UDC 004.421.5

A PSEUDORANDOM SEQUENCES GENERATOR BASED ON THE MULTIMODULO TRANSFORMATION

Yurii Gorbenko[1], Tetiana Grinenko[2], Oleksii Nariezhnii[3], Nikołay Karpinskiy[4]

[1,3] V.N. Karazin Kharkiv National University, Svobody sq., 4, Kharkov, 61022, Ukraine
narlexa69@mail.ru

[2] Kharkiv National University of Radio Electronics, Nauka Ave, 14, Kharkov, 61166, Ukraine
t_lame@mail.ru

[4] University of Bielsko-Biala, Willowa St., 2, 43-309 Bielsko-Biala, Poland
mkarpinski@ath.bielsko.pl

Reviewer: Victor Krasnobayev, Dr., Full Professor, V.N. Karazin Kharkiv National University, Svobody sq., 4, Kharkov, 61022, Ukraine
krasnobayev@karazin.ua

Received on January 2016

Abstract. *Main theoretical statements and practical research results of pseudorandom sequences over arbitrary alphabet generation based on multimodulo transformation in the finite field GF(p^n) are given, results of properties analysis on distinguishing, unpredictability, irreversibility, repetition period and complexity (performance) are brought.*

Keywords: *pseudorandom sequence, pseudorandom sequence generator, multimodulo transformation, Galois field, distinguishing, unpredictability, irreversibility, repetition period, complexity.*

1 Introduction

The keys management tools are important components of cryptography systems, which characteristics and properties on, depend their resistance and the level of security in whole. At the different stages of key management it is needed to generate key data, key information and different options, having quite complex properties requirements. In practice, depending on the requirements, two methods are applied for key generation, based on random and pseudorandom sequences (PRS), which are brought about in the form of corresponding cryptographic tools.

As main demands to such generators are set out requirements of direct and reverse unpredictability (structural security), irreversibility concerning the used key, distinguishing of sequences, promptitude and repetition period difficulties for pseudorandom sequences are set out [1]. Wherein the level of key generators warranty depends to a considerable extent on the key source entropy, which should be from 128 to 512 bits for now.

Nowadays was developed a range of methods and PRS generation means on its basis. Their peculiarity is that they are built, well researched and applied as a rule for alphabet with $m = 2$ basis. At the same time a range of updates needs PRS generation means that can be resumed in space and time with acceptable complexity and random basis beginning with $m = 2$. The studies have shown that this problem can be solved through the transformations known as multimodulo.

Some regulations of multimodulo transformations for prime field $GF(p)$ are published in the work [2]. PRS generation on basis of multimodulo transformation in Galois field $GF(p)$ is offered in the work [2]. Such a method really allows generating PRS with random alphabet *m*, specified period of repetition and certain but not researched enough distinguishing properties. The elaboration of PRS generation method with certain alphabet of *m* symbols on basis of multimodulo transformations using Galois field $GF(p)$ elements, besides results of irreversibility and distinguishing properties research are published in the work [3]. The work [3] consists of definition of the conditions of pseudorandom sequences existence with equally possible letter distribution of *m* alphabet in

the class of multimodulo transformations and valuating of lower limit of irreversibility.

But in the mentioned works [2,3] a range of theoretical grounds of properties doesn't have generalized character of unified theory, in addition to that there wasn`t undertaken enough field research, which would verify theoretical results as regard to distinguishing, irreversibility, unpredictability, repetition period and complexity. The results of studies in works [2,3] also have constrained character, as they were undertaken only for multimodulo transformations over prime Galois field $GF(p)$.

The aim of the work is development of theoretical basis of PRS generation method with arbitrary alphabet of *m* symbols based on multimodulo transformations using elements of arbitrary Galois field, which at the theoretical stage would allow providing properties of distinguishing, irreversibility, unpredictability, repletion period and complexity for the finite field $GF(p^n)$ [1,4-6]. As a regard to this method it is needed to undertake a range of theoretical and field studies concerning definition of necessary and sufficient conditions of providing of predetermined repetition period, alphabet basis, probability of appearance of alphabet symbols at repetition period, features of irreversibility, unpredictability and distinguishing considering guarantees [4-7].

2 Method of multimodulo transformation in the finite field

Let us consider PRS generation method with certain symbols alphabet, for example *m*, on based on arbitrary Galois field $GF(p^n)$. For general case we will think that there is made up *k* transformations of units of Galois field $GF(p^n)$ extension, corresponding to modules $(f(X), f_1(X)),(f_1(X), f_2(X)),\ldots,(f_{k-2}(X), f_{k-1}(X))$ and the last module *m*. General options which are enough to generate elements a_i of $GF(p^n)$ field is tuple $(f(X), p, n, \theta_j)$, where $f(X)$ – irreducible polynomial of degree *n* over finite field $GF(p)$, and θ_j – primary element chosen from magnitude $\{\theta\}$ of division $\varphi(p^n-1)$, where $\varphi(\ )$ – Euler's function [8]. In such a case generation of field elements is carried out according to the rule:

$$a_i = (\theta_j)^i (\mathrm{mod}(f(X), p, n)). \quad (1)$$

It is shown [9], that if the above-said requirements to tuple $(f(X), p, n, \theta_j)$ have been fulfilled, (1) would generate finite Galois field with repetition period p^n-1. Let us denote that above-said is true for $p = 2,3,5,7$ and subsequent prime numbers. When $p = 2$, there will be extension of Galois field $F(2)$.

In the following, let $(f_s(X), p_s, n_s)$ be tuple of general options, for example of polynomials (among them irreducible) $f_s(X), s = (1, k-1)$, and n_s – their degrees, from this point on we need irreducibility of polynomials to provide their coprimality when necessary [9].

Also let degrees of polynomials (among them irreducible) n_s fulfill requirements:

$$n_1 > n_2,\ n_2 > n_3,\ \ldots,\ n_{k-2} > n_{k-1}, \quad (2)$$

wherein basis of m alphabet is any number, besides ineqautions are fulfilled:

$$p^{n_1} >> p^{n_2},\ p^{n_2} >> p^{n_3},\ldots, p^{n_{k-2}} >> p^{n_{k-1}},\ p^{n_{k-1}} >> m. \quad (3)$$

The statement 1 is fair.

Statement 1.

