Journal of Information Security, 2014, 5, 218-227
Published Online October 2014 in SciRes. http://www.scirp.org/journal/jis
http://dx.doi.org/10.4236/jis.2014.54020

Scientific Research

Method of Designing Generators of Pseudorandom Sequences for Information Protection Based on Shift Register with Non-Linear Feedback Function

Saleh Al-Omar

Department of Engineering, Al-Ahliyya Amman University, Amman, Jordan
Email: salehalomar@yahoo.com

Received 18 July 2014; revised 25 August 2014; accepted 20 September 2014

Abstract

This paper proposes an efficient, high-tech method of construction of pseudorandom binary sequences generators with a repetition period 2^n for n-bit shift register with a nonlinear feedback function. The developed method is illustrated by constructing a nonlinear function feedback shift register. It is proved that the offered method requires the realization of a memory size proportional to n^2 that allows making successful use of suitable generators for practical use on the shift register of the longer word.

Keywords

Pseudorandom Sequences, Non-Linear Feedback Function, Information Protection

1. Introduction

In tasks of information protection, statistical monitoring and diagnosis, modeling and designing, generators of pseudorandom sequences are widely used. The effectiveness of these generators in informational technologies is defined by specific features of using them [1].

Pseudorandom objects are very important in modern systems of information protection [2]. Especially important role in this dynamically developing domain is played by pseudorandom binary sequences, which are used as a baseline for one of three types of information security algorithms—flow algorithms [3]. Pseudorandom sequences are widely used for generation of keys for the symmetrical data protection algorithms and pseudorandom binary strings for protocols of authentication of remote users in integrated systems [4]. In modern condi-

How to cite this paper: Al-Omar, S. (2014) Method of Designing Generators of Pseudorandom Sequences for Information Protection Based on Shift Register with Non-Linear Feedback Function. *Journal of Information Security*, **5**, 218-227.
http://dx.doi.org/10.4236/jis.2014.54020

tions of productivity growth of computational systems and possibilities to unite a huge number of computers in a network for solving problems of security breach, the problem of adequate increasing the reliability of information security becomes more actual, including methods of improving means of obtaining and using pseudorandom sequences.

Protective features of pseudorandom sequences and functional transformations in theoretical plan are defined by the principal impossibility of analytical solutions of the systems of non-linear Boolean equations [5]. This feature of Boolean transformations lies in the heart of using binary sequences and Boolean functions in information security systems. That is why the level of protection of data with using pseudorandom sequences is directly dependent on the nonlinearity of Boolean functions, which are used to generate pseudorandom sequences. From this, important reserve for increasing effectiveness of means of information protection, which use pseudorandom sequences, is in improving means of formation such sequences in the direction of increasing the nonlinearity of Boolean transformations, which they use.

Thus, at the current stage of development of information protection, the problem of developing methods and approaches for increasing effectiveness of hardware-software means of generation of pseudorandom binary sequences is actual.

2. Analysis of the Current State of the Problem of Effective Generation of Pseudorandom Sequences for Information Security Problems

2.1. Modern Generators of Pseudorandom Sequences Oriented to Be Used in Systems of Information Protection Are Built on the General Principles [5]

- Divergence—the notion which came from the theory of dynamical systems and consisted in the fact that algorithm of a generator should provide diverging sequence of binary n-bit words, that is the sequence, the repetitive cycle of which tends to 2^n [5].
- Nonlinearity of used functions of transformations, which provides complexity of recognition of generation functions of pseudorandom sequences.
- Computing complexity which means computational incompressibility of the procedures for pseudorandom sequences generation.

Mentioned principles, formulated on the theoretical level, are not strictly defined and partly overlap each other [6]. The notion of divergence is often used if the base for the functional transformation, which lies in the base of the generator, is a special case of the theory of numbers [6]. For bit transformations the problem of divergence which means to provide the repetitive cycle of $2^n - 1$, theoretically can be solved only for linear feedback functions: they must be isomorphic irreducible polynomials in Galois fields. For nonlinear functions of the bit transformation, divergence problem has a solution only for special cases [7].

2.2. A Criteria of Quality of the Generators of Pseudorandom Sequences Oriented to Be Used in Systems of Information Protection the Following Indicators Are Mostly Used [8]

- Statistical criteria which allows to estimate the probability characteristics of the generated sequence [6];
- Divergence criteria of generator work—is estimated by the minimal length of cycle appeared during the run;
- Unpredictability of the generator—is estimated by the complexity of the special algorithm-recognizer which allows to distinguish the generated sequence from random [7];
- Rate of sequence generation.

