A MULTICORE COMPUTER SYSTEM FOR DESIGN OF STREAM CIPHERS BASED ON RANDOM FEEDBACK SHIFT REGISTERS

Borislav BEDZHEV[1]
Tihomir TRIFONOV[2]
Nikolay NIKOLOV[3]

Abstract

The stream ciphers are an important tool for providing information security in the present communication and computer networks. Due to this reason our paper describes a multicore computer system for design of stream ciphers based on the so - named random feedback shift registers (*RFSR*s). The interest to this theme is inspired by the following facts. First, the *RFSR*s are a relatively new type of stream ciphers which demonstrate a significant enhancement of the crypto – resistance in a comparison with the classical stream ciphers. Second, the studding of the features of the *RFSR*s is in very initial stage. Third, the theory of the *RFSR*s seems to be very hard, which leads to the necessity *RFSR*s to be explored mainly by the means of computer models.

The paper is organized as follows. First, the basics of the *RFSR*s are recalled. After that, our multicore computer system for design of stream ciphers based on *RFSR*s is presented. Finally, the advantages and possible areas of application of the computer system are discussed.

[1] *University of Shumen "Bishop Konstantin Preslavsky" Faculty of Technical Sciences*
[2] *University of Shumen "Bishop Konstantin Preslavsky" Faculty of Technical Sciences*
[3] *University of Shumen "Bishop Konstantin Preslavsky"Faculty of Technical Sciences.*

Key words: information protecting, stream ciphers, random feedback shift registers, multicore computer system

1. Introduction

The proliferation of computers and communications systems in the 1960s brought with it a demand from the private sector for means to protect information in digital form and to provide security services. The stream ciphers are an important tool for solving of this problem. Despite of their large application, it is very hard or may be impossible to describe all factors, which influence over the performance quality of the stream ciphers. Anyway, surely it depends on their crypto resistance, velocity and effectiveness of hardware implementation. Mostly the crypto resistance of a stream cipher is connected with it ability to generate pseudo random sequence (*PRS* or *gamma*) with following properties:

(P1) it should have enormous period;

(P2) it should demonstrate uniform distribution of d-tuples (for a large range of d);

(P3) it should exhibit a good structure (usually a lattice structure) in high dimensions.

Unfortunately, the mentioned factors are in contradiction, because if the structure of the stream cipher is simple in order to provide high performance velocity and cost-effective hardware implementation, then the crypto reliability is low. For instance, the classical fast and cheap *Linear Feedback Shift Registers (LFSRs)* are vulnerable to the so - named "Berlekamp–Massey crypto attack" [1], [2], [3]. This attack allows finding of all bits of a *LFSR* output sequence, if *2n* its consequent bits are known. Here *n* is the number of the cells connected in the *LFSR*.

Anyway, the advantages of the stream ciphers, based on *LFSR*s, have stimulated the theoretical researches and the practical design of devices with high crypto reliability, which are built by appropriate combined crypto vulnerable, but fast and cheap, elements (including *LFSR*s).

Principally the crypto resistance of stream cipher, based on *LFSR*s, can be enhanced by two alternative methods. The first method uses an appropriate combining of the outputs of several *LFSR*s, as it is shown on Fig.1a. These generators of *PRS*s are called "Combination Generators". The other alternative is to generate the *PRS* as a non-linear function from conditions of the triggers of a single *LFSR* (Fig.1b). These generators of *PRS*s are named "Filter Generators".

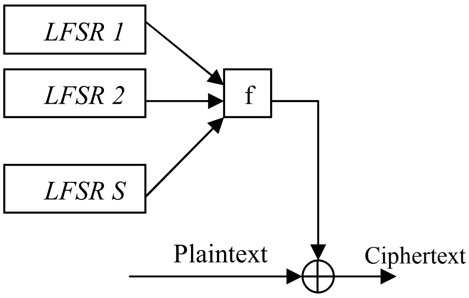

Fig. 1a: *Combination generator*

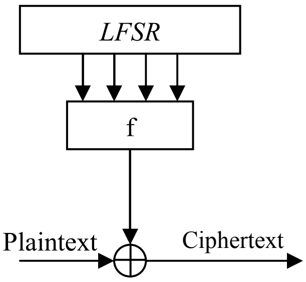

Fig.1b:*Filter-generator*

On Fig. 1 the symbol "$\oplus$" means "summation modulo 2".