Deterministic PRS generator, which is functioning according to algorithm of multimodulo transformations:

$$b_i = ((\theta_j)^i (\mathrm{mod}(f(X), p, n), (f_1(X), p_1, n_1), (f_2(X), p_2, n_2),\ldots \quad (4)$$

$$\ldots, (f_{k-1}(X), p_{k-1}, n_{k-1}), (f_m(X), m)),$$

where $(f_s(X), p_s, n_s)$ – tuple of general options, m – certain integer, k – degree of multi modulari-

ty, p_m – number (not necessarily prime), m – positive integer, provides generation of PRS (symbols) with repetition period $p^n - 1$, equally possible with a certain basis of m alphabet, under condition that:

1) (1)–(3) requirements are fulfilled;

2) modules (couple of polynomials)

$$\left(f(X), f_1(X)\right), \left(f_1(X), f_2(X)\right), \ldots, \left(f_{k-2}(X), f_{k-1}(X)\right) \tag{5}$$

are coprime and tuple $\left(f_m(x), m\right)$ is undefined.

In (4) $\left(f_m(x), m\right)$ means that module *m* is given as a polynomial.

Under fulfillment of (4)–(5) conditions PRS (symbols) generation is provided with following properties and characteristics:

– arbitrary alphabet *m* basis;

– $p^n - 1$ repetition period;

– symbols are generated equally possible or "almost" equally possible;

– by ensemble of isomorphism's $\varphi\left(p^n - 1\right)$.

The statement 2 is fair too.

Statement 2.

Deterministic PRS generator, which is functioning according to algorithm of multimodulo transformations:

$$b_i = \left(\left(\theta_j\right)^{K_0+i}\left(\mathrm{mod}\left(f(X), p, n\right), \left(f_1(X), p_1, n_1\right), \left(f_2(X), p_2, n_2\right), \ldots\right.\right. \tag{6}$$

$$\left.\left. \ldots, \left(f_{k-1}(X), p_{k-1}, n_{k-1}\right), \left(f_m(X), \overleftarrow{m}\right)\right)\right),$$

where $K_0 + i$ – current generator key, wherein K_0 is primary key and i is session key, which is noninvertible with complexity not less than $O(n)$ [10].

Let us further observe isolated case of statements 1 and 2 for three modulo transformation, when elements of Galois field extension are also generated according to (1), but (2)–(6) take the form of (7)–(10):

$$n_1 > m. \tag{7}$$

$$p^{n_1} >> p^m. \tag{8}$$

$$b_i = \left(\left(\theta_j\right)^i\left(\mathrm{mod}\left(f(X), p\right), \left(f_1(X), p_1, n_1\right), \left(f_m(X), \overleftarrow{m}\right)\right)\right). \tag{9}$$

$$b_i = \left(\left(\theta_j\right)^{K_0+i}\left(\mathrm{mod}\left(f(X), p\right), \left(f_1(X), p_1, n_1\right), \left(f_m(X), \overleftarrow{m}\right)\right)\right). \tag{10}$$

For conditions (7)–(10) let us present statement 1 for three modulo transformation in form of theorem 1.

Theorem 1. Deterministic PRS generator, which is functioning according to algorithm of multimodulo transformations on the basis (1) according to the rules:

$$b_i = \left(\left(\theta_j\right)^i\left(\mathrm{mod}\left(f(X), p, n\right), \left(f_1(X), p_1, n_1\right), \left(f_m(X), \overleftarrow{m}\right)\right)\right) \tag{11}$$

or

$$b_i = \left((\theta_j)^{k_0+i} \left(\mathrm{mod}\left(f(X), p, n \right), \left(f_1(X), p_1, n_1 \right), \left(f_m(X), \tilde{m} \right) \right) \right), \quad (12)$$

under fulfillment of conditions (2)–(8) provides generation of PRS (symbols) numbers with undefined basis of *m* alphabet, with repetition period $p^n - 1$, with equally possible appearance of symbols at the repetition period $p^n - 1$ and with ensemble of isomorphism $\varphi(p^n - 1)$.

Theorem 1 for three modulo transformation proving.

Regarding the last module *m* it can take arbitrary value and it will be presented as polynomial. Let us mark that $f(x)$ and $f_1(x)$ in (11) are irreducible polynomials, which can be presented over the field $F(2)$, i.e. as polynomial of n degree over $F(2)$.

In regard to repetition period, since $\{\theta_i\}$ – primary elements, for providing maximum period $p^n - 1$ it is necessary and enough for $f(x)$ to be irreducible over the field $GF(p^n)$ [9]. Since $f(x)$ is irreducible over the field $GF(p^n)$, according to (1) elements of Galois field are generated with period $p^n - 1$ and each element appears only one time.

Let us define m-symbols (finite alphabet) appearance equiprobability degree, i.e. define conditions, under which symbols of m alphabet appear equally possible. Symbols will be determined with the help of polynomials $f_m(x)$ not higher than n_m degree.

Let us present all elements of field $GF(p^n)$ as positive integers from $\theta^0 = 1$ to $p^n - 1$.

Then let us sort numbers $1 \div p^n - 1$ according to the ascending order

$$1, 2, 3, \ldots, |f(x)|, |f(x)|+1, \ldots, 2|f(x)|, 2|f(x)|+1, \ldots, 3|f(x)|, 3|f(x)|+1, \ldots \quad (13)$$

$$\ldots, p^n - 1 - f(x), p^n - f(x), \ldots, p^n - 1,$$

where $|f(x)|$ is element value of field $f(x)$.

Let us bring the row (13) according to module $|f_1(x)|$, as a result we will get:

$$1, 2, 3, \ldots, |f_1(x)|\text{-}1, 0, 1, 2, 3, \ldots, |f_1(x)|\text{-}1, 0, 1, 2, 3, \ldots, |f_1(x)|\text{-}1, 0, 1, \ldots, V, \quad (14)$$

where $0 \le V \le |f_1(x)| - 1$.

Let us present the array (14) as:

$$\overbrace{1,2,3,\ldots,|f_1(x)|-1,0;}^{1} \ \overbrace{1,2,3,\ldots,|f_1(x)|-1,0;}^{2} \ldots \quad (15)$$

$$\ldots; \overbrace{1,\ 2,\ 3,\ldots,\ |f_1(x)|-1,0;}^{z-1} \ \overbrace{1,\ 2,\ 3,\ldots,\ V}^{z},$$

where $V \le |f_1(x)| - 1$.

On the whole there will be sequence elements $((z-1)|f_1(x)| + V$ in the PRS (15). Besides in the last unit there will be no sequence elements beginning with $(V+1)$ to $|f_1(x)| - 1$ and 0.

Farther on symbols $1, 2, 3, \ldots, V$ appear z times $V+1, V+2, \ldots, |f_1(x)| - 1, 0$ – $(z-1)$ times.

Probabilities of elements $1, 2, 3, \ldots, V$ appearance will be correspondingly:

$$R_1 = \frac{z}{p^n - 1}, \quad (16)$$

and $V+1, V+2, \ldots, |f_1(x)| - 1, 0$

$$R_2 = \frac{z-1}{p^n - 1}. \quad (17)$$

Thus symbols $1,2,3,\ldots,|f_1(x)|-1,0$ appear almost with almost equal probability, i.e. equally possible at the period $p^n - 1$ as a result of conversion according to the second $|f_1(x)|$ module.

