSR) are popular building blocks in stream ciphers, but LFSRs have a drawback, as their linear complexity is equal to their length. In recent years, nonlinear feedback shift registers (NLFSR) have received much attention in designing numerous cryptographic algorithms. In most cases, NLFSRs have much bigger linear complexity than LFSRs of the same length.

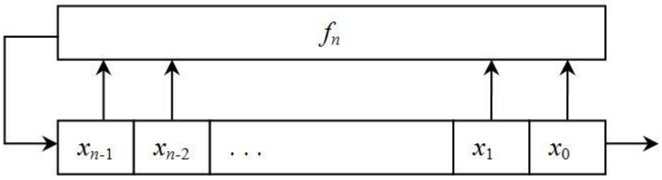

Figure 3. A Feedback Shift Register

The scheme of a (Fibonacci, binary) feedback shift register (FSR) is presented in Fig. 3. A shift register is composed of *n* cells indicated by:

$$x_0, x_1, \ldots, x_{n\text{-}1}, \text{ where } x_i \in \{0,1\} \text{ for } i= 0, 1, ..., n–1.$$

If a Boolean feedback function $f_n : \{0,1\}^n \text{ –> } \{0,1\}$ is linear, we call such register the LFSR, if nonlinear – the NLFSR. There are well known methods of constructing linear feedback functions for LFSRs such that generated sequences are maximal with period 2^n-1. Methods for constructing nonlinear functions for feedback shift registers generating maximal period sequences are developed.

We used the implementation of NLFSRs in Field Programmable Gate Arrays [22] to perform a search of NLFSRs with the order up to n=31 and the maximum period equal to $2^n−1$. Another implementation and parallel computing gave similar results [27]. Maximal linear complexity of sequences taken from such NLFSRs is $2^n−2$. They worked out many nonlinear functions *f* as feedbacks of shift registers, for instance: (3) and (4). These NLFSRs have the maximum period and the linear complexity close to the period.

The algebraic normal form of the function f_{27}: (3)

$$\begin{aligned}&x_0 + x_4 + x_8 + x_9 + x_{11} + x_{12} + x_{15} + x_{16} + x_{23} +\\&x_{12}x_{22} + x_{13}x_{23} + x_{13}x_{25} + x_{22}x_{23} +\\&x_7x_8x_{24} +\\&x_{12}x_{14}x_{26} + x_6x_{11}x_{19}x_{22}\end{aligned}$$

The algebraic normal form of the function f_{29}: (4)

$$\begin{aligned}&x_0 + x_2 + x_3 + x_4 + x_7 + x_{12} + x_{20} + x_{21} +\\&x_2x_{25} + x_{15}x_{21} + x_{17}x_{21} +\\&x_5x_{17}x_{22} + x_5x_{23}x_{26} + x_{11}x_{14}x_{15} +\\&x_{11}x_{24}x_{26} + x_{14}x_{21}x_{25} +\\&x_7x_{10}x_{20}x_{24} + x_{11}x_{13}x_{17}x_{21}\end{aligned}$$

Addition and multiplication are performed in (3) and (4) modulo 2.

5 Nonlinear filtering, combining functions

Nonlinear Boolean functions are often used in stream ciphers together with linear feedback shift registers in order to increase security of keystream generators. There are two kinds of such functions: filtering and combining ones. The combining function is an output function, which is fed by several LFSRs. The filtering function is an output function, which is fed by the state of one LFSR. Various properties of such functions are critical for ensuring the security of keystream generators. Such functions should:

- be balanced,

- have high nonlinearity in the meaning of the Hamming distance to affine functions and to functions with linear structure [24],
- have high algebraic degree (nonlinear order) and many nonlinear components in its algebraic normal form,
- have correlation immunity of high order (note, that high algebraic degree restrict the maximum possible correlation immunity).

Perfect nonlinear functions (e.g. bent functions), have maximum nonlinearity, but are not balanced. Fortunately, it is easy to balance such functions not much reducing nonlinearity. Correlation immunity for high nonlinearity is low. However, we can increase correlation immunity by adding memory to the function.

Functions proposed in [19] for S-boxes, based on the Rothaus' construction, can be utilized as nonlinear filters and combiners of feedback shift registers in keystream generators. After balancing, a nonlinear Boolean bent function has high nonlinearity and many nonlinear components in its algebraic normal form.