Really important when these criteria are considered is the principle, formulated in [8] and postulating that the criterion for determining the randomness is always determined by the class of the problems in which this randomness will be used, that is: use should determine the target probability distribution with the specified degree of approximation and the importance of each of the above criteria.

The third of mentioned earlier criteria of quality of pseudorandom sequences generators is their algorithmic unpredictability, which stipulates the complexity of the algorithm which can predict next values by using data from previous samples [9]. From the point of reliability of data protection it is the third criterion is a determining, because all breaches in security provided by using pseudorandom sequences are reduced to the extrapolation of the recent [9]. It is proved that irreversible transformations should be used to make the sequence meet the un-

predictability demand. For flow algorithms widely used in data transfers across networks and telecommunication systems, high speed of generation of the sequence is very important [3]. Exclusion of the possibility of gaining access to programs which implement the generation process is also very important. Based on the last two demands, more preferable is a variant of hardware realization of the generators. In this case, significant criterion of quality of means of pseudorandom sequences generation is their complexity. In practice, the most important class of generators of pseudorandom sequences is the generators based on shift registers with linear feedback—LFSR (Linear Feedback Shift Register). The main advantage of such generators is the fact that due to the developed method of synthesis of the feedback functions their acyclic work is provided (it means the cycle is $2^n - 1$ with n-bit shift register) along with the simplicity of circuit implementation on a hardware level and high performance [8]. So far as the maximum number of different combinations of n binary digits is 2^n, period of the generated sequence cannot exceed 2^n. If LFSR gets into a state with zero values of all bits, it cannot go to another state. That is why zero state shouldn't appear if the initial state nonzero. Therefore, the maximum number of possible states is $2^n - 1$. Maximal length sequence will be provided in case when the feedback function is a simple polynomial in Galois fields [4]. Thus the number $G(n)$ of simple polynomials in Galois fields, and therefore the quantity of sequences with length of $2^n - 1$ formulated by LSFR is determined by the formula:

$$N_L(n) = \frac{\phi\left(2^n - 1\right)}{n} \tag{1}$$

where $\phi(2^n - 1)$—Euler number—quantity of integer, including ones, which are less than $2^n - 1$ and those, which don't have common divisors with $2^n - 1$.

Theoretically the maximum possible number of states of n-bit shift register equals 2^n. Respectively, maximum repetition period of the binary sequence formulated by a nonlinear shift register of such length is 2^n too. Such sequences are called Bruijn sequences. Number $N(n)$ of nonlinear feedback functions of n-bit shift register which provide the maximum repetition period is determined by the formula:

$$N(n) = 2^{2^{n-1}-n} \tag{2}$$

For example, for the 6-bit shift register ($n = 6$) according to (1) exist only 6 simple polynomials in Galois fields and respectively 6 linear functions which provide period of $2^6 - 1 = 63$ which is close to the maximum, though that number of nonlinear feedback functions for the register with the same bit number which provide the maximum period—2^6 according to (2) is $2^{26} = 67108864$.

Despite such a big number of nonlinear feedback functions, providing the maximum period of repetition of the sequence, finding them is a challenge, which hasn't acceptable solution to date [4]. So, for $n = 6$ assuming that the total number of functions is 2^{64}, the maximum period is provided, on average, only by one of $2^{64-26} = 2^{40}$ functions.

3. Method of Finding Nonlinear Feedback Functions

The state of a shift register is characterized by w code which corresponds to the binary vector of X_w values of register bits:

$$X_w = \left\{x_0^w, x_1^w, \cdots, x_{n-2}^w, x_{n-1}^w\right\}$$

During the register shift new value of v code is determined by the following method:

$$X_v = \left\{x_1^w, x_2^w, \cdots, x_{n-2}^w, f\left(X_w\right)\right\}, \quad v = \left(2 \cdot w\right) \bmod 2^n$$

Suppose $f(x_1, x_2, \ldots, x_n)$—Boolean feedback function of n-bit shift register for which the following condition is true:

$$f\left(x_0, x_1, \cdots, x_{n-1}\right) = 1 \oplus f\left(x_0 \oplus 1, x_1, \cdots, x_{n-1}\right) \tag{3}$$

If the feedback function $f\left(x_1, x_2, \cdots, x_n\right)$ satisfies the condition (3), then each v code in the register is preceded by only one w code.

A set of codes which sequentially formulated in shift register with feedback feature which satisfies the condition [4] we will name the ring. Each of 2^n possible codes in n-bit register is included into one of the rings.