Let us observe the stage of conversion according to the third module, which is according to the theorem 1, can be undefined number $|f_m(x)|$.

While analyzing for frequency let us define an array $0,1,2,3,\ldots,|f_1(x)|-1$ as

$$1,2,3,\ldots,|f_1(x)|. \quad (18)$$

We will bring (18) together according to module $|f_m(x)|$ and get the row:

$$1,2,3,\ldots,|f_m(x)|-1,0,1,2,3,\ldots,|f_m(x)|-1,0,1,2,3,\ldots,|f_m(x)|-1,0,1,2,3,\ldots,V, \quad (19)$$

where $0 \le V \le |f_m(x)|-1$.

Analyzing in (19) $0,1,2,3,\ldots,|f_m(x)|-1$ symbols appearance probability we will get the same assed values as in (16) and (17).

It is also should be pointed out that in (16) V symbols appearance unequiprobability is no more than 1 in number of appearance of symbols $0,1,2,3,\ldots,V$, and also as an assessed value of probability for each symbol $\Delta p = \frac{1}{p^n - 1}$.

Thus theorem 1 for three modulo transformation is proved. Also it should be mentioned that above-described theorem 1 proving can be applied to *k*-modulo transformation, of course under condition when couples of polynomials $(f(X), f_1(X)),(f_1(X), f_2(X)),\ldots,(f_{k-2}(X), f_{k-1}(X))$ are coprime and tuple $f_m(X), m$ is undefined, module *m* value is meant.

On the whole the procedure of PRS generation based on multimodulo transformation can be brought to the following.

1. To set or generate system options – general options tuples $(f_s(X), p_s, n_s)$ according to the requirement of statement 1.

2. To set or install secret key of generator k, $k = 1 \div p^n - 1$.

3. Determine initial value of generator a_0 using the rule:

$$a_0 = \theta^k (\mathrm{mod}(f(x), n)),$$

where $(f(x), n)$ – basic transformation module.

4. Determine element a_i of generator using the rule:

$$a_i = a_{i-1}\theta(\mathrm{mod}(f(x), n)) = R_{(f(x),n_1)}(a_0\theta^i),$$

where $i \ge 1$ – number of PRS generating element, a_{i-1} – $(i-1)$ element of an array over a field of extension p^n.

5. Determine element b_i of PRSG using the rule:

$$b_i = a_i(\mathrm{mod}(f_1(x), n_1)) = R_{(f_1(x),n_1)}(a_i) = R_{(f_1(x),n_1)}\left(R_{(f(x),n)}(a_0\theta^i)\right),$$

where $1 < (f_1(x), n_1) < (f(x), n)$.

6. Determine element c_i of PRSG using the rule:

$$c_i = R_{(f_n(x),m_n)}\left(R_{(f_{n-1}(x),m_{n-1})}\left(\ldots\left(R_{(f_2(x),m_2)}\left(R_{(f_1(x),m_1)}(a_0\theta^i)\right)\right)\ldots\right)\right), 0 \le i \le \varphi(p),$$

where $i \ge 1$ – the number of PRS generating element, $(f_1(x), n_1),\ldots,(f_n(x), n_n)$ – intermediate modules.

7. If necessary determine hash-value number i from b_i and accept it as random word number i, i.e.:

$$y_i = H(b_i).$$

The scheme of algorithm (variant) that implements above-described method of determined random number generator (DRNG) is illustrated on fig. 1.

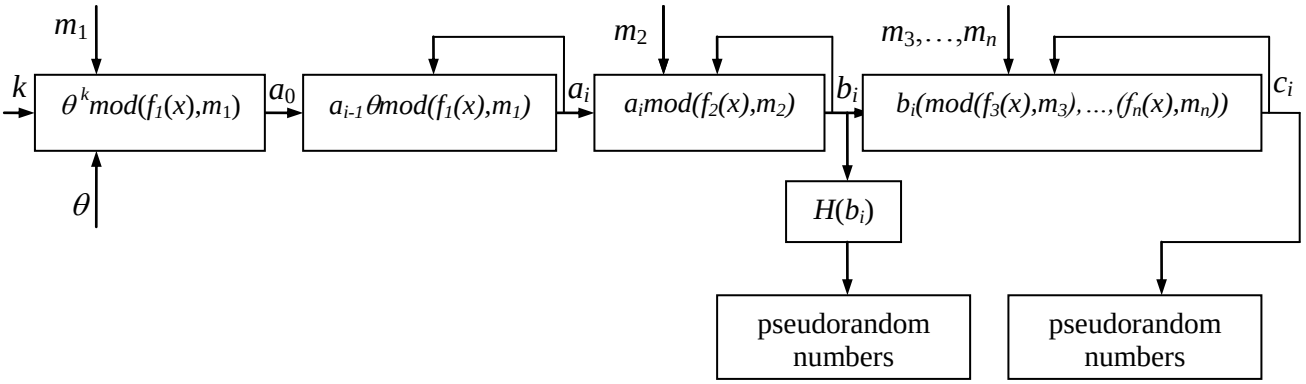

Fig.1 – The scheme of algorithm of determined random sequences generation in the finite field of $p^n - 1$ division by method of multimodulo transformation

3 Properties of PRS of multimodulo transformations

Let us farther observe the method of PRS generation with certain alphabet of symbols, e.g. m, based on multimodulo transformations in finite Galois field $GF(p^n)$, $n \geq 1$. It is considered that k transformations of Galois field $GF(p^n)$ extension elements are carried out according to modules $(f(X), f_1(X)), (f_1(X), f_2(X)), \ldots, (f_{k-2}(X), f_{k-1}(X))$ and the last module m. General options is tuple $(f(X), p, n, \theta_j)$, where $f(X)$ – irreducible polynomial of *n* degree over field $GF(p)$, and θ_j – primary element chosen from magnitude $\{\theta\}$ of division $\varphi(p^n - 1)$.

We will also observe special case of theorem 1 for three modulo transformation. In this case Galois field extension elements are also generated according to (1). And in such a case (2)–(6) look like:

$$n_1 > m \text{ i } p^{n_1} >> p^m;$$

$$b_i = \left((\theta_j)^{K_0+i} \left(\operatorname{mod}(f(X), p), (f_1(X), p_1, n_1), \left(f_m(X), \overleftarrow{m} \right) \right) \right),$$

where $K_0 + i$ is current generator key, $K_0 + i$ is primary key and i is session key as above.

Complexity assessment of PRS generator inversion.

Let us make complexity assessment of discrete logging for three modulo and multimodulo transformations.