For even q, a Boolean bent function $g : \{0,1\}^q \rightarrow \{0,1\}$ has:

- nonlinearity (Hamming distance do the nearest affine function): $2^{q-1}-2^{q/2-1}$,
- nonlinear order < $q/2$.

Boolean balanced function $gb : \{0,1\}^q \rightarrow \{0,1\}$, derived from a bent function g, has:

- nonlinearity > $2^{q-1}-2^{q/2}$,
- nonlinear order: q-1.

There are in [19] described methods of generating and constructing bent functions for q= 4, 6, 8, and for q>8. In order to balance bent function for even q we should negate $2^{q/2-1}$ bits in its truth table – 'zeros' if its Hamming weight is $2^{q-1}-2^{q/2-1}$ or 'ones' if its Hamming weight is $2^{q-1}+2^{q/2-1}$. We can observe that these balanced functions have higher nonlinear order than bent functions.

We generate three functions gb_0, gb_1 and gb_2 for q=8. Below we present the truth tables of these functions – sets of output bits written in hexadecimal of eight bits for 256 consecutive inputs from `00` to `FF`.

gb_0 = {`7F,1C,0B,76,9E,88,8D,28,ED,FB,85,2C,A0,CA,8E,31,`
`B2,C2,7A,55,26,22,88,1F,DE,58,0B,B4,C7,9F,7C,D9`}
gb_1 = {`64,F2,43,FD,37,11,E9,26,18,3F,BD,2B,C6,92,9A,3F,`
`35,40,D4,D2,85,88,8D,62,B6,73,D5,7B,9B,F4,01,85`}
gb_2 = {`3A,9E,22,FB,F3,B9,AC,64,60,6B,66,45,0C,6A,4D,BC,`
`BB,87,5B,7C,02,41,AF,0A,16,8D,E0,B5,52,2D,BB,2D`}

Nonlinearity of gb_0, gb_1 and gb_2 is 114, nonlinear order is 7. The Table 1 contains numbers of nonlinear components in algebraic normal forms of these functions.

Table 1. Numbers of nonlinear components in gb for q=8

degree	**2**	**3**	**4**	**5**	**6**	**7**
$\boldsymbol{gb_0}$	16	36	49	40	15	5
$\boldsymbol{gb_1}$	17	27	32	27	17	3
$\boldsymbol{gb_2}$	17	27	22	15	14	5

We generate three functions gb_{10}, gb_{11} and gb_{12} for q=18, too. The Table 2 contains numbers of nonlinear components in algebraic normal forms of these functions.

Table 2. Numbers of nonlinear components in gb for q=18

degree	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
$\boldsymbol{gb_{10}}$	80	356	1185	2863	5454	9091	14173	19261	20460	16064	9245	4363	1505	382	77	9
$\boldsymbol{gb_{11}}$	82	370	1176	2692	4229	5444	7568	13542	18343	15588	9292	4224	1539	405	75	9
$\boldsymbol{gb_{12}}$	83	363	1187	2583	4563	6722	9973	15171	18500	15679	9361	4314	1488	398	86	10

There are too many nonlinear components to implement a function gb for q=18 in a reasonable hardware, but a truth table of such function needs only 32kByte of memory.

6 Modified Alternating Step Generators

In the section 4, we proposed binary nonlinear feedback shift registers with maximum period and high linear complexity. In the section 5, we introduced nonlinear, balanced filtering/combining Boolean functions with high nonlinearity. In this section, we propose modifications of the alternating step generator. These modifications use overall concepts of the alternating step generator and modified alternating k-generators (ASG/MAG$_k$).

6.1 MASG and MASG$_0$

Our first approach to the modification of the ASG is to replace controlled registers LFSR$_1$ and LFSR$_2$ by NLFSRs with feedback functions (3) and (4) in order to introduce nonlinearity to the ASG. Upon (1) and (2), for m_1=29, m_2=27 and M_1=2^{29}-1, M_2=2^{27}-1, and L_1=2^{29}-2, L_2=2^{27}-2, the output sequence of the modified alternating step generator has the period *T'* and should have the linear complexity no less than *L'*:

$$T' = M_1 M_2 2^k \quad (5)$$

$$L' > (L_1+L_2)2^{k-1} \quad (6)$$

where:

- the period of de Bruijn sequence is K=2^k (suggested length of the controlling register is $k \geq 127$);
- gcd(M_1,M_2)=1.