In fact the filter generators could be studied as a particular case of the combination generators when $S=1$ on Fig. 1a.

The general structure, shown on Fig. 1a, has been exploited in several new architectures. It should be mentioned the so-named summation generators, shrinking generators, *N*-adic feedback with carry shift registers *(N-FCSR*s) and summation – shrinking generators. They are promising candidates for high-speed encryption applications due to their simplicity and provable properties. Anyway, the permanent advancement in the computers technique makes possible the practical usage of more and more sophisticated attacks to the stream ciphers. With regard to this situation our paper presents a multicore computer system for design of stream ciphers based on random feedback shift registers (*RFSR*s). The *RFSR* is a new architecture, which still has not studied extensively. Today knowledge shows that *RFSR*s it possess very high crypto resistance, obtained by the cost of acceptable complexity.

The paper is organized as follows. First, the basics of the *RFSR*s are recalled. After that, our multicore computer system for design of stream ciphers based on *RFSR*s is presented. Finally, the advantages and possible areas of application of the computer system are discussed.

2. Basics of the random feedback shift registers

The *RFSR*s were proposed by the Russian theorists M. A. Ivanov and I. V. Chugunkov in 2003 [4]. The genesis of the idea could be traced back up to 2001 [5] and even though to 1991 [6]. Despite of the positive features of the *RFSR*s, a brief survey of the papers, focused on this topic, shows that only several resources are available by Internet. This situation could be explained by the short time of exploring of the *RFSR*s and the hardness of their theoretical study.

The *RFSR*s generalize the architecture of the classical stream ciphers, based on *LFSR*s [4]. This fact will be explained by Fig. 2 in more details.

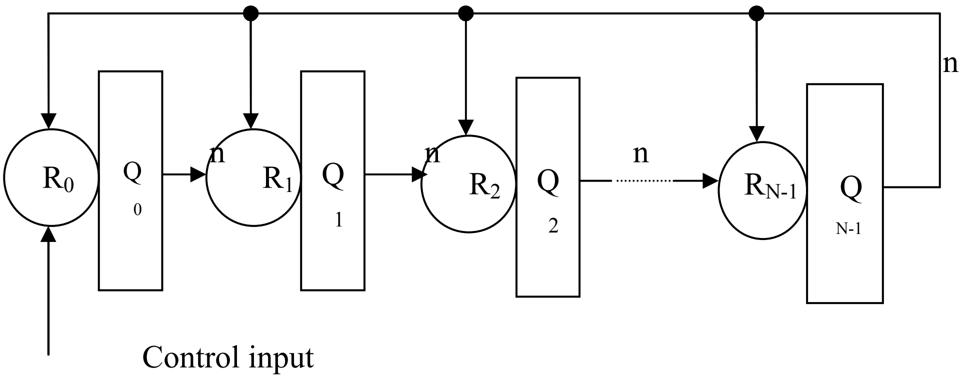

Fig. 2a: *The first type of the general structure of the RFSRs*

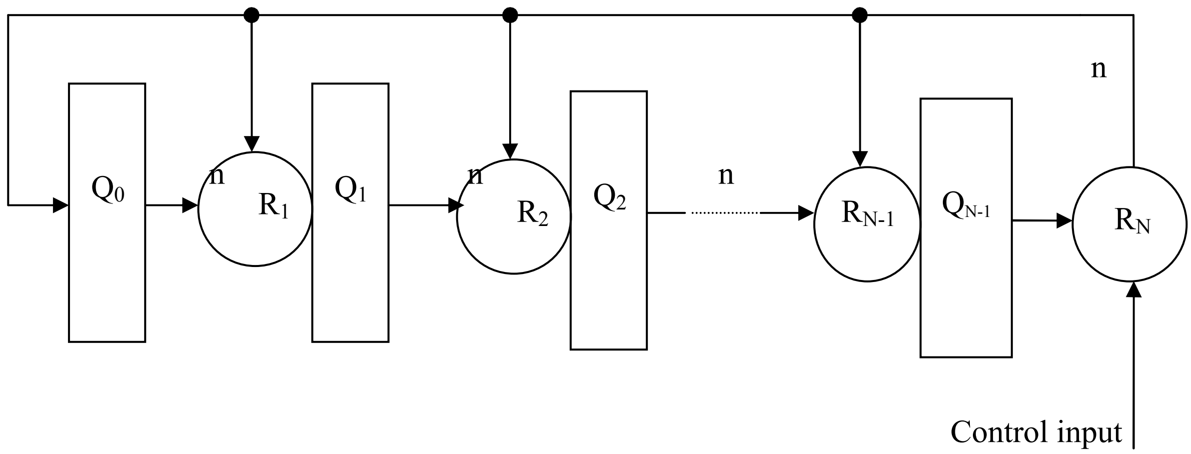

Fig. 2b: *The second type of the general structure of the RFSRs*

On the base of Fig. 2 the following conclusions can be made:

