Quasigroups and Related Systems, **4** (1997), 89–102

NLPN Sequences over $GF(q)$

Czesław Kościelny

Abstract

PN sequences over $GF(q)$ are unsuitable directly for cryptography because of their strong linear structure. In the paper it is shown that in order to obtain the sequence with the same occurence of elements and with the same length as PN sequence, but having non-linear structure, it simply suffices to *modulate* the PN sequence by its cyclic shift using two-input quasigroup operator. Thus, such new sequences, named NLPN sequences, which means Non-Linear Pseudo-Noise sequences, can be easily generated over $GF(q)$ for $q \geq 3$. The method of generating the NLPN sequences is exhaustively explained by a detailed example concerning non-linear pseudo-noise sequences over $GF(8)$. In the other example the way of constructing good keys generator for generalized stream-ciphers over the alphabet of order 256 is sketched. It is hoped that NLPN sequences will find many applications in such domains as cryptography, Monte-Carlo methods, spread-spectrum communication, GSM systems, random number generators, scrambling, testing VLSI chips and video encryption for pay-TV purposes.

1. Introduction

Non-binary pseudo-random sequences over $GF(q)$ of length $q^m - 1$, called PN sequences have been known for a long time [3,6,7]. Although they are used in many domains of modern technology, they are

1991 Mathematics Subject Classification: 94A55, 94A60, 20N05

Keywords: PN sequences, NLPN sequences, random number generators, cryptographic keys for generalized stream ciphers, finite field arithmetic, fast software encryption, quasigroups, Latin squares.

unsuitable directly for cryptographic applications, mainly because of their strong linear structure. Therefore, several concepts have been proposed in order to demolish this structure (e.g. non-linear filter generators [2,7] and multiplexed sequences [4]), consisting in non-linear *filtering* or *modulating* PN sequences over $GF(2)$.

The presented method concerns sequences over $GF(q)$ for $q \geq 3$ and it uses a quasigroup operators in order to transform PN sequence into a sequence, having much more randomness than the former. Thus the generator of NLPN sequences consists of two identical linear shift registers with feedback, determined by the same primitive polynomial of degree m over $GF(q)$, which are equipped with the possibility of *tuning* the initial states. The method is very simple and it is well adapted for both software and hardware implementations.

2. A Quasigroup-Based Method of Constructing NLPN Sequences over $GF(q)$ and Their Properties

Let

$$\mathbf{a} = a_0 a_1 \cdots a_{q^m-2} \tag{1}$$

be an arbitrary sequence of elements from $GF(q)$, and let

$$R = \begin{bmatrix} a_i & a_{i+1} & \cdots & a_{i+m+c-1} \\ a_{i+1} & a_{i+2} & \cdots & a_{i+m+c} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{i+m+c-1} & a_{i+m+c} & \cdots & a_{2(i+m+c-1)} \end{bmatrix}, \tag{2}$$

be an $(c+m) \times (c+m)$ matrix over $GF(q)$, the rows of which are consecutive elements from the sequence (1). The subscripts i, $0 \leq i \leq q^m - 2$, are taken modulo $q^m - 1$.

Definition: A sequence (1) is called a non-linear PN sequence and further denoted as NLPN sequence, if

$$\exists\, i,\ 0 \leq i \leq q^m - 2,\ \exists\, c \geq 1\ [\det(R)\ \neq 0], \tag{3}$$

and if in the sequence only one element of $GF(q)$ occurs $q^{m-1}-1$ times, while every other element from $GF(q)$ occurs q^{m-1} times.