Note, that L_1=2^{29}-2 and L_2=2^{27}-2 are maximal for these NLFSRs, but we know how to check these values for NLFSRs of order up to 25 in this moment. The linear complexity of the NLFSR shows as how it is secure against the Berlekamp-Massey synthesis algorithm – it is the length of the shortest LFSR generating the same sequence, hence we replaced m_1 and m_2 from (2) with L_1 and L_2 in (6).

The modified alternating step generator produces binary sequences with better linear complexity and seems more secure than the ASG because of nonlinear registers, but we should find nonlinear functions for longer NLFSRs than 31. These functions should have high nonlinear order, many nonlinear components and give maximal sequence.

We will call the Modified Alternating Step Generator (MASG) the alternating step generator, where the output is produced by the XOR sum of binary sequences from two NLFSRs alternately clocked by the de Bruijn sequence. De Bruijn sequence is obtained from LFSR by adding zero-bit after *k*-1 zeros.

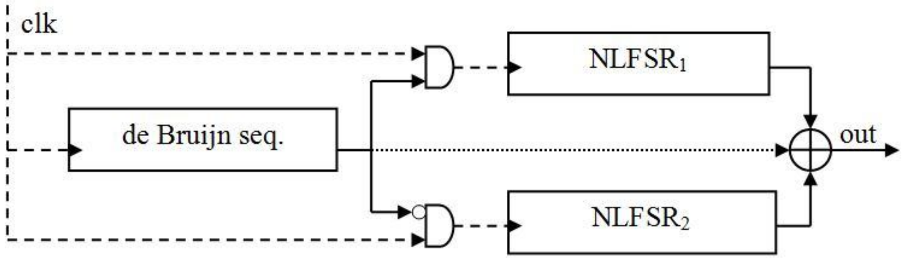

Figure 4. The Modified Alternating Step Generators: MASG and MASG$_0$

We will call the MASG$_0$ the alternating step generator, where the output is produced by the XOR sum of binary sequences from all three registers. In other words, MASG$_0$ is the MAG$_k$ with LFSR as the controlling register and NLFSRs as controlled ones. The dotted line in Fig. 4 represents the difference between the MASG and the MASG$_0$.

6.2 MASG$_1$

MASG$_1$ – is the MAG$_k$, where all linear feedback shift registers are equipped with filtering functions of the class *gb*, described in the section 5 for *q*=8. Controlling register LFSR$_0$ has parameters (*length*, *period*): (*k*, *K*); controlled registers LFSR$_1$ and LFSR$_2$: (m_1, M_1) and (m_2, M_2), respectively.

The output of the MASG$_1$ is the XOR sum of outputs of three functions: gb_0, gb_1 and gb_2 (as we showed in Fig. 5). Inputs to these functions are taken from shift registers. For each register and associated function, distances between cells in registers (indicated by *x* in Fig. 3) taken to the functions are fixed and equal to δ_0, δ_1 and δ_2, respectively for controlling and controlled registers.

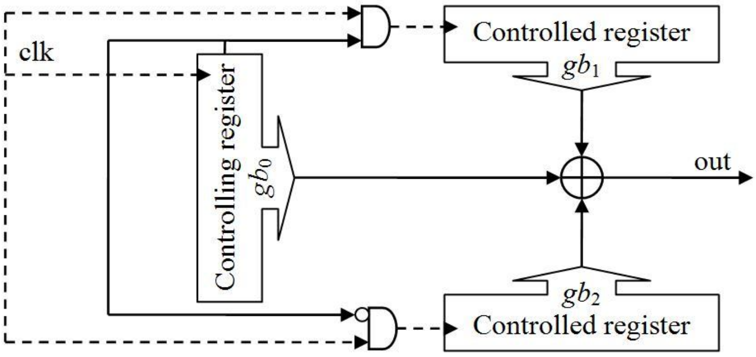

Figure 5. The Modified Alternating Step Generator MASG$_1$

If registers in the MASG$_1$ are linear with feedbacks defined by primitive polynomials with periods relatively prime to distances δ_0, δ_1 and δ_2:

$$\gcd(K, \delta_0),\ \gcd(M_1, \delta_1) \text{ and } \gcd(M_2, \delta_2) \tag{7}$$

and functions *gb* have nonlinear orders:

$$q_0\text{-}1< k, \quad q_1\text{-}1< m_1 \quad \text{and} \quad q_2\text{-}1< m_2$$

so independently treated outputs sequences of regularly clocked registers LFSR$_{0,1,2}$ with filters $gb_{0,1,2}$ have (see [23]) linear complexities no less than:

$$L_{0,1,2} \geq \begin{pmatrix} k \\ q_{0,1,2}-1 \end{pmatrix} - (N_{0,1,2}-1) \tag{8}$$

respectively, where $N_{0,1,2}$ denote numbers of components of maximal nonlinear order in $gb_{0,1,2}$ functions (see column for degree 7 in the Table 1).