Suppose we have two rings—A and B. If code w is included in A and the symmetrical to it code $v = \left(w + 2^{n-1}\right) \bmod 2^n$ is included in B, then inverting the value of the function in these codes will lead to unite of A and B rings.

Proof: Binary vector $X_w = \left\{x_0^w, x_1^w, \cdots, x_{n-2}^w, x_{n-1}^w\right\}$ of state of the register corresponds to w code. Code e follows after w in the A ring: $X_e = \left\{x_1^w, x_2^w, \cdots, x_{n-2}^w, f\left(X_w\right)\right\}$. When inverting the value of the feedback function on code w, the following after w will be code u: $X_u = \left\{x_1^w, x_2^w, \cdots, x_{n-2}^w, f\left(X_w\right) \oplus 1\right\}$. Code v is symmetrical to w: $X_v = \left\{x_0^w \oplus 1, x_1^w, \cdots, x_{n-2}^w, x_{n-1}^w\right\}$. According to (3.1) $f\left(X_v\right) = 1 \oplus f\left(X_w\right)$, that is why code u follows after v in the B ring. When inverting functions on code v, the following code will be e: $X_e = \left\{x_1^w, x_2^w, \cdots, x_{n-2}^w, f\left(X_w\right) \oplus 1\right\}$.

Therefore, when inverting functions on symmetrical codes which are included into different rings A and B, after w code there will be transition to code $u \in$ B. Later, serial pass of all codes of the ring B is performed, last of which is code v. From this code transition to code $e \in$ A is performed. Further all codes of the ring A are passed serially. Thus, rings A and B are combined into one ring.

Offered method of synthesis of the nonlinear feedback function which provides full period is based on basic procedure of combining rings, obtained during cyclic shift.

The function of cyclic shift of the shift register equals to senior bit of the current code K.

$$f\left(X_n\right) = \left[\frac{k}{2^{n-1}}\right] = x_0^k \tag{4}$$

Obviously, that function (4) satisfies the condition (3). Function (4) forms N_R rings, each of which includes codes with equal number of ones. Let's denote with $R(k)$ the ring, which includes code k. Suppose L(A)—number of ones in codes of the cyclic ring A. For example, when n = 4 and k = 6: $R(k)$ = {6(0110), 12(1100), 9(1001), 3(0011)}, $L(R(\mathrm{k}))$ = 2.

Each of rings has only one minimal code. It is obvious that for any cyclic ring A ≠ R(0), minimal code q = min(A) is an odd number $\left(x_{n-1}^q = 1\right)$. It means that nonzero minimal code q of the cyclic ring can always be represented as $q = 2 \cdot d + 1$.

4. The Basic Procedure for Combining Cyclic Rings Is in Performing the Following Sequence

1. Initial value of the current code j is chosen randomly, $0 < j < 2^n$. Counter of h codes, for which the value of the feedback function is determined, is set to one: $h = 1$.
2. $u = (2 \cdot j) \bmod 2^n + 1$ is calculated. If calculated code u is minimal in its ring, which means $u = \min(R(u))$, then the value of the feedback function on code j is determined as an inversion of the cyclic shift:
   $f\left(X_j\right) = \left[\frac{j}{2^{n-1}}\right] \oplus 1 = x_0^j \oplus 1$, otherwise $f\left(X_j\right) = \left[\frac{j}{2^{n-1}}\right] \oplus 1 = x_0^j \oplus 1$, where [$x$]—the less of integers, which is greater than x.
3. The new value of the current code j: = $(2 \cdot j) \bmod 2^n + f(X_j)$ is calculated. Increment of the counter is performed: h: = h + 1. If $h \le 2^n$, then return to item 2, otherwise-end.

The function of cyclic transfer forms N_R rings. Described procedure provides a connection of all these rings into one.

Proof: Combining of the rings is performed in pairs. Let's consider random ring B, which consist of codes which include m ones ($0 < m < n$), minimum of them is denoted as $\beta_{\min} = \min(\mathrm{B})$. Previous to his code is $\beta \ge 2^{n-1}$ what is more $\beta_{\min} = 2 \cdot \beta - 2^{n-1} + 1$. Obviously that code $\alpha = \beta - 2^{n-1}$ differs from code β only with one in senior bit and therefore belongs to another ring A. Since $\beta_{\min} = 2 \cdot \alpha + 1$, then respectively to mentioned procedure the feedback function will change its value on a X α set in a way in which code α will be followed by the code $\beta_{\min}$. Value of the function in code β will also be changed in a way when this code will be followed by the code $\alpha_{next} = 2 \cdot \alpha$. In this way the described procedure provides a union of the rings A and B.