The presented method stems from the following

Conjecture: Let $q = p^k > 2$, p - prime, k - positive integer ≥ 1 and let $\mathbf{a}$ and $\mathbf{a}^i$ denote a PN sequence of length $q^m - 1$ over $GF(q)$ and its cyclic shift i places to the right, respectively. Then there exist a quasigroup

$$Q = \langle SQ, \bullet \rangle, \tag{4}$$

of order q, viz. $|SQ| = q$, SQ - set of the elements of a quasigroup, represented in the same manner as the elements of $GF(q)$, such that sequences

$$\mathbf{a} \bullet \mathbf{a}^i, \quad \mathbf{a}^i \bullet \mathbf{a} \tag{5}$$

are NLPN sequences, if

$$i \neq 0 \pmod{(q^m - 1)/(q-1)}. \tag{6}$$

It may be supposed that in the case when in the main diagonal of the quasigroup's operation table any element occurs only once, the fulfilment of condition (6) may not be required.

The number of quasigroups, satisfying this conjecture is not yet known, and one would rather expect that it will not be determined in the near future. The experiments show, however, that it is hard to find a true quasigroup, which does not produce NLPN sequences according to the presented method.

The proof of the conjecture is the subject of current work and will be reported in due course.

At present, the author knows only the following properties of NLPN sequences:

Property I – The Number of Occurrences of Elements of $GF(q)$ in an NLPN *sequence*: If 0 denotes the identity element of the additive group of $GF(q)$, then the element equal to $0 \bullet 0$ occurs in the NLPN sequence $q^{m-1} - 1$ times, while the remaining elements of $GF(q)$ occur in this sequence q^{m-1} times.

An algebraic system $\langle SQ, \bullet \rangle$ is called a *quasigroup* if there is a binary operation $\bullet$ defined in SQ and if, when any two elements $a, b \in SQ$ are given, the equations $a \bullet x = b$ and $y \bullet a = b$, each, have exactly one solution [1].

Property II – The Set of All NLPN *Sequences Derived from One PN Sequence and One Quasigroup Q*: Let $\mathcal{S}_{\mathsf{NLPN}}$ denote the set of all different NLPN sequences generated by one PN sequence and one quasigroup. Then

$$k(q^m - 1) \geq |\mathcal{S}_{\mathsf{NLPN}}| \geq k(q^m - q - 1), \tag{7}$$

where $k = 1$ if a quasigroup Q is abelian, and $k = 2$ if it is non-abelian. This number depends on the elements forming the main diagonal of the quasigroup's operation table.

Property III – Autocorrelation function: Each NLPN sequence belonging to $\mathcal{S}_{\mathsf{NLPN}}$ has distinct autocorrelation function resembling the autocorrelation function of the random sequence of elements of $GF(q)$ having the length $q^m - 1$.

3. Example 1

Since the presented method is rather a new one, it will now be exhaustively explained.

Let 8-element finite field $GF(8) = \langle\{0, 1, \ldots, 7\}, +, \cdot\rangle$ be constructed using the polynomial $x^3 + x + 1$. Then the operations $+$ and $\cdot$ can be defined as follows:

$$x + y = x \text{ XOR } y,$$

$$x \cdot y = \begin{cases} 0 \text{ if } x = 0 \text{ or } y = 0, \\ \text{etn}(\text{nte}(x) + \text{nte}(y) \pmod 7) \text{ otherwise.} \end{cases}$$

It is easy to observe that the above representation of $GF(8)$ results from the assumption that α, primitive element of $GF(8)$ and also a root of the polynomial $x^3 + x + 1$ over $GF(2)$, is denoted by 2. Thus, $\alpha^i = \text{etn}(i)$ for $i = 0, 1, \ldots, 6$. The functions $\text{nte}(x)$ and $\text{etn}(x)$, named according to the tasks which they perform (nte - number to exponent of α conversion, etn - exponent of α to number conversion) are defined in Table 1.

The values of nte(0) and etn(7) are not used, therefore, they are not defined.

Table 1: Functions nte(x) and etn(x) used for Multiplying in $GF(8)$

x	0 1 2 3 4 5 6 7
nte(x)	? 0 1 3 2 6 4 5
etn(x)	1 2 4 3 6 7 5 ?