In a case of finite Galois field $GF(p)$ we have:

$$b_i = \left((\theta_j)^X (\operatorname{mod}(P), (P_1), (m)) \right), \quad (20)$$

where $X = K_0 + i$ belongs to definition, under condition, that some array of symbols b_i is known, primary element θ_j and tuple of options (P, P_1, m).

While using «brute force» attack can be applied the following main methods: keys search, table attack and attack with dictionary [11,12].

While applying «brute force» attack it is considered that the length of key k is not more than the one of generated PRS and counterfeiter while searching key X, make an attempt to get a value

$$b_i^* = b_i . \quad (21)$$

Under condition fulfillment (21) generator key will be determined.

For assessment of possibility of applying «brute force» attack can be used such data as N_k – number of keys, safe time t_s, P_p – probability of successful cryptanalysis, etc [11,12,13]. Value t_s can be determined according to the formula [13]:

$$t_s = \frac{N_k}{\gamma K} P_p ,$$

where γ – capacity of cryptanalytic system, $K = 3.15 \cdot 10^7$ – the number of seconds in a year.

Table attack and attack with dictionary based on using mathematical tool called «birthday problem»: method of collisions creation [14]. For this method options: collisions probability P_k, cryptanalyst's attempts number k and exhaustive set of possible output values n are bounded with each other with parametric equation [14,15]:

$$1 - P_k = e^{-(k(k-1))/2n}$$

or of closed form :

$$k^2 - k + 2n \ln(1 - P_k) = 0 . \quad (22)$$

Correlation (22) allows assessing a number of experiments needed to carry out to implement collision with applying mathematical tool «birthday problem».

In some cases couple «generator key – PRS output unit» can be received with the help of a dictionary. In such a case couples «generator key – PRS output unit» are generated or collected in the special dictionary. And key search is implemented by method of PRS embedding searching that corresponds to generator output according to the dictionary.

Let us carry out an analysis of possibilities and conditions of implementation of attack like «brute force», which is carried out in regard to (20) with an aim of field $(\theta_j)^X (\mathrm{mod}\ p)$ element determining. In a case of (20) for achieving (21) let us observe model of transformation of m-ary symbol into p-ary one.

Let the lengths of symbols in binary representation be l_p, l_{p_1} and l_m correspondingly to modules p, $p1$ and m. Let us define the possibility of guessing through b_i symbol of p-ary symbol, in essence definition of $\theta_j^{K_0+i}$.

Theorem 2. For conditions (20) possibility of correct (guessing) transformation of P_{CT} m-ary b_i symbol into p-ary $\theta_j^{K_0+i}$ is determined with correlation:

$$P_{\mathrm{CT}} = 2^{l_m - l_p} , \quad (23)$$

where l_p and l_m – binary representation of lengths of symbols p and m.

Let us observe theorem 2 proving. When the length of m-ary b_i symbol in binary representation is l_m, the number of his possible modes is defined as 2^{l_m}. During transformation according to module p_1 the length of symbol in binary representation will be l_{p_1} and the number of possible modes will be defined as $2^{l_{p_1}}$. Where degree of alphabet extension can be assessed as

$$\mu_2 = 2^{l_{p_1}} \Big/ 2^{l_m} = 2^{l_{p_1} - l_m} .$$

During transformation according to module p the length of symbol in binary representation will be l_p, and the number of possible modes will be defined as 2^{l_p}. Degree of alphabet extension during switching to transformation according to p will be:

$$\mu_1 = 2^{l_p} \Big/ 2^{l_{p_1}} = 2^{l_p - l_{p_1}} .$$

Correspondingly the possibility of guessing an alphabet symbol according to module p_1 is defined as

$$P_{p_1} = 1/\mu_2 = 2^{l_m - l_{p_1}}. \quad (24)$$

The possibility of guessing an alphabet symbol according to module p is defined as

$$P_p = 1/\mu_1 = 2^{l_{p_1} - l_p}. \quad (25)$$

Thus theorem is proved. The general possibility of guessing an alphabet P_{G} symbol according to module p during switching from m-ary source to p-ary will be defined with multiplication of events P_{p_1} (24) and P_p (25), i.e.:

$$P_{\text{G}} = P_{p_1} \cdot P_p = 2^{l_m - l_{p_1}} \cdot 2^{l_{p_1} - l_p} = 2^{l_m - l_p}. \quad (26)$$

Using (26) the one can define complexity I_{G} of one alphabet symbol according to module p during switching from m-ary source to p-ary as

$$I_{\text{G}} = 1/P_{\text{в}} = 2^{l_p - l_m}.$$

Thus while applying of generator scheme without hashing the complexity I_{KR} of key reconstruction $X = K_0 + i$ is determined with a formula:

$$I_{\text{KR}} = I_{\text{G}} \cdot I_{\text{DL}} = 2^{l_p - l_m} \cdot \exp\left(\varepsilon \ln(p)^{\upsilon} \ln\ln(p)^{(1-\upsilon)}\right). \quad (27)$$

For a case of applying of generator schemes with guessing a field element, discrete logarithm solution and hashing the complexity I_{KRH} of key reconstruction $X = K_0 + i$ is determined with a formula:

$$I_{\text{KRH}} = I_{\text{G}} \cdot I_{\text{DL}} \cdot I_{\text{H}} = 2^{l_p - l_m} \cdot \exp\left(\varepsilon \ln(p)^{\upsilon} \ln\ln(p)^{(1-\upsilon)}\right) \cdot 2^{\text{n}/2}. \quad (28)$$

It is necessary to point out that formulae (27) and (28) received for a case, when PRS is produced by mean of applying only one m-ary symbol. If for producing of PRS μ of m-ary symbols is used and value of i is getting bigger according to a known rule, then (27) and (28) can be applied to assessment of cryptographic resistibility of offered PRS generator. If i is getting bigger according to an unknown rule, then besides it is necessary to solve a task of determination of rule of its changing. But as we consider that cryptanalyst knows the rule of i changing, (27) and (28) are recommended for assessment of PRS generators inversion complexity of a type that is observed. In the Table 1 are given assessments of PRS generator inversion complexity according to (27) and (28). Data analysis of Table 1 allows making a conclusion that PRS generator inversion complexity has an exponential character and it is bigger than complexity of «brutal force» method.