Transfer to the minimal code of the B ring possible from one code of the ring A: L(B) = L(A) + 1 except the situation when L(B) = 0. In this case one code is a predecessor of minimal codes for both rings. For example, when n = 4 code 1(0001) precedes the minimal code 0 of the ring {0} and the minimal code 1(0001) of the ring {1, 2, 4, 8}.

The minimal code of each of the rings (except the ring, which consist of zero code) is preceded by the code

with less number of ones from other ring. It means that described procedure provides a binding of all rings, made by cyclic shift.

When k = 1 according to (4) RN(5,1) = 1 and, respectively, exist only one partial ring G_1^1, which includes only one set X_w = {10000}. Respectively, $X'_w = \{0000\}$ and $\varphi(0000) = 1$.

When k = 2 there are two partial rings: G_1^2, G_2^2. When j = 1 from G_1^2—first partial ring, set X_w = {10010}; $X'_w = \{0010\}$ and $\varphi(0010) = 1$ is chosen randomly. When j = 2 from G_2^2—second partial ring, set X_w = {11000} and, respectively, $X'_w = \{1000\}$ и $\varphi(1000) = 1$ is chosen. When k = 3 there are two partial rings: G_1^3, G_2^3. When j = 1 from G_1^3—first partial ring, which contains 3 sets, set X_w = {11010} is randomly chosen. Respectively, $X'_w = \{1010\}$ and $\varphi(1010) = 1$. When j = 2 from G_2^3—second partial ring, set X_w = {11100}: $X'_w = \{1100\}$ and $\varphi(1100) = 1$ is also randomly chosen.When k = 4 there is only one partial ring G_1^4, which includes 4 sets. For example, second of them is randomly chosen: X_w = {11101}. Respectively, $X'_w = \{1101\}$ and $\varphi(1101) = 1$.When k = 5 there is only one partial ring G_1^5, which includes only one set X_w = {11111} for which $X'_w = \{1111\}$ and $\varphi(1111) = 1$.

The resulting table of the values of the Boolean function $\varphi(x_1, x_2, x_3, x_4)$ is showed in **Table 1**.

In algebraic normal form the synthesized function $\varphi(x_1, x_2, x_3, x_4)$ has the following form: $\varphi(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_3 \oplus x_1 \cdot x_3 \oplus x_3 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_4 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_4$. Therefore, the feedback function of the shift register providing repetitive cycle 2^5 will have the following form:

$$f(x_1, x_2, x_3, x_4, x_5) = x_5 \oplus \varphi(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_1 \cdot x_3 \oplus x_3 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_4 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_4 .$$

Let's demonstrate that the proposed method of synthesis of the feedback function is constructive. An operation of setting value of the function $\varphi(x_1, x_2, \cdots, x_{n-1})$ into one which is used in this method corresponds to coupling of the ring which includes code $X_i = \{x_1, x_2, \cdots, x_{n-1}, x_n = 1\}$ and the ring which include code $X_j = \{x_1, x_2, \cdots, x_{n-1}, x_n = 0\}$, where $i \in \{1, \cdots, NR(n,k)\}$, $j \in \{1, \cdots, NR(n, k-1)\}$. As the proposed method suggests a choice of one X code from all rings then it means that each ring is connected with one of the rings, codes of which have less number of ones. Thus all rings are connected into one that provides the maximum period of 2^n repeats of the code in the shift register.

Described procedure for constructing the nonlinear feedback function opens opportunities for optimizations of parameters of generated pseudorandom binary sequences by choosing the ring from set of G. Since the procedure is consistent and includes element of choice, then choice can be done from selected criteria, for example, criterion of the maximum of nonlinearity or compliance to avalanche effect.

For comparative analysis of the effectiveness of the proposed method it is important to estimate the number of nonlinear feedback functions, which can be obtained from its use. The number $NF(n)$ of Boolean functions which allow to synthesize the proposed method is determined by number of variants of choice of codes from partial rings. Since this choice is independent for each partial ring, then numeric value $NF(n)$ is determined as a product of a number of variants for choice. With fixed number of k ones in the code of the ring, which contains n codes, number of variants to choose the code from partial ring is k. The total number of such rings is $NF(n)$, numeric value of which is determined by the formula (2.12). Therefore, the total number of variants for choice of the code which contains k ones from the rings with length n is $k^{NR(n,k)}$. For rings with less length the number of variants of choice of the code is $\frac{k \cdot d}{n}$, where d—divider of n and number of such rings is $NR\left(d, \frac{k \cdot d}{n}\right)$. Thus, the total number of

Table 1. Values of $\varphi(x_1, x_2, x_3, x_4)$.