First, two similar types of the general structure of the *RFSR*s are possible. They differ by the moments, when the crypto-graphical transformations are performed.

Second, the both general structures of the *RFSR*s repeat the structure of the classical *LFSR*s, that generate *PRS*s which elements belong to the finite algebraic field (i.e. Galois Field) $GF(p^n)$, where p is an arbitrary prime positive number. From the point of view of the practical implementations the most convenient case is $p = 2$. In this situation on Fig. 2:

- every line (excluding the Control inputs) denotes a *n*-bits parallel data bus;
- the rectangular boxes, numbered as $Q_0, Q_1, ..., Q_{N-1}$, present a *n*-bits register.

Third, despite of the great similarity, a very significant difference exists between *RFSR*s and *LFSR*s. Namely, in the feedback the summation modulo 2 is removed by the so – named Random blocs (*R* blocs).

Forth, as in the situation of *LFSR*s, some of the random blocks can be omitted.

The work process of a *RFSR* can be described as follows.

1) The whole performance of the *RFSR* is controlled by the clock pulses, which are uniformly distributed in the time.

2) During the *j*-th clock cycle the contents of all registers (excepting the most right register Q_{N-1}) are shifted in one position to the right. The content of the most right register Q_{N-1} has two purposes. First, it is output as the *j*-th element γ_j of the produced gamma (*PRS*). Second, it is entered in the most left register Q_0.

It should be stressed on the following peculiarities of the *RFSR*s.
First, during the *j*-th clock cycle the word $c_j(k-1)$, which is the content of the register Q_{k-1}, is transformed by the random block R_k in another *n*-bits word before entering in the register Q_k.

Second, every element of the produced gamma $\gamma_1, \gamma_2, ..., \gamma_j, ...$ consists of *n* bits and from algebraic point of view is an element of the finite field *GF*(2^n).

The transformations, performed by the random blocks, shall be explained by Fig. 3 and Fig. 4.

As shown on Fig. 3, every random block contains two massives of memory, named auxiliary and basic massive respectively. They have 16 words, where every word comprises 4 bits, which are elements of *GF*(2^4). The content of the basic massive is an arbitrary permutation of the sequence of the numbers $0 = 0000,\ 1 = 0001,\ ..., 15 = 1111$. The content of the auxiliary massive is a mirror copy of the basic massive. In other words if the address No *X* in the basic massive is filled by the word *Y*, then the address No *Y* in auxiliary massive is filled by the word *X*. Except this, every random block has two inputs, which are denoted by *A* and *B* respectively. For R_k and *j*-th clock cycle the input *A* enters the content of the register Q_{k-1} and the input *B* enters the word, produced by the feedback during the preceding clock cycle (see Fig. 2).

On Fig. 3 is depicted the case when the input *A* of the random block is the word 0010. It is considered as the address No 2 for the auxiliary massive, which content is $6 = 0110$. This content 6 is summed with the word of input *B*, which is chosen to be 6 on Fig. 3. The ordinary sum $(12 = 1100)$ of the words *A* and *B* is considered as the address No 12 for the basic massive, which content $10 = 1010$ is the output of the random block.