Thus, the period and the linear complexity of the output sequence of the MASG$_1$ can be derived from (5) and (6). Recall that we assumed q_0, q_1, q_2 = 8; suggested LFSRs lengths ~128 (for example: k=127, m_1=131 and m_2=137). Characteristic polynomials should define many connections (>10) to feedbacks of shift registers.

6.3 MASG$_2$

MASG$_2$ – is the MAG$_k^3$ with the output function of the class *gb* – in other words – it is the MASG$_1$ with one large nonlinear function for q=18. Overall scheme of the MASG$_2$ is presented in Fig. 5. Selected states bits from each three linear feedback shift registers feed this function. For each register, distances between cells in the registers (denoted by x in Fig. 3) taken to the output function are fixed and equal to δ_0' for controlling and to δ_1' and δ_2' for controlled registers. These distances fulfill requirements from (7), too.

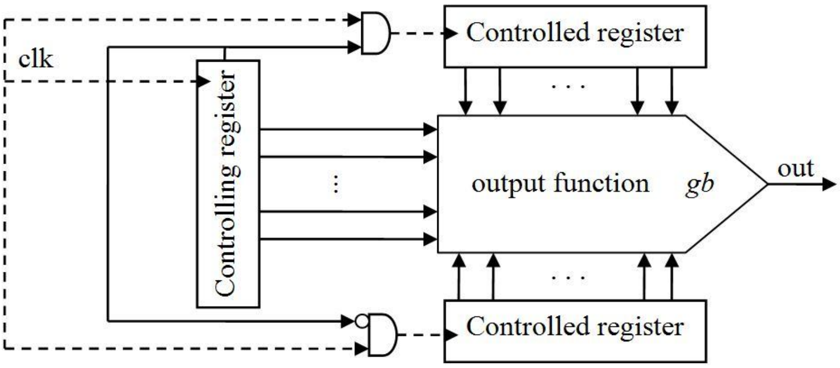

Figure 6. The Modified Alternating Step Generator MASG$_2$

The period and the linear complexity of the output sequence can be derived from (5), (6) and (8) and will depend on the algebraic normal form of *gb* and numbers of state's bits feeding parts of this function. For q = 18 nonlinear order of the output function of class *gb* is 17, but parts dedicated to each register are lower. Therefore we should plan connections between functions and registers so that nonlinear order of each part is not smaller than 6 in this case.

7 Randomness properties

We checked experimentally randomness properties of binary sequences produced by the MASGs described in the previous section and produced by the original ASG and modified ASG i.e. MAG$_k$. We tested the randomness using seven basic statistical tests [17], [21]:

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

The tests use as reference distributions the chi-square distributions and the standard normal distribution. Observed frequencies of events are compared with their expected frequencies. We split calculated statistics into 8 classes according to the range of significance level. The class A identifies a group of the best statistics and the class H identifies the worst case in terms of randomness, but all cases are possible with suitable probabilities as it is shown in the Table 3. For popular level of significance α=0.05, sequences passed tests if their statistics are a class A, B or C.

Table 3. Expected percentages of appearances of classes

Classes	Exp. %	α
A + B + C	95	$(1-\alpha)<0.95$
A	80	$(1-\alpha)<0.80$
B	10	$0.8\le(1-\alpha)<0.9$
C	5	$0.9\le(1-\alpha)<0.95$
D	2.5	$0.95\le(1-\alpha)<0.975$
E	1.5	$0.975\le(1-\alpha)<0.99$
F	0.5	$0.99\le(1-\alpha)<0.995$
G	0.4	$0.995\le(1-\alpha)<0.999$
H	0.1	$0.999\le(1-\alpha)$

We tested 1GByte sequences produced by the ASG, MAG$_k$ and MASGs starting from randomly selected initial states. Obtained results of experiments for overall sequences are given in the Table 4. The Table 5 contains percentages of classes of statistics for 1MB subsequences of examined sequences.