Table 1 – Complexity of generator inversion

Method		p, p_1, m				
		$2^{256}, 2^{128}, 2^{8}$	$2^{256}, 2^{128}, 2^{64}$	$2^{512}, 2^{256}, 2^{8}$	$2^{1024}, 2^{512}, 2^{256}$	$2^{2048}, 2^{1024}, 2^{512}$
I_{KR}		$5.0543 \cdot 10^{089}$	$7.0143 \cdot 10^{072}$	$2.1618 \cdot 10^{172}$	$1.4827 \cdot 10^{259}$	$1.1867 \cdot 10^{500}$
I_{KRH}	n					
	160	$6.1103 \cdot 10^{113}$	$8.4798 \cdot 10^{096}$	$2.6135 \cdot 10^{196}$	$1.7925 \cdot 10^{283}$	$1.4346 \cdot 10^{524}$
	256	$1.7199 \cdot 10^{128}$	$2.3868 \cdot 10^{111}$	$7.3562 \cdot 10^{210}$	$5.0453 \cdot 10^{297}$	$4.0381 \cdot 10^{538}$
	384	$3.1726 \cdot 10^{147}$	$4.4029 \cdot 10^{130}$	$1.3570 \cdot 10^{230}$	$9.3071 \cdot 10^{316}$	$7.4490 \cdot 10^{557}$
	512	$5.8524 \cdot 10^{166}$	$8.1219 \cdot 10^{149}$	$2.5031 \cdot 10^{249}$	$1.7168 \cdot 10^{336}$	$1.3741 \cdot 10^{577}$

Let us observe one more way to solve a task of PRS generator inversion of the form (20), which is based on residue classes. For this aim let us give (20) of the following form:

$$\begin{aligned} & b_i = \Theta^{x_i} \pmod P \pmod{P_1} \pmod m, \\ & \Theta^{x_i} \pmod P \pmod{P_1} = q_i \cdot m + b_i,\ 0 \le q_i \cdot m + b_i < P_1, \\ & \Theta^{x_i} \pmod P = l_i \cdot P_1 + q_i \cdot m + b_i,\ 0 \le l_i \cdot P_1 + q_i \cdot m + b_i < P, \\ & \Theta^{x_i} \pmod P = l_i \cdot P_1 + q_i \cdot m + b_i \pmod P. \end{aligned} \tag{29}$$

Direct analysis (29) is showing that x_i, l_i, q_i are unknown and should be determined. Now let us take into account that rule of changing of x_i is known. On the basis of (29) it is possible to make equation system of the following form:

$$\begin{cases} \Theta^{x_i} \pmod P = l_i \cdot P_1 + q_i \cdot m + b_i \pmod P; \\ \Theta^{x_{i+1}} \pmod P = l_{i+1} \cdot P_1 + q_{i+1} \cdot m + b_{i+1} \pmod P; \\ \ldots\ldots\ldots\ldots \\ \Theta^{x_{i+k}} \pmod P = l_{i+k} \cdot P_1 + q_{i+k} \cdot m + b_{i+k} \pmod P. \end{cases} \tag{30}$$

The equation system (29) analysis is showing that each new equation in the system adds 2 variables, but there exists linear dependence between x_i and x_{i+1} etc. On the whole in a system of k division there will be $2k+1$ variables, even if we consider that only x_i is variable.

Thus an equation system of the form of (30) with $2k+1$ variables has no solution. Also it should be pointed out that by analogy with three modulo transformation, during multimodulo transformation every new additional modulo transformation adds two variables.

Thus properties of inconvertibility of PRS generator in essence are connected with solving of discrete logarithm equations, e.g. for three modulo transformations of the form of (6) as to i and $K_0 + i$.

For a successful cryptanalysis of generator, firstly, it is needed to solve a discrete logarithm equation and find element – operand. First of all in this case operand of correspondent element of A_i field should be found, and then a discrete logarithm equation with complexity I_{DL} should be solved.

For condition (20) a possibility of correct transformation (of guessing) of P_{CT} of m-ary b_i symbol into p-ary $\theta_j^{K_0+i}$ is determined with correlation (23).

The equation system analysis (29) is showing, that every new equation in the system adds 2 variables; in addition to this there exists linear dependence between x_i and x_{i+1} etc. In a system of k-division there will be $2k+1$ variables. That is why an equation system of the form of (30) with $2k+1$ variables has no solution.

4 Investigation of distinguishing properties of PRS generated on the basis of multimodulo transformations

Applying of PRS on the basis of multimodulo transformations in the finite fields $GF(p)$ and $GF(p^n)$ is possible only under condition of providing good distinguishing properties. Where by distinguishing is meant degree of resembling of PRS to physically random sequence. The main requirements to such sequences from the point of distinguishing are given in [4,6,7].

Below are given the results of assessments in regard to properties of distinguishing of PRS generation based on multimodulo transformations in finite Galois fields $GF(p)$ and $GF(p^n)$, which output values are hashed.

The four types of PRSG are considered. The first one is PRSG in the field $GF(p)$ without hashing; the second one is PRSG in $GF(p)$ with hashing, the third one is PRSG in the field $GF(p^n)$ without hashing, the fourth one is PRSG in $GF(p^n)$ with hashing according to [13,14].

4.1 PRSG with multimodulo transformation in the field $GF(p)$

Data used during PRSG implementation is given below. On the whole there were implemented 10 PRSGs with different output options (Table 2).

PRSG options without hashing.

The value of the first module p with the size of 1024 bytes was chosen from ISO/IEC 9796-3 standard [15], besides it was the same for all implementations:

p=ffffffffffffffffc90fdaa22168c234c4c6628b80dc1cd129024e088a67cc74020bbea63b139b22514a0
8798e3404ddef9519b3cd3a431b302b0a6df25f14374fe1356d6d51c245e485b576625e7ec6f44c42e9
a637ed6b0bff5cb6f406b7edee386bfb5a899fa5ae9f24117c4b1fe649286651ece65381ffffffffffffffff.

The value of the second module p_1 (160 bytes) was also chosen from ISO/IEC 9796-3 standard [15], the same for all implementations:

p_1 =ffd5d55fa9934410d3eb8bc04648779f13174945.

The value of the third module was chosen the same for all implementations, i.e. the alphabet basis $m = 2$.

The value of primary element θ (1023 bytes) was chosen from ISO/IEC 9796-3 standard [15], the same for all implementations:

θ =7fffffffffffffffe487ed5110b4611a62633145c06e0e68948127044533e63a0105df531d89cd912
8a5043cc71a026ef7ca8cd9e69d218d98158536f92f8a1ba7f09ab6b6a8e122f242dabb312f3f637a2
62174d31bf6b585ffae5b7a035bf6f71c35fdad44cfd2d74f9208be258ff324943328f67329c0fffffffffff
fffff.

The value of generator k key for all implementations was generated at random under condition that $k = 1 \div p - 1$. The values of PRSG options are given in the table 2.