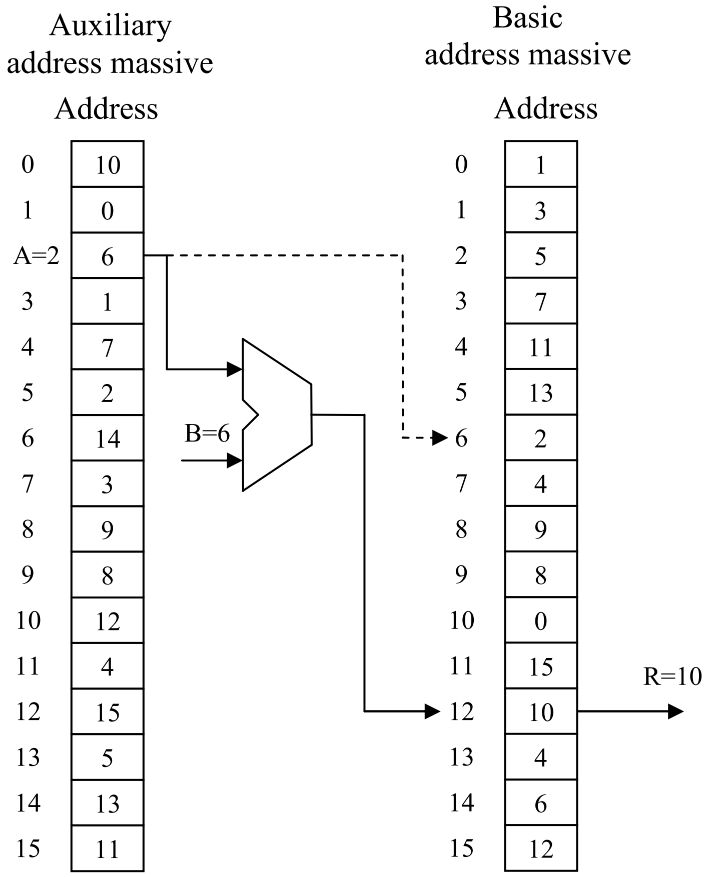

Fig. 3: *The structure of the random blocks*

The performance of the random block can be modified in order to obtain higher crypto – resistance. One possible approach [4] is shown on Fig. 4.

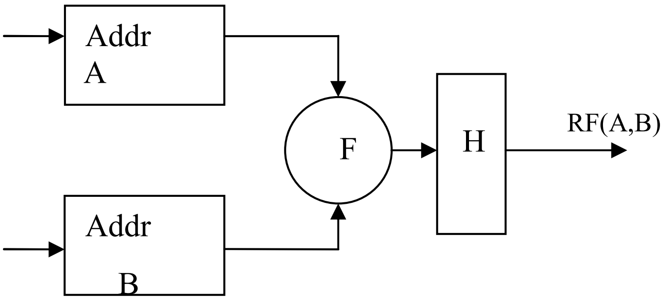

Fig. 4: *A possible modification of the structure of the random blocks*

As depicted on fig. 4, the second word of the input *B* is taken from another auxiliary massive. Except this, the summation of the words *A* and *B* is replaced by a function as: sum modulo 2^n, sum modulo *p* ($p \neq 2^n$), bitwise AND, bitwise XOR and so on. In this case the blocks should be denoted as *RF*-blocks [4].

At the end of this part of the paper, a simple example of the full structure of the *RFSR*s is presented on Fig. 5.