X'	$\varphi(X')$	X'	$\varphi(X')$
0 0 0 0	1	1 0 0 0	1
0 0 0 1	0	1 0 0 1	0
0 0 1 0	1	1 0 1 0	1
0 0 1 1	0	1 0 1 1	0
0 1 0 0	0	1 1 0 0	1
0 1 0 1	0	1 1 0 1	1
0 1 1 0	0	1 1 1 0	0
0 1 1 1	0	1 1 1 1	1

$NF(n)$ feedback functions which allows to synthesize the proposed method is defined by the following expression:

$$NF(n) = \prod_{k=1}^{n-1} k^{NR(n,k)} \cdot \prod_{d=2}^{n/2} \psi(n,k,d), \text{єдиное}$$

$$\psi(n,k,d) = 1 : \xi(n,d) \cdot \xi\left(d, \frac{k \cdot d}{n}\right) = 0 \qquad (5)$$

$$\psi(n,k,d) = \left(\frac{k \cdot d}{n}\right)^{RN\left(d, \frac{k \cdot d}{n}\right)} : \xi(n,d) \cdot \xi\left(d, \frac{k \cdot d}{n}\right) = 1$$

Analysis shows that developed method allows synthesizing 10 times more functions comparing with knows methods.

For n = 5 the proposed method allows to obtain $NF(5)$ = 144 feedback functions, when the existent methods propose not more than 15 functions [8]. At the same time the proposed method is quite simple and technological [4] that provides higher performance compared to known methods. The proposed method is realized in software and successfully passed all tests.

The disadvantage of the method of building LFSR based on described procedure of combining elements of closed code groups is the fact, that time and amount of memory proportional to 2^n is required for its implementation. A modification is proposed to significantly decrease computational complexity by narrowing the class of synthesized feedback functions and optimization of the procedure of building the function in disjunctive normal form.

Suppose the current vector of the values of bits of n-bit shift register is $R = \{r_0 r_1 \cdots r_{n-1}\}$. As it was proved earlier feedback function $f(R)$ provides full repetition period if on sets to which the minimal in its ring $w = \sum_{k=0}^{n-1} r_k \cdot 2^{n-k-1}$ code corresponds, $f(R)$ equals the inversion of the senior bit of the set:

$$f(R) = r_0 \oplus 1$$

and $f(R) = r_0$ on other sets.

Feedback function will formed as:

$$f(R) = \overline{r_0} \cdot \overline{\mu(X)} \vee r_0 \cdot \mu(X)$$

where $X = \{x_0, x_1, \cdots, x_{n-1}\} = \{r_1, r_2, \cdots, r_{n-1}, 1\}$ —vector of bits obtained by shifting R to the left by one bit with filling vacated junior bit with one; $\mu(X)$—function which takes zero value if X corresponds to the minimal value of the code in the ring and one otherwise.

$\mu(X)$ can be formed as disjunction of all t_{ij} terms:

$$t_{ij} = \mathop{\&}_{k=0}^{i-2} \left(x_k \oplus x_{(j+k) \bmod n}\right) \cdot x_{i-1} \cdot x_{(j+i-1) \bmod n},$$
$$i = 1, \cdots, n-1, \quad j = 1, \cdots, n-1.$$

Set cannot be minimal if it contains at least one sequence of bits $S_j = \{x_j, x_{j+1}, \cdots, x_{(j+i-1) \bmod n}\}$ with length i, with initial bit in position j, $0 < j < n-1$, if numeric value of such sequence is less than value of the sequence with the same length of i senior bits of the set $S_0 = \{x_0, x_1, \cdots, x_{i-1}\}$ .. (If $j + k > n$, then senior bits of the set are used).

In particular, the set isn't minimal if $i - 1$ of corresponding senior bits of this sequence are equal and the junior bit of the sequence S_0 is greater than junior bit of the sequence S_j:

$$x_0 = x_j,$$
$$x_1 = x_{(j+1) \bmod n},$$
$$\vdots$$
$$x_{i-2} = x_{(j+i-2) \bmod n},$$
$$x_{i-1} > x_{(j+i-1) \bmod n}.$$

In each term t_{ij} check of satisfying this condition is performed. If $i-1$ of corresponding senior bits of sequences S_0 and S_j are equal, and $x_{i-1} > x_{j+i-1}$, then t_{ij} is equal to one that means that this set isn't the minimal in its ring.