Table 2: Addition and Multiplication Tables in $GF(8)$

+	0 1 2 3 4 5 6 7
0	0 1 2 3 4 5 6 7
1	1 0 3 2 5 4 7 6
2	2 3 0 1 6 7 4 5
3	3 2 1 0 7 6 5 4
4	4 5 6 7 0 1 2 3
5	5 4 7 6 1 0 3 2
6	6 7 4 5 2 3 0 1
7	7 6 5 4 3 2 1 0

·	0 1 2 3 4 5 6 7
0	0 0 0 0 0 0 0 0
1	0 1 2 3 4 5 6 7
2	0 2 4 6 3 1 7 5
3	0 3 6 5 7 4 1 2
4	0 4 3 7 6 2 5 1
5	0 5 1 4 2 7 3 6
6	0 6 7 1 5 3 2 4
7	0 7 5 2 1 6 4 3

Although the operations in $GF(8)$ are simple, the reader can easier follow the presented example by using the tables of addition and multiplication in $GF(8)$, given in Table 2.

Let $\mathbf{s}= s_0 s_1 \cdots s_{62}$ be a PN sequence obtained from the primitive polynomial $x^2 + 2x + 2$ over $GF(8)$. Therefore

$$s_{i+2} = 2s_{i+1} + 2s_i, \;\; i = 0,\ 1,\ \ldots,\ 60.$$

If one specifies the initial values as $s_0 = 1,\ s_1 = 0$, then the whole PN sequence will be

$$\mathbf{s} = \text{concat}(\gamma, \alpha\gamma, \alpha^2\gamma, \alpha^3\gamma, \alpha^4\gamma, \alpha^5\gamma, \alpha^6\gamma) = s_0 s_1 \cdots s_{62}, \qquad (8)$$

where $\gamma = 102476232$ and $\alpha = 2$. Finally $\mathbf{s} =$

$$1024762322043574644036153733067216566075427177051345255012631 41. \qquad (9)$$

Further let

$$\mathbf{S}^i = s_{62-i+1}s_{62-i}\cdots s_0 s_1 \cdots s_i s_{i+1}\cdots s_{62-i}$$

where $i \in \{0,\ 1,\ \ldots,\ 62\}$ and the subscripts are computed modulo 63, denote the PN sequence $\mathbf{S}$ shifted i places to the right. Let also

$$Q = \langle\{0,1,2,3,4,5,6,7\}, \bullet\rangle,$$

be a quasigroup with operation $\bullet$ defined in Table 3.

Table 3: Operation Table in the Quasigroup Q

$\bullet$	0	1	2	3	4	5	6	7
0	4	5	7	1	6	0	2	3
1	3	2	0	6	1	7	5	4
2	5	3	6	7	0	1	4	2
3	0	1	2	3	4	5	6	7
4	6	0	3	5	2	4	7	1
5	1	7	4	2	5	3	0	6
6	7	6	5	4	3	2	1	0
7	2	4	1	0	7	6	3	5

A half of all NLPN sequences $\mathbf{s}(i) = \mathbf{S} \bullet \mathbf{S}^i$, where

$$i \in \{1,\ 2,\ \ldots,\ 62\} \setminus \{9,\ 18,\ 27,\ 36,\ 45,\ 54\},$$

obtained by means of the proposed method from the PN sequence $\mathbf{S}$ in Tables 4 and 5 is presented. One can get the other half of these sequences as $\mathbf{s}(i) = \mathbf{S}^i \bullet \mathbf{S}$ since the quasigroup Q is non-abelian. In this way one quasigroup of order 8 and one primitive polynomial of degree 2 over $GF(8)$ give 110 different NLPN sequences. Taking into account that the number of primitive polynomials of degree 2 over $GF(8)$ equals to 18, that the total number of loops of order 8 is equal to 535,281,401,856 [1], and that there are much more quasigroups of the same order, which are not loops, one may easily appreciate the importance of the proposed method for cryptograhic practice, where q is often of order of a few dozen or of several hundreds.