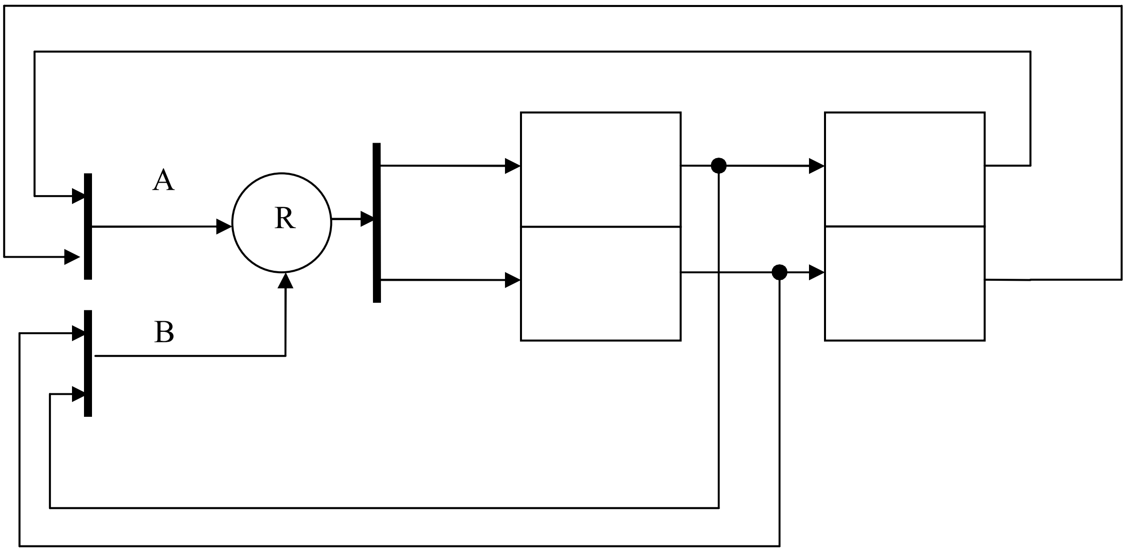

Fig. 5: *The full structure of a simple RFSR with parameters* $n = 2,\ N = 2$

The *PRS*s, which can be generated by the RFSR from Fig. 5, are shown on Fig.6.

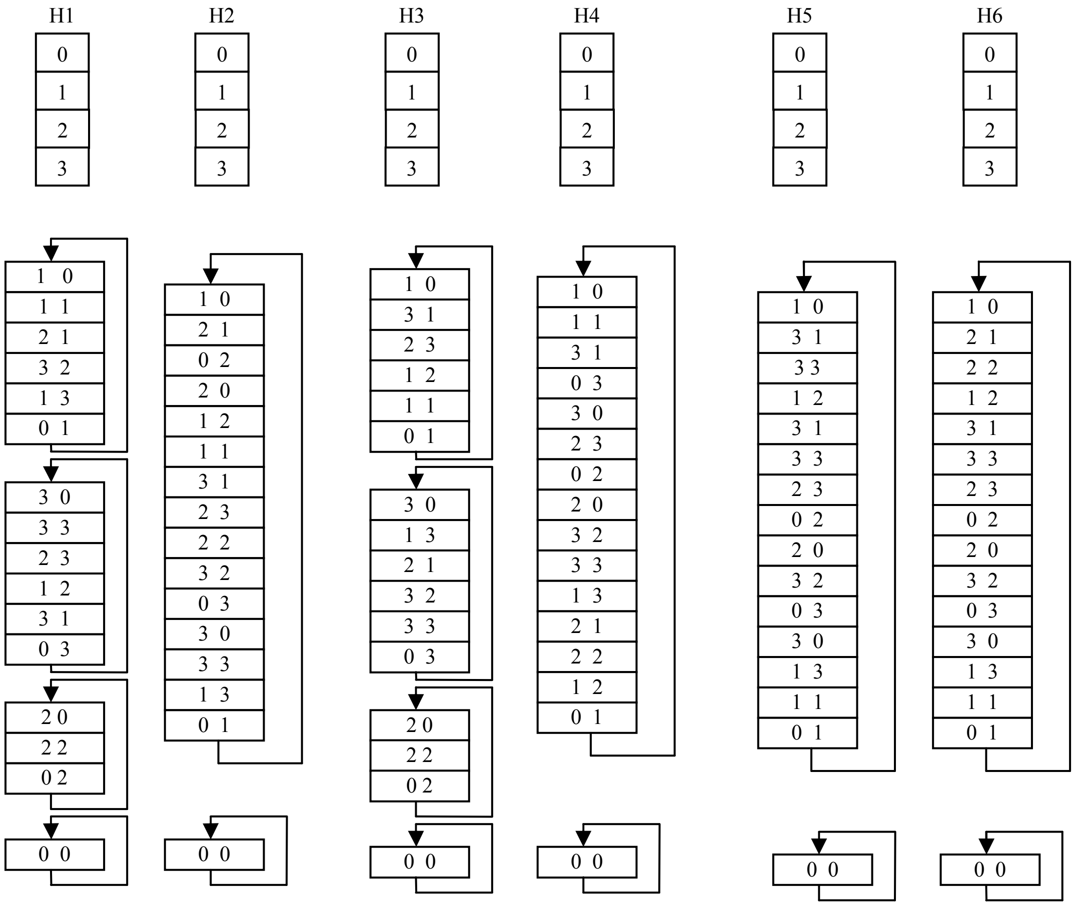

Fig. 6: *The PRSs, generated by the RFSR from Fig. 5*

As the content of an auxiliary massive is a mirror copy of the content of the respective basic massive, on Fig 6 only the possible contents H1, H2, …, H6 of the basic massive are shown.