Table 4. Classes of statistics for 1GB sequences

Test	1	2	3	4	5	6	7
ASG	A	A	A	A	A	B	B A A A A A A A
MAG$_k$	A	A	A	A	B	A	A A A A B B B A
MASG	A	A	A	A	A	A	A A B A A A A A
MASG$_0$	A	A	A	B	A	A	A A C A B A A A
MASG$_1$	A	A	A	A	A	A	A A A A A A A A
MASG$_2$	A	H	H	H	H	F	H H H G A A C A

Table 5. Percentages of classes for 1MB subsequences

Class	ABC	A	B	C	D	E	F	G	H
ASG	95.03	79.24	10.76	5.04	2.66	1.37	0.43	0.42	0.08
MAG$_k$	95.31	80.13	10.63	4.55	2.32	1.53	0.49	0.17	0.18
MASG	94.91	79.73	10.32	4.85	2.47	1.55	0.59	0.39	0.10
MASG$_0$	94.98	79.14	10.73	5.11	2.72	1.62	0.36	0.25	0.07
MASG$_1$	94.46	79.16	10.25	5.05	2.61	1.72	0.60	0.54	0.07
MASG$_2$	93.40	75.91	11.44	6.05	3.39	2.01	0.57	0.49	0.14

Results for the ASG, MAG$_k$, MASG, MASG$_0$ and MASG$_1$ are as we expected for random sequences. (Some differences may result from the low precision calculations of statistics and percentiles of chi-square distributions). For MASG$_2$ we observed worse statistics.

Having regard to the results given above, we prefer MASG$_1$ as easier to implement and to choose proper nonlinear functions and also its connections with registers. MASG and MASG$_0$ give sequences with good statistics, but need longer NLFSRs. Sequences produced by MASG$_2$ (or MAG$_k^3$ proposed in [11]) with inappropriate output function may be distinguishable from random sequences.

8 Summary

We proposed in this paper four modified alternating step generators MASG, MASG$_0$, MASG$_1$ and MASG$_2$. All generators base on the alternating step generator introduced by C. G. Günther and its further modifications – especially on the modified alternating *k*-generators proposed by Białota and Kawa, where a function of output bits from all three registers (linear and nonlinear) generates the output bit stream.

In the MASG and MASG$_0$ we replaced controlled registers by nonlinear feedback shift registers. We applied nonlinear functions proposed in [22] and [27] as feedbacks in order to obtain output sequences with maximal period and the linear complexity close to the period. It might be a good design, if applying longer, maximal NLFSRs.

In the MASG$_1$ we added nonlinear filtering functions to all linear feedback shift registers. In the MASG$_2$ nonlinear filtering/combining function replaced the XOR sum in the output of the generator. We proposed methods for constructing such functions in order to achieve balanced Boolean functions with high nonlinearity and many nonlinear components in algebraic normal form. Correlation immunity needs to add bits of memory to these functions or to the output function of the generator.

The main aim of proposed modifications was to increase security of a stream cipher, where the ASG plays the role of the keystream generator. We have checked basic parameters of sequences produced by the MASGs: the period, the linear complexity and randomness. Properly selected registers and nonlinear functions in MASG, MASG$_0$ and MASG$_1$ give binary sequences with the maximum period, high linear complexity and good randomness properties and should enhance the resistance of the alternating step generator to the cryptanalysis. (Note that, nonlinear feedback shift registers in MASG and MASG$_0$ should be longer, than proposed). For MASG$_2$

with one big nonlinear function in the output we obtained bad results of randomness tests. MASG$_2$ is the example realization of the third class of the modified alternating *k*-generator [11].

We conclude that among the MASGs the MASG$_1$ is the easiest to implement and to choose proper nonlinear functions. Nonlinear filtering functions should significantly difficult reconstruction of linear feedback shift registers from the output sequence. Additionally, it would be desirable to replace XOR function with a nonlinear combining function or finite state machine in the output of the MASG$_1$.

Further research will focus on the security parameters of the modified alternating step generators – how much nonlinear functions, suggested in this paper, increase the resistance of the alternating step generator to known attacks.