A *loop* $\langle L,\ +\rangle$ is a quasigroup with an identity element: that is, a quasigroup in which there exists an element $e \in L$ with the property that $e + x = x + e = x$ for every $x \in L$.

Table 4: A Half of the Set $\mathcal{S}_{\mathsf{NLPN}}$ Obtained from the PN Sequence (9) and the Quasigroup defined in Table 3

i	$\mathbf{s}(i)=\mathbf{s} \bullet \mathbf{s}^i$
1	25537042767642613726045750731732060212264014453171551430335660 1
2	15361326677305040276476016371424457122311170543635020340756255 2
3	26302507461321752122433264533405504610306261155640477537717107 6
4	65004764247524224416631133151007316562607557713747633250012023 5
5	51324652022334163543411500610457670055337424677627111726762530 0
6	02707630603722137250735542365202641107404433566441165371115576 2
7	27454331566164035621501047032667442713177135250507062135602274 3
8	35670601357270155366324567401141602472540465323312172610273504 7
10	70501171035625575537036377646704222041201645632145242763316314 0
11	00164543305004711357271255466327563401017212362037456673452162 7
12	77142666736011120731253536742375617243026450430050130465427525 1
13	10646732475217233177350140576170340621525534142356661543220070 6
14	66136251641047465212240407353330172160065725515076712357430734 2
15	21041210362515407323656422430074131415621703357755705636026746 7
16	75726515537737706636710221041250535742435206233426404160160141 3
17	30357265155362426067405431206462115371374756022601734013705721 4
19	43544405610614356205051661327777051362273665011573743023224012
20	23266674160407174025170573237520626316715447051236143032550513 4
21	45212233017451431506163445621515143032756732603260766700547772 0
22	03315711700353205755462127767416332206352505461663602472741044 1
23	17205055376423666371132701472506075423702026340243514645511631 7
24	76614020131254647032361712243117051340557073035367525067242656 4
25	40035627515543142607642514026036654731662471202773127103031557 3
26	33121037250035632465216744107354044576031031721025567212466677 5
28	52516367323657022340060034415710414457255151376166037621340227 7
29	72465202233103453430572460145626100547112764736533712616517005
30	22173725063076246520227111635343355017043576656014624731474051 0
32	05443157107116562056557304261673274302123621063554217070624337 4
33	47376140216377517401420352122440261533345613700616255205770366 5
34	53603246520226410645337456017103167756503710271344750122214765 3
35	32234116453444500160141326505537237677767600126272005115323456
37	61575536142673730013022246250746547365242230017113460056371175 4
38	41763061412706625103777737520223012635413055107434532506154624 6

Table 5: Continuation of Table 4

i	$s \bullet s^i$
39	637521223407615433157033104572652504664766743114002216521073507
40	712507476324602122331041543405613641457705115372026573665030672
41	207104562657503604277646061362117775114503207524623102331434355
42	551554704270633771462026725113667216520723431730033455204014166
43	623400735433106716104547750554710237673200422166525461517236123
44	314064237521273417635306157006007532751053724275463264166104521
46	165536132716661014720075251737636335202734027451041062473045425
47	560601357215005367426743417130212452506106364757322644270533711
48	120224411735313150706136572750557623276363150467203520410674644
49	464277763111322703025151764236523274301345403055216406076650157
50	730773534305720611356250055470424736466441221317110165620752326
51	602233007454360577171170627565534012617331634122240754535642061
52	111171045720525546732653630703122067250546672470612731464453033
53	367453440517130145627520536032764660704221176254530537171212630
55	075202244066304441510144135627511566032307771601227235753637536
56	676674170462023025113756246521227340635143016103313077552554470
57	016323752022410764533430711601354255055661467673700442762372115
58	577317031267452510417461652122343033534615620704756760251360024
59	426515547132657647003062037251645762375712274063054134012061331
60	134305600701407275755442326776315110261602536415724357426331260
61	062714625014753236521266300635447107007413542652153311775764203
62	370135726565567732616672703020135201736532445207644413150441102