Condition is checked for all length of sequences $i = 1, \cdots, n-1$, and for all values distances between initial sets of sequences $j = 1, \cdots, n-1$. Combining all conditions through the operation of disjunction as it is during formation of the $\mu(X)$ function, provides a check if the value of the code, to which vector X corresponds, is minimal in the ring. In **Table 2** there is an example of formation all terms t_{ij} for a 5-bit register.

To extend the procedure of formation of the feedback functions, which provide full period, the previously described method of pre-unification of the two rings can be used. Let's consider examples of work of the $f_1(X)$ function for $n = 5$ with different values of bits vector X.

Example 1. $X = \{x_0, x_1, x_2, x_3, x_4\} = \{0,0,1,0,1\}$

Let's determine terms for sequences with the length of one. The sequence of senior bits is $S_0 = \{x_0\} = \{0\}$. Next, let's compare X_0 with each bit x_1, x_2, x_3, x_4:

$$t_{11} = x_0 \cdot x_1 = 0,$$
$$t_{12} = x_0 \cdot x_2 = 0,$$
$$t_{13} = x_0 \cdot x_3 = 0,$$
$$t_{14} = x_0 \cdot x_4 = 0.$$

Note that for all sets with zero senior bit, values of t_{1j} will always be zero.

Let's determine terms for sequences with length $i = 2$: $S_0 = \{0, 0\}$.

$$p_{21} = (x_0 \oplus x_1) = (0 \oplus 0) = 1, \quad t_{21} = p_{21} \cdot x_1 \cdot x_2 = 1 \times 0 \times 1 = 0;$$
$$p_{22} = (x_0 \oplus x_2) = (0 \oplus 1) = 0, \quad t_{22} = p_{22} \cdot x_1 \cdot x_3 = 0 \times 0 \times 0 = 0;$$
$$p_{23} = (x_0 \oplus x_3) = (0 \oplus 0) = 1, \quad t_{23} = p_{23} \cdot x_1 \cdot x_4 = 1 \times 0 \times 1 = 0;$$
$$p_{24} = (x_0 \oplus x_4) = (0 \oplus 1) = 0, \quad t_{24} = p_{24} \cdot x_1 \cdot x_0 = 0 \times 0 \times 0 = 0.$$

It is obvious that all t_{2j} have zero value. As long as values of the following t_{ij} depend on p_{2j}, we will determine terms only for those values j, where p_{2j} has value of one, thus $j = 1$ and $j = 3$. The length of the sequence $i = 3$, $S_0 = \{0, 0, 1\}$. Let's find terms for $j = 1$ and $j = 3$.

$$p_{31} = p_{21} \cdot (x_1 \oplus x_2) = 1 \times (0 \oplus 1) = 0, \quad t_{31} = p_{31} \cdot x_2 \cdot x_3 = 0 \times 0 \times 1 = 0;$$
$$p_{33} = p_{23} \cdot (x_1 \oplus x_4) = 1 \times (0 \oplus 1) = 0, \quad t_{33} = p_{33} \cdot x_2 \cdot x_3 = 0 \times 0 \times 1 = 0;$$

Zero values of p_{31} and p_{33} during calculations of the following terms convert the result into zero. Therefore, all t_{ij} for this set are equal to 0. $f_1(X)$, formed as disjunction of all t_{ij} terms, also obtains zero value. It means that code

Table 2. Terms t_{ij} for $n = 5$.

	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	$x_0 \cdot x_1$	$x_0 \cdot x_2$	$x_0 \cdot x_3$	$x_0 \cdot x_4$
$i = 2$	$(x_0 \oplus x_1) \cdot$ $\cdot x_1 \cdot x_2$	$(x_0 \oplus x_2) \cdot$ $\cdot x_1 \cdot x_3$	$(x_0 \oplus x_3) \cdot$ $\cdot x_1 \cdot x_4$	$(x_0 \oplus x_4) \cdot$ $\cdot x_1 \cdot x_0$
$i = 3$	$(x_0 \oplus x_1).$ $\cdot (x_1 \oplus x_2).$ $\cdot x_2 \cdot x_3$	$(x_0 \oplus x_2).$ $\cdot (x_1 \oplus x_3).$ $\cdot x_2 \cdot x_4$	$(x_0 \oplus x_3).$ $\cdot (x_1 \oplus x_4).$ $\cdot x_2 \cdot x_0$	$(x_0 \oplus x_4).$ $\cdot (x_1 \oplus x_0).$ $\cdot x_2 \cdot x_1$
$i = 4$	$(x_0 \oplus x_1).$ $\cdot (x_1 \oplus x_2).$ $\cdot (x_2 \oplus x_3).$ $\cdot x_3 \cdot x_4$	$(x_0 \oplus x_2).$ $\cdot (x_1 \oplus x_3).$ $\cdot (x_2 \oplus x_4).$ $\cdot x_3 \cdot x_0$	$(x_0 \oplus x_3).$ $\cdot (x_1 \oplus x_4).$ $\cdot (x_2 \oplus x_0).$ $\cdot x_3 \cdot x_1$	$(x_0 \oplus x_4).$ $\cdot (x_1 \oplus x_0).$ $\cdot (x_2 \oplus x_1).$ $\cdot x_3 \cdot x_2$