References

[1] S. W. Golomb. Shift Register Sequences. San Francisco, Holden-Day, 1967, revised edition, Laguna Hills, CA, Aegean Park Press, 1982.

[2] T. Beth, F. Pipper. The stop-and-go-generator. Advances in Cryptology – Eurocrypt'84, LNCS 209, pp. 88-92, 1985.

[3] C. G. Günther. Alternating step generators controlled by de Bruijn sequences, Advances in Cryptology – Eurocrypt'87, LNCS 304, pp. 5-14, 1988.

[4] N. G. de Bruijn. A combinatorial problem. Indag. Math., 8(1946), pp. 461-467.

[5] S. A. Tretter. Properties of PN2 sequences. IEEE Trans. on Information Theory, vol. IT-20, pp. 295-297, 1974.

[6] T. Johansson. Reduced complexity correlation attacks on two clock-controlled generators. In proceedings of Asiacrypt, pp. 342-356, 1998.

[7] A. A. Kanso. The alternating step(r,s) generator. SECI, Tunis, 2002.

[8] A. A. Kanso. More generalized clock-controlled alternating step generator. Proc of ACNS'04, LNCS 3089, pp. 326-338, 2004.

[9] A. A. Kanso. Modified clock-controlled alternating step generator. Computer Communications 32, Elsevier, pp. 787-799, 2009.

[10] S. Su, K. Chiu, L. Wuu. The Cryptanalysis of LFSR/FCSR based alternating step generator. ICCES. 2006.

[11] R. Białota, G. Kawa. Modified alternating *k*-generators. Design, Codes and Cryptography, 35, pp. 159-174, Kluwer Academic Publishers, 2005.

[12] 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.

[13] M. M. Hassanzadeh, T. Helleseth. Algebraic attack on the second class of modified alternating *k*-generators. NISK conference, 2010.

[14] M. M. Hassanzadeh, T. Helleseth. Algebraic attack on the more generalized clock-controlled alternating step generators. Proceeding of SPCOM 2010, pp. 1-5, 2010.

[15] S. W. Golomb, G. Gong. Signal Design for Good Correlation. For Wireless Communication, Cryptography, and Radar. Cambridge University Press, 2005.

[16] K. Mandal, G. Gong. Probabilistic generation of good span n sequences from nonlinear feedback shift registers. University of Waterloo, preprint, 2012.

[17] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. Handbook of applied cryptography. CRC Press, 1997.

[18] M. S. Turan. On the nonlinearity properties of maximum-length NFSR feedbacks. Cryptology ePrint Archive, 2012/112. eprint.iacr.org/2012/

[19] R. Wicik. Utilization of Boolean bent functions for the construction of large S-boxes. RCMCIS. Zegrze, 1999.

[20] M. Borowski R. Wicik. How to speed up a stream cipher. RCMCIS 2002, Biuletyn WIŁ, Zegrze, 2003.

[21] R. Wicik, M. Borowski. Randomness testing of some random and pseudorandom sequences. Military Communication Conference, Prague, 2008.

[22] T. Rachwalik, J. Szmidt, R. Wicik, J. Zabłocki. Generation of nonlinear feedback shift registers with special purpose hardware. Military Communication Conference, Gdańsk, 2012. Cryptology ePrint Archive, 2012/314. eprint.iacr.org/2012/

[23] R. A. Rueppel. Analysis and Design of Stream Ciphers. Springer-Verlag, Berlin, Heidelberg, 1986.

[24] W. Meier, O. Staffelbach, Nonlinearity criteria for cryptographic functions. Advances in Cryptology - Eurocrypt'89, LNCS Vol. 434. Springer-Verlag, Berlin, Heidelberg, pp. 549-562, 1990.

[25] S. Khazaei, S. Fisher, W. Meier, Reduced complexity attacks on the alternating step generator. Proceedings of SAC'07, Springer-Verlag, Berlin, Heidelberg, pp. 1-16, 2007.

[26] J. Golic, R. Menicocci, Correlation analysis of the Alternating Step Generator. Design, Codes and Cryptography, 31, pp. 51-74, Kluwer Academic Publishers, 2004.

[27] J. Szmidt, P. Dąbrowski, G. Łabusek, T. Rachwalik. The generation of nonlinear feedback shift registers with parallel computing. MCC, Saint Malo, France 2013. Cryptology ePrint Archive, 2013/542. eprint.iacr.org/2013/.