For such values of q the number of quasigroups can be expressed as a factorial of astronomical number. Speaking more precisely, the number of quasigroups of order n equals to the number of latin squares of the same order $L(n)$, which, for $n > 10$ satisfies [5]

$$\prod_{k=1}^{n} k!^{\frac{n}{k}} \geq L(n) \geq \frac{n!^{2n}}{n^{n^2}}. \tag{10}$$

At last it may be interesting to see the autocorrelation functions of several NLPN sequences over $GF(8)$ and to compare them with autocorrelation functions of a PN sequence and of a random sequence. Therefore, one period of the autocorrelation function for all PN sequences $\mathbf{s}^i$ in Fig. 1 can be seen.

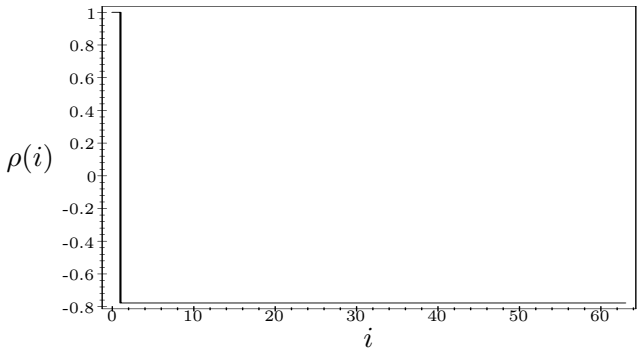

Figure 1: Autocorrelation function of all sequences $\mathbf{s}^i$

This function is of course the same for all PN sequences of length 63 over $GF(8)$, no matter which primitive polynomial of degree 2 over $GF(8)$ has been used.

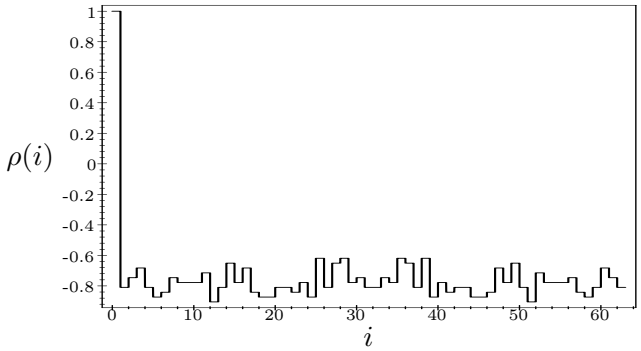

Figure 2: Autocorrelation function of the sequence $\mathbf{s}\bullet\mathbf{s}^{14}$

Figures 2, 3, 4 and 5 show one period of NLPN sequences $\mathbf{s}(i) = \mathbf{s} \bullet \mathbf{s}^i$ for $i = 14,\ 28,\ 42$ and 56, respectively, while one period of the autocorrelation function of truly random sequence of length 63 over $GF(8)$

$$413402332717176544257642215026016410750637616550600040162402102 \tag{11}$$

with the occurrence of elements

$$0-12,\ 1-9,\ 2-9,\ 3-4,\ 4-8,\ 5-6,\ 6-9,\ 7-6$$

in Fig. 6 is presented.