Table 2 – PRSG options in $GF(p)$ used during testing

PRSG implementation	Size p, bytes	Size p_1, bytes	Value m	Size θ, bytes	k, 128 bytes
1	1024	160	2	1023	e6894898f9976ba42761f201cc2ff016
2	1024	160	2	1023	84b1c668a99815a269eb15fc87315efc
3	1024	160	2	1023	f4bf155fa99f25a259ebf5f1f73f5ef1
4	1024	160	2	1023	44b4554a541473419942eb45a2595e41
5-SHA-1 (3)	1024	160	–	1023	f4bf155fa99f25a259ebf5f1f73f5ef1
6-SHA-1 (4)	1024	160	–	1023	44b4554a541473419942eb45a2595e41
7-SHA-256 (1)	1024	160	–	1023	e6894898f9976ba42761f201cc2ff016
8-SHA-384(1)	1024	160	–	1023	e6894898f9976ba42761f201cc2ff016
9-SHA-384 (2)	1024	160	–	1023	84b1c668a99815a269eb15fc87315efc
10-SHA-512 (1)	1024	160	–	1023	e6894898f9976ba42761f201cc2ff016

The results of experimental research of theses generators are given in the tables 3 and 4.

4.2 PRSG with transformations in the field $GF(p^n)$

The research of such a PRSG was carried out without hashing. On the whole 5 PRSGs with different output options were implemented.

The value of the first module $f_1(x)$ was chosen from DSTU 4145 [16] the same for all implementations:

$$f_1(x)=x431+x5+x3+x+1.$$

The value of the second module $f_2(x)$ was chosen from DSTU 4145 the same for all implementations:

$$f_2(x)=x163+x7+x6+x3+x+1.$$

The value of the third module $f_3(x)$ was chosen from DSTU 4145 the same for all implementations:

$$f_3(x)=28.$$

The value of primary element θ was chosen from DSTU 4145 the same for all implementations:

θ = x425 + x424 + x423 + x422 + x419 + x418 + x417 + x412 + x406 + x403 + x400 + x395 + x394 + x393 + x392 + x390 + x389 + x387 + x385 + x382 + x381 + x380 + x375 + x371 + x370 + x369 + x368 + x367 + x366 + x361 + x358 + x357 + x355 + x354 + x352 + x351 + x350 + x349 + x348 + x347 + x346 + x345 + x343 + x339 + x338 + x333 + x332 + x331 + x330 + x328 + x325 + x322 + x321 + x320 + x319 + x318 + x314 + x311 + x310 + x309 + x308 + x307 + x304 + x302 + x299 + x298 + x297 + x294 + x293 + x291 + x288 + x280 + x277 + x276 + x274 + x271 + x270 + x268 + x266 + x264 + x263 + x261 + x260 + x259 + x258 + x257 + x256 + x255 + x254 + x253 + x252 + x251 + x248 + x247 + x243 + x239 + x238 + x236 + x235 + x231 + x230 + x228 + x225 + x223 + x219 + x217 + x215 + x213 + x211 + x210 + x209 + x207 + x205 + x203 + x202 + x201 + x199 + x198 + x196 + x195 + x194 + x193 + x191 + x188 + x186 + x185 + x184 + x182 + x180 + x179 + x176 + x173 + x172 + x170 + x169 + x167 + x166 + x162 + x161 + x158 + x157 + x155 + x153 + x152 + x151 + x149 + x147 + x146 + x142 + x140 + x137 + x136 + x134 + x133 + x131 + x129 + x128 + x124 + x123 + x119 + x117 + x115 + x114 + x113 + x109 + x107 + x106 + x104 + x103 + x102 + x97 + x96 + x92 + x89 + x87 + x86 + x83 + x81 + x78 + x75 + x72 + x69 + x68 + x64 + x60 + x58 + x57 + x56 + x55 + x54 + x52 + x51 + x49 + x47 + x45 + x42 + x38 + x37 + x35 + x32 + x31 + x30 + x26 + x25 + x22 + x15 + x14 + x11 + x9 + x7 + x6 + x5 + x4 + x + 1.

The value of generator k key was generated at random under condition that $k = 1 \div p^n - 1$.

1 – DRNG in $GF(p^n)$: k =x207 + x206 + x205 + x204 + x203 + x202 + x201 + x200 + x199 + x198 + x197 + x196 + x195 + x194 + x193 + x192 + x187 + x186 + x185 + x183 + x182 + x181 + x179 + x177 + x174 + x173 + x172 + x171 + x165 + x160 + x129 + x128 + x122 + x120 + x119 + x117 + x116 + x115 + x114 + x113 + x112 + x111 + x109 + x104 + x101 + x98 + x82 + x81 + x80 + x77 + x75 + x74 + x73 + x72 + x71 + x70 + x69 + x68 + x67 + x65 + x64 + x4 + x2 + 1;

2 – DRNG in $GF(p^n)$: k =0xFFFF 0EEA7821 00000003 05bfa124 00072FFB 00000007 00000015;

3 – DRNG in $GF(p^n)$: k =0x151 1596FBBC 47F9B44C ADBC8541 9841BACD FF841632 001F589F 0EEA7821 0034814F 05BFA124 02F846FB 07894ABC 05519584;

4 – DRNG in $GF(p^n)$: k =0x3CE 10490F6A 708FC26D FE8C3D27 C4F94E69 0134D5BF F988D8D2 8AAEAEDE 975936C6 6BAC536B 18AE2DC3 12CA4931 17DAA469 C640CAF3;

5 – the implementation 2 of DRNG in $GF(p^n)$ with hashing with the help of SHA-384.

The data of experimental research of theses generators are given in the tables 3 and 4.

For testing of developed PRSG was applied NIST STS method, recommended by the National Institute of Standards and Technology USA [6]. NIST STS packet consists of 16 static tests. These

tests are applied for checking of the hypothesis about randomness of binary arrays of undefined length have generated by RSG or PRSG. Taking into consideration the results of all the tests the decision about, whether array of zeros and units will be set at random or not, is made.

With application of NIST STS method was carried out a testing of pseudorandom sequences and also properties comparison of these sequences with properties of PRS generator of pseudorandom bytes BBS (test sample, recommended by NIST). The data about PRS tests pass according to the rule 1 [6] is given in the table 3. And the data about BBS generator was taken for reference.

Table 3 – The results of PRS testing on distinguishing according to the rule 1

Generator	Tests quantity, which passed more than 99% arrays	Tests quantity, which passed more than 96% arrays
BBS	134 (70,8%)	189 (100%)
1 - DRNG $GF(p)$	136 (71,95%)	189 (100%)
2 - DRNG $GF(p)$	124 (65,6%)	189 (100%)
3 - DRNG $GF(p)$	140 (74,07%)	187 (98,94%)
4 - DRNG $GF(p)$	130 (68,78%)	187 (98,94%)
5 - SHA-1 (3)	128 (67,72%)	189 (100%)
6 - SHA-1 (4)	129 (68,25%)	189 (100%)
7 - SHA-256 (1)	129 (68,25%)	189 (100%)
8 - SHA-384 (1)	143 (75,66%)	189 (100%)
9 - SHA-384 (2)	130 (68,78%)	188 (100%)
10 - SHA-512 (1)	122 (64,55%)	189 (100%)
1 - DRNG $GF(p^n)$	138 (73%)	189 (100%)
2 - DRNG $GF(p^n)$	132 (69,84%)	187 (98,94%)
3 - DRNG $GF(p^n)$	126 (66,67%)	189 (100%)
4- DRNG $GF(p^n)$	134 (70,8%)	187 (98,94%)
5-SHA-384 2- DRNG $GF(p^n)$	139 (73,5%)	189 (100%)

In the table 4 is given the summary data about tests passes by generators according to the rule 2 [6].