w, corresponding to $X = \{0, 0, 1, 0, 1\}$ set, has the minimal value in the ring. Really, $w = 16 \times 0 + 8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 5$ and the ring, where it is included, is {5, 10, 20, 9, 18}. We can see that 5 is the minimal value.

Example 2. $X = \{x_0, x_1, x_2, x_3, x_4\} = \{0,1,1,0,1\}$

2.1 Since $x_0 = 0$, then terms $t_{11} = t_{12} = t_{13} = t_{14} = 0$.

2.2 Let's determine term for sequences with length of $i = 2$: $S_0 = \{0, 1\}$.

$$p_{21} = (x_0 \oplus x_1) = (0 \oplus 1) = 0, \quad t_{21} = p_{21} \cdot x_1 \cdot x_2 = 0 \times 1 \times 1 = 0;$$
$$p_{22} = (x_0 \oplus x_2) = (0 \oplus 1) = 0, \quad t_{22} = p_{22} \cdot x_1 \cdot x_3 = 0 \times 1 \times 0 = 0;$$
$$p_{23} = (x_0 \oplus x_3) = (0 \oplus 0) = 1, \quad t_{23} = p_{23} \cdot x_1 \cdot x_4 = 1 \times 1 \times 1 = 0;$$
$$p_{24} = (x_0 \oplus x_4) = (0 \oplus 1) = 0, \quad t_{24} = p_{24} \cdot x_1 \cdot x_0 = 0 \times 0 \times 0 = 0.$$

All terms for t_{2j} have zero values. Among values of p_{2j} there is only one value $p_{23} = 1$, which can be used for further calculations. The rest of these values turn terms, to which they are included, into zero. Let's consider the sequence of length $i = 3$. The sequence of three senior $S_0 = \{0, 0, 1\}$. For all j values except $j = 3$, terms t_3j will be zero. Determining p_{33} and t_{33}:

$$p_{33} = p_{23} \cdot (x_1 \oplus x_4) = 1 \times (1 \oplus 1) = 1, \quad t_{33} = p_{33} \times x_2 \times x_3 = 1 \times 1 \times 0 = 1.$$

If we have at least one term with value of 1, the value of the function $f_1(X)$ will also be one.

Therefore, for $X = \{0, 1, 1, 0, 1\}$, $f_1(X) = 1$, it means code $w = 13$ isn't a minimum in its ring. Let's make sure that it is true. We obtain a ring by cyclic shift: {13, 26, 21, 11, 22}. Here we can see that the minimal value in this ring is 11, not 13. **Table 3** shows the result of construction of the feedback function for a 5-bit shift register with minimal repetitive cycle.

Complexity of the developed algorithm of construction of the feedback function in Zhegalkin's algebra is equivalent to the function itself.

Let's estimate the complexity of the $f_1(X)$ function. To calculate one term t_{ij} the following number operations are required:

- $i - 1$ operations NXOR $(\oplus)$ to compare corresponding bits of the sequences;
- 1 operation AND to check the condition $x_{i-1} > x_{(j+i-1) \bmod n}$;
- $i - 1$ operations AND to unite all conditions.

So, in total we have $2i - 1$ operations. The total complexity of calculation the terms for all values i and j is:

$$\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} (2 \cdot i - 1) = 2 \cdot (n-1) \cdot \sum_{i=1}^{n-1} (2 \cdot i - 1) = 2 \cdot (n-1)^3 \quad (5)$$

Another $(n - 2)^2$ operations OR are required to unite all terms, that is to calculate the function $\mu(X)$ $2(n - 1)^3 + (n - 2)^2$ operations are required, therefore, the complexity is proportional to n^3.