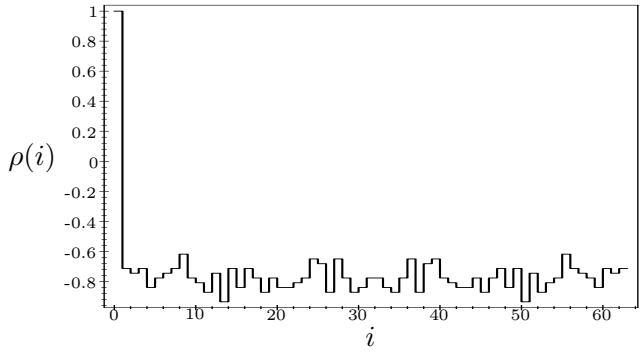

Figure 3: Autocorrelation function of the sequence $\mathbf{S}\bullet\mathbf{S}^{28}$

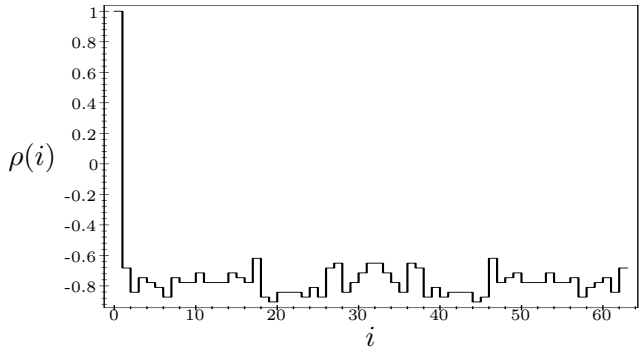

Figure 4: Autocorrelation function of the sequence $\mathbf{S}\bullet\mathbf{S}^{42}$

The autocorrelation function $\rho(i)$ is here defined as follows. Let A be the number of places where the sequence $s_0 s_1 \cdots s_{62}$ and its cyclic shift $s_i s_{i+1} \cdots s_{i-1}$ agree, and D the number of places where they disagree. Then

$$\rho(i) = \frac{A - D}{63}.$$

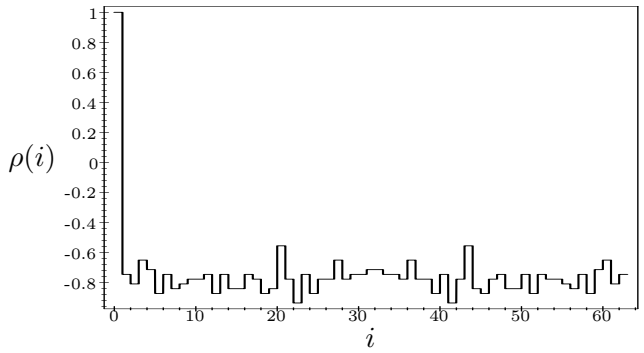

Figure 5: Autocorrelation function of the sequence $\mathbf{s} \bullet \mathbf{s}^{56}$

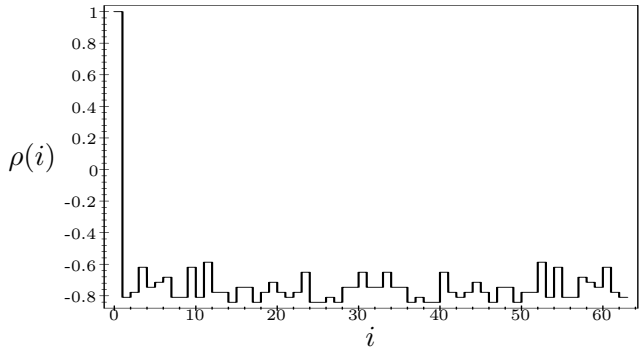

Figure 6: Autocorrelation function of the random sequence (10)

4. Example 2

In the same manner as in Example 1 the Tables of addition and multiplication in $GF(256) = \langle \{0,\ 1,\ \ldots,\ 255\}, +, \cdot \rangle$ were constructed using the primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$ over $GF(2)$. Then the primitive polynomial $P(x) = x^3 + 132x^2 + 152x + 2$ over $GF(256)$ was found and the PN sequence $\mathbf{s}$ of length 16777215 was generated using the following recurrence relation over $GF(256)$

$$s_{i+3} = 132s_{i+2} + 152s_{i+1} + 2s_i, \quad i = 0,\ 1,\ \ldots,\ 16777214.$$