Table 4 – The results of PRS testing on distinguishing according to the rule 2

Generator	Tests quantity, in which possibility value is $P \le 0{,}01$	Tests quantity, in which possibility value is $P \le 0{,}001$
1	**2**	**3**
BBS	0	0
1 - DRNG $GF(p)$	4	0
2 - DRNG $GF(p)$	3	0
3 - DRNG $GF(p)$	4	0
4 - DRNG $GF(p)$	0	0
5 - SHA-1 (3)	2	0
6 - SHA-1 (4)	2	0
7 - SHA-256 (1)	2	0
8 - SHA-380 (1)	1	0
9 - SHA-380 (2)	0	0
10 - SHA-512 (1)	2	0

Continuation of Table 4

1	2	3
1 - DRNG $GF(p^n)$	0	0
2 - DRNG $GF(p^n)$	4	0
3 - DRNG $GF(p^n)$	2	0
4 - DRNG $GF(p^n)$	1	0
5-SHA-384 2- DRNG $GF(p^n)$	1	0

In the fig. 2 and 3 as examples are given phase portraits of distinguishing received with the application of NIST STS [6] test method. Their analysis allows making a conclusion about high quality of distinguishing (randomness).

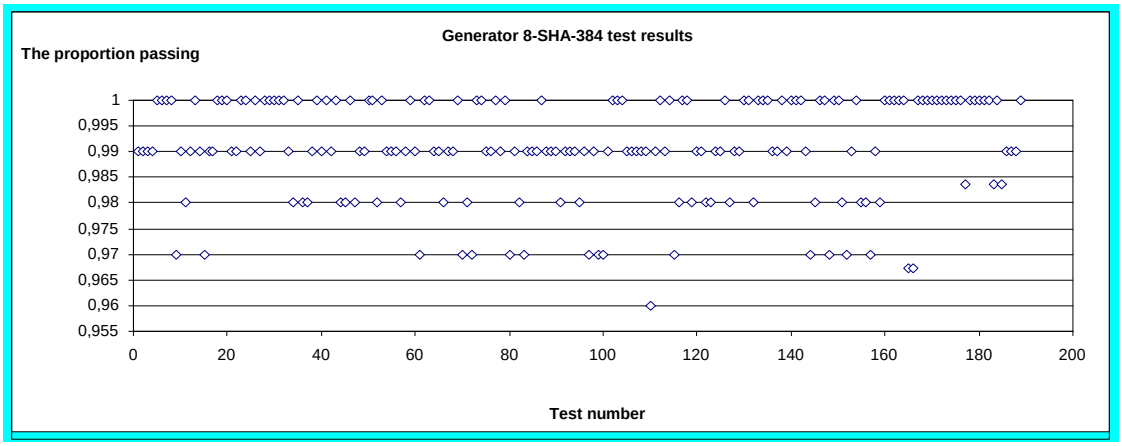

Fig. 2 – The results of experimental research of DRNG 8-SHA-384

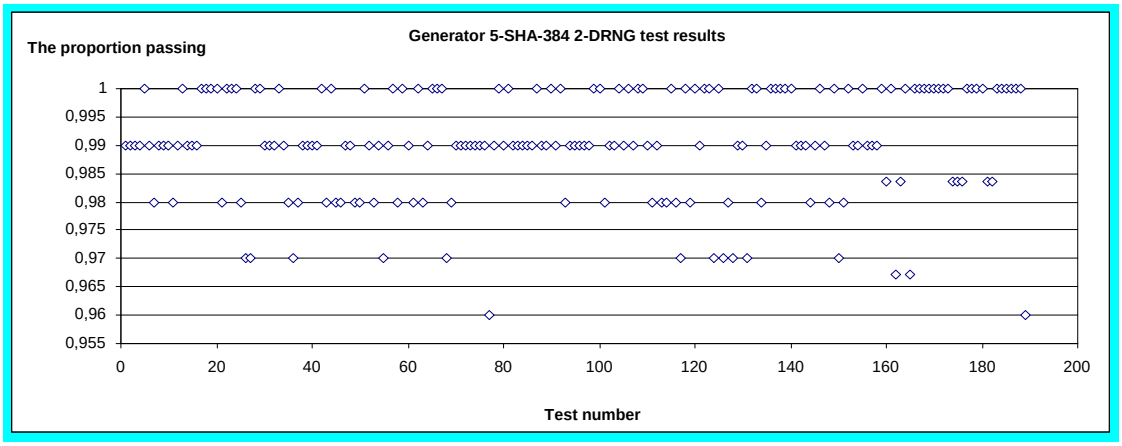

Fig. 3 – The results of experimental research of DRNG 5-SHA-384 2-DRNG $GF(p^n)$

The PRS analysis was carried out according to rating K1 – K4 AIS 20 [4] requirements, which are summarized in the table 5.

Also let us point out that ratings are hierarchically dependent, i.e. each following rating completely includes the previous one and adds its new requirements. Above-given results of research allow making a conclusion that PRS of multimodulo transformations can be applied almost for most of the cryptographic applications. Restrictions can occur only because of complexity of transformation (performance).

Aforementioned requirements are setting up all the level of security from the lowest (an application of DRNG as a counter) to the highest (analyst even knowing certain internal states of generator cannot compromise the whole array).

Table 5 – Comparison of functional ratings К1–К4

Functional rating	Requirements to DRNG	Cryptographic systems, which DRNG of this rating is applied for
К1	К1(i)	Interactive protocols
К2	К1(i) + К2(ii)	Steam ciphers
К3	К1(i) + К2(ii) + К3(iii) + К3(iv)	Key generation, Generation of digital DSS (secret key x or random number k), Password generation
К4	К1(i) + К2(ii) + К3(iii) + К3(iv) + К4(v)	Key generation, Generation of digital DSS (secret key x or random number k), Session key for symmetric cryptographic mechanisms, Password generation

Besides AIS 20 testing method can be applied either in actual time, during the process of research or technological testing.