As we can see, the function is redundant. In each term t_{ij} calculations, which are made for t_{i-1j}, are performed. During paired comparison of the elements of the sequences S_0 and S_j with length of $i > 2$ the results of the first $i - 2$ elements are known form the previous step. That is why the formula to determine t_{ij} can be written as:

$$t_{ij} = p_{i,j} \cdot x_{i-1} \cdot x_{(j+i-1) \bmod n},$$

where $p_{i,j}$-term, which compares for equality senior $i - 1$ bits of the sequences.

p_{ij} can be determined as:

$$p_{ij} = 1, \text{ if } i < 2, \qquad p_{ij} = (x_{i-2} \oplus x_{(j+i-2) \bmod n}) \cdot p_{i-1j}, \text{ if } 2 < j < n-1.$$

Since for calculations of each t_{ij} constant number of operations is performed (1 operation when $i = 1$ and 4 otherwise), the complexity of the function $\mu(X)$ will be proportional to n^2: $4n(n-1) + 2n = 2n(2n-1)$.

The amount of memory required to perform the algorithm is determined by necessity to save all terms t_{ij}, $i = 1, \cdots, n-1$, $j = 1, \cdots, n-1$. Therefore, memory expenses are proportional to n^2.

5. Conclusions

An approach is proposed for the construction of the important element of the contemporary systems for the pro-

Table 3. Construction of the feedback function.

No.	R	X	μ	f
0	00000	00001	0	1
1	00001	00011	0	1
2	00010	00101	0	1
3	00011	00111	0	1
4	00100	01001	1	0
5	00101	01011	0	1
6	00110	01101	1	0
7	00111	01111	0	1
8	01000	10001	1	0
9	01001	10011	1	0
10	01010	10101	1	0
11	01011	10111	1	0
12	01100	11001	1	0
13	01101	11011	1	0
14	01110	11101	1	0
15	01111	11111	0	1
16	10000	00001	0	0
17	10001	00011	0	0
18	10010	00101	0	0
19	10011	00111	0	0
20	10100	01001	1	1
21	10101	01011	0	0
22	10110	01101	1	1
23	10111	01111	0	0
24	11000	10001	1	1
25	11001	10011	1	1
26	11010	10101	1	1
27	11011	10111	1	1
28	11100	11001	1	1
29	11101	11011	1	1
30	11110	11101	1	1
31	11111	11111	0	0

tection of information-generators of pseudorandom binary sequences with the repetition period 2^n on the basis of n-bit shift register with the nonlinear feedback function. Such generators are considerably simpler and more efficient compared with the generators constructed in the form of LFSR system and nonlinear inverter.

Within the common approach, based on the technology of the association “rings”, realization of which requires memory with size 2^n, is developed with high-tech modification, which makes it possible to build the generators of pseudorandom sequences on the basis of the shift register with the nonlinear feedback function of with the use of memory, proportional to n^2. The use of the method proposed makes it possible to build a suitable for the practical use in the telecommunication technologies. An experimental study made it possible with the use of the method

proposed to build generators of word length up to 128 bit.

References

[1] Obeidat, A. (2012) Burst Error Correction Method Based on Arithmetic Weighted Checksums. *Engineering*, **4**, 768-773. http://dx.doi.org/10.4236/eng.2012.411098

[2] Luby, M. (1996) Pseudorandomness and Cryptographic Applications. Princeton University Press, Princeton, 273 p.

[3] Golomb, S.W. (1982) Shift Register Sequences. Aegean Park Press, Laguna Hill, 324 p.

[4] Menezer, A.J., Van Oorschot, P.C. and Vanstone, S.A. (1997) Handbook of Applied Cryptography. CRC Press, Boca Raton, 780 p.

[5] Sarkar, P. and Maitra, S. (2001) Efficient Implementation of "Large" Stream Cipher Systems. *Proceeding of* 3*th International Workshop* "*Cryptographic Hardware and Embedded Systems*" (*CHES*-2001), Springer-Verlag, 319-332.

[6] Key, E.L. (1976) An Analysis of the Structures and Complexity of Nonlinear Binary Sequence Generators. *IEEE Transaction on Information Theory*, **22**, 732-736. http://dx.doi.org/10.1109/TIT.1976.1055626

[7] Sidorenko, V., Richter, G. and Bossert, M. (2011) Linearized Shift-Register Synthesis**.** *IEEE Transaction on Information Theory*, **57**, 6025-6032.

[8] (2000) NIST Special Publicaion 800-22: A Statistical Test Suite for Random and Pseudorandom Number. 348 p.

[9] Gutierrez, J., Shparlinski, I.A. and Winterhof, A. (2003) On the Linear and Nonlinear Complexity Profile on Nonlinear Pseudorandom Number Generators. *IEEE Transaction on Information Theory*, **49**, 60-64.