This sequence and its cyclic shift $\mathbf{s}^i$, i satisfying (5), was written to the disk files, say, $F1$ and $F2$. To create a disk file $F3$ containing NLPN sequence of length 16777215, as a quasigroup Q an isotope [1] of the additive group of $GF(256)$ was used. The generated NLPN sequence was then tested by means of the battery of DIEHARD tests of randomness [10], passing them all perfectly. Compared with the length of the NLPN sequence (16777215 Bytes), which may be used as a cryptographic key, to generate it one can use significantly smaller data, namely 65545 Bytes at most (addition table in the quasigroup Q, coefficients of the polynomial $P(x)$, three Bytes of initial condition for the recurrence relation and the number of places in the cyclic shift – about 0.39% of 16777215 Bytes). Since

$$7.53 \cdot 10^{102804} \geq L(256) \geq 3.04 \cdot 10^{101723},$$

it is evident that in a very easy way one can construct simple yet very good generators of cryptographic keys for universal stream-ciphers over the alphabet, containing 256 characters (ASCII code), using NLPN sequences.

5. Conclusions

In the paper only a tiny piece of the iceberg's tip of the possibilities, resulting from the application of quasigroups for generating the sequences of elements of $GF(q)$ with the desired complexity and degree of randomness is presented. E.g. by applying two PN sequences of the same length, but generated by the feedback shift registers, specified by two various primitive polynomials of the same degree, to the inputs of a quasigroup operator, an almost random non-linear sequence will appear on its output with an irregular, but flat distribution of elements and with a high degree of complexity (as, e.g, the sequence (11)). Furthermore, it is possible to combine different quasigroup operators with all linear and non-linear devices and to construct random $GF(q)$-element generators with controlled properties, having many various structures.

The method is especially convenient for fast software encryption. However, it also should be noted that there exists a large class of

quasigroup operators which are easily implemented by means of binary logical circuits, appropriate for the implementation of very secure and extremely fast hardware-oriented quasigroup-based generalized stream-ciphers [6].

References

[1] **J. Dénes, A. D. Keedwell:** *Latin Squares and Their Applications*, Budapest, Akadémiai Kiadó, 1974.

[2] **J. Dj. Golić:** *On the Security of Nonlinear Filters Generators*, in D. Gollman (editor) Fast Software Encryption — Cambridge '96, LNCS, **1039** (1996). 173 – 188.

[3] **S. W. Golomb:** *Shift Register Sequences*, San Francisco: Holden Day, 1967.

[4] **S. M. Jennings:** *A Special Class of Binary Sequences*, PhD thesis, University of London, 1980.

[5] **M. T. Jacobson, P. Matthews:** *Generating Uniformly Distributed Random Latin Squares*, Journal of Combinatorial Designs **4** (1996), 405 – 437.

[6] **C. Kościelny:** *A Method of Constructing Quasigroup-Based Stream-Ciphers*, Applied Math. Comp. Sci. **6** (1996), 109 – 121.

[7] **R. Lidl, H. Niederreiter:** *Introduction to Finite Fields and Their Applications*, Cambridge University Press, 1986, 185 – 245.

[8] **F. J. MacWilliams, N. J. A. Sloane:** *Pseudo-Random Sequences and Arrays*, Proceedings of the IEEE, Vol. 64, NO 12, Dec. 1976, 1715 – 1729.

[9] **R. Rueppel:** *Analysis and Design of Stream Ciphers*, Springer Verlag, Berlin 1986.

[10] `http://stat.fsu.edu/~geo/diehard.html`

Technical University of Zielona Góra
Department of Robotics and Software Engineering
ul. Podgórna 50
65-246 Zielona Góra, Poland
e-mail: C.Koscielny@irio.pz.zgora.pl

or
Higher College of Engineering
ul. Jaworzyńska 151
59-220 Legnica
Poland

Received 15 December, 1997