5 Conclusions

Currently a number of methods and on its basis means of PRS formation have been developed. Their peculiarity includes the fact that they are built as a rule for binary basis $m = 2$. The aim to develop PRS generation methods and means with necessary properties of probability and undefined (certain) alphabet basis is important and necessary. From our point of view the rating of multimodulo transformations should be called the most promising among ratings of such transformations.

Determined PRS generator, which is functioning according to three modulo transformation on the basis (11) or (12) under conditions (2)–(8) fulfillment, provides generation of PRS (symbols) numbers with certain basis of m alphabet, repetition period $p^n - 1$, equally possible appearance of symbols at the repetition period $p^n - 1$ and ensemble of isomorphisms $\varphi\left(p^n - 1\right)$.

On the whole the method and directly PRS generator based on multimodulo transformation can be applied in a number of cryptographic and other applications, in which are set conditions of the high equiprobability and the necessity of undefined basis of PRS symbols appearance.

For through study of PRS generation complexity additional studies are needed. As rough assessments can be used the ones given in [10] concerning the complexity of cryptographic transformations in the finite field $GF\left(p^n\right)$.

References

[1] Methods and means of pseudorandom sequences generation /Y. Gorbenko, T. Grinenko, N. Shapochka, A. Neyvanov, R. Mordvinov// Applied Radio Electronics . - 2011. - Vol. 10. - №2. - P. 141-152. (in Ukrainian).

[2] Potiy A.V. Method of multimodulo transformation of numbers / A.V. Potiy // Information processing and control of managing systems reliability: collection of the science papers. – Kh., 1997. – P.63-68. (in Ukrainian).

[3] Grinenko T.O. Properties of determined random sequences generated on the basis is of multimodulo transformation in Galois fields/ T.O. Grinenko, Y.I. Gorbenko // Collection of the science papers of Kharkiv University of the Science Force. – 2011. – Pub. 1(27). – P.136–139. (in Ukrainian).

[4] Application Notes and Interpretation of the Scheme (AIS) 20. Functionality classes and evaluation methodology for Deterministic random number generators. Certification body of the BSI in context of certification scheme. BSI, 1999.

[5] Information technology. Security techniques. Random bit generation: ISO/IEC 18031. - 2005.

[6] Potiy A.V. Static testing of random and pseudorandom numbers generators with the use of NIST STS static test collection/ A.V.Potiy, S.Y.Orlova, T.A Grinenko // Legal, regulatory and metrological support of information security in Ukraine. – 2001. – Pub. 2. – P.206–214. (in Ukranian).

[7] NIST SP 800-22. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. April 2000. http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.

[8] Vinogradov I.M . Main theories of numbers/ I.M Vinogradov. – M.: Science, 1981. – 177 p. (in Russian).

[9] Lidl R. The finite fields: In 2 vol. / R. Lidl, G. Niderrayter. – M.: Mir, 1988. – Vol.2. – 822 p.(in Russian).

[10] Gorbenko Y.I. Methods of pseudorandom sequences generator assessment based on multimodulo transformations in the finite fields/ Y.I. Gorbenko // Radio technique: All-Ukranian. Mezhved. Scien-Tech. Col. – 2011. – Pub. 165. – P.249-253. (in Russian).

[11] Shnayer B. Applied Cryptography. Protocols, algorithms, reference texts in SI language/ B. Shnayer. – M.: Triumph, 2002. – 797 p. (in Russian).

[12] Stollings V. Cryptography and nets security / V. Stollings. – M. :Wiliams, 2001. – 669 p. (in Russian).
[13] Information technology. Security techniques. Hash-functions. Part 2. Hash-functions using an n-bit block cipher: ISO/IEC 10118-2.
[14] Information technology. Security techniques. Hash-functions. Part 3. Dedicated hash-functions: ISO/IEC 10118-3.
[15] Information technology. Security techniques. Digital signature schemes giving message recovery. Part 3. Discrete logarithm based mechanisms: ISO/IEC 9796-3:2006.
[16] Information technologies. Cryptographic information security. Digital signature based on elliptic curves. Forming and Checking: DSTU (State Standards of Ukraine) 4145-2002.

Рецензент: Віктор Краснобаєв, д.т.н., проф., Харківський національний університет імені В.Н. Каразіна, Харків, Україна. E-mail: krasnobayev@karazin.ua

Надійшло: січень 2016.

Автори:
Юрій Горбенко, к.т.н., Харківський національний університет імені В.Н. Каразіна, Харків, Україна.
Email: YuGorbenko@iit.kharkov.ua
Тетяна Гріненко, к.т.н., доцент, Харківський національний університет радіоелектроніки, Харків, Україна.
Email: t_lame@mail.ru
Олексій Нарєжній, к.т.н., Харківський національний університет імені В.Н. Каразіна, Харків, Україна.
Email: narlexa69@mail.ru
Микола Карпінський д.т.н., проф., університет Бельсько-Бяла, Польща. E-mail: mkarpinski@ath.bielsko.pl

Генератор псевдовипадкових послідовностей на основі багатомодульних перетворень.
Анотація. Викладаються основні теоретичні положення та практичні результати дослідження методу генерування псевдовипадкових послідовностей з довільним алфавітом на основі багатомодульних перетворень в скінченному полі $GF(p^n)$, наводяться результати аналізу властивостей нерозрізнюваності, непередбачуваності, необоротності, періодів повторення та складності (швидкодії).

Ключові слова: псевдовипадкова послідовність, генератор псевдовипадкової послідовності, багатомодульне перетворення, поле Галуа, нерозрізнюваність, непередбачуваність, необоротність, період повторення.

Рецензент: Виктор Краснобаев, д.т.н., проф., Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: krasnobayev@karazin.ua

Поступила: январь 2016.

Авторы:
Ю. Горбенко, к.т.н., Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина.
Email: YuGorbenko@iit.kharkov.ua
Т. Гриненко, к.т.н., доцент, Харьковский национальный университет радиоэлектроники, Харьков, Украина.
Email: t_lame@mail.ru
А. Нарежний, к.т.н., Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина.
Email: narlexa69@mail.ru
Николай Карпинский д.т.н., проф., университет Бельсько-Бяла, Польша. E-mail: mkarpinski@ath.bielsko.pl

Генератор псевдослучайных последовательностей на основе многомодульных преобразований.
Аннотация. Излагаются основные теоретические положения и практические результаты исследования метода генерации псевдослучайных последовательностей с произвольным алфавитом на основе многомодульных преобразований в конечном поле $GF(p^n)$, приводятся результаты анализа свойств неотличимости, непредсказуемости, необратимости, периодов повторения и сложности (скорости).

Ключевые слова: псевдослучайная последовательность, генератор псевдослучайной последовательности, многомодульное преобразование, поле Галуа, неотличимость, непредсказуемость, необратимость, период повторения.