4. A multicore computer system for design of stream ciphers based on random feedback shift registers

With regard to positive features of the stream ciphers based on *RFSR*s, 10 month ago in the Facuty of Technical sciences of the University of Shumen "Bishop Konstatin Preslavsky" a study of the *RFSR*s began. The most significant part of the study was the building of a multicore computer system (cluster) for modeling and design of *RFSR*s. The main objective of the study is the finding of irreducible polynomials over *GF* (2) with degrees as high as possible and with large number of involved monomials. This objective is inspired by the following reasons. First, the obtained up to the moment experience with *RFSR*s [4], shows that *PRS*s with maximal period could be obtained if the polynomials, describing the connections in the feedback of the *RFSR*, should be primitive. Second, the words of the basic massive of a random block can be filled with the consecutive *n*-bits segments (i.e. *n*-tuples) of a gamma, generated by a *LFSR*, because it possesses the property P2, mentioned in the part 1.

The process of our multicore computer system (cluster) for modeling and design of *RFSR*s has passed over the following steps.

Step 1. Initially we were using a heterogeneous cluster, involving 15 conventional personal computers (*PC*s), based on the operation system (*OS*) Windows and the program media for automated engineer design Matlab. The performed experiments revealed the following weakness of this approach:

- the impossibility to assign different tasks for the different cores (*PC*s) of the cluster;
- after long work (15 -20 hours) the load of the RAM exceeded 1.5 GB;
- in Matlab exist hard restrictions over the maximal number of the monomials (about 50 monomials), which can contain an irreducible polynomial, describing the feedback of the *RFSR*s;
- the executing of long tasks (over 25 days) led to a difficult communications among the elements of the cluster.

All these disadvantages did not allow obtaining of irreducible polynomials with degree higher that 51. Due to this reason we modified the cluster.

Step 2. We replaced the OS Windows with the OS Linux. This measure allowed the *PC*s to work in a truly focused over the tasks manner without of supporting of any graphical and other services. Except this, the open source code of the OS Linux made possible to integrate compilators and other program products. As a result the term of the performance of the tasks reached 40 days with a near 100% load of the cores and without disturbances and malfunctions.

Step 3. We finished the improvements of our cluster by exploiting the library NTL [7] for testing the irreducibility of the generated polynomials. Except this, as a tool for adequate distribution and control over the computing processes the Open MPI (Open Source High Performance Computing) was applied.

After the performed improvements, our cluster demonstrates the following advantages:

1) Every core can work with a near 100 % load over its partial task.

2) The distribution of the load among the cores is very precise and is near optimal.

3) After very long tasks (about 40 days) the load of the memory for the every PC do not exceed 270 MB.

4) The exploitation of the OS Linux provides extra computational resources, because it is not necessary to support any graphical and antivirus services. This leads to a valuable enhancement of the computational effectiveness of the cluster as a whole.

The main result, obtained during the exploitation of our cluster, is the creating of massives of irreducible polynomials of degrees 257, 1024, 2500, 5000, 7500, 10001. The "weight" of the polynomials (i.e. the number of involved monomials) is greater than 11. As an illustration we provide a table of irreducible polynomials over *GF* (2) of degree 1024, which weight is 11.

Table I

Irreducible polynomials over GF (2) of degree 1024

	$x^{1024}+x^{51}+x^{8}+x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$
	$x^{1024}+x^{602}+x^{8} +x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$
	$x^{1024}+x^{513}+x^{9}+x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$
	$x^{1024}+x^{320}+x^{9}+x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$
	$x^{1024}+x^{950}+ x^{11}+$ $x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$
	$x^{1024}+x^{52}+ x^{10}+ x^{7}+x^{6}+x^{5}+x^{4}+x^{3}+x^{2}+x^{1}+x^{0}$

	$x^{1024}+x^{851}+ x^{12}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{816}+ x^{13}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{706}+ x^{10}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{945}+ x^{13}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{429}+ x^{12}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{839}+ x^{14}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{586}+ x^{12}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{625}+ x^{12}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{857}+ x^{15}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{547}+ x^{13}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{969}+ x^{15}+ x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{647}+ x^{12}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{1007}+ x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{606}+ x^{13}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{228}+ x^{14}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{702}+ x^{12}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{897}+ x^{16}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{707+} x^{12}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{545}+ x^{14}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{778}+ x^{13}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{330}+ x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{483}+ x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{788}+ x^{14}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$

	$x^{1024}+x^{390}+x^{16}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{610}+x^{16}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{724}+x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{823}+x^{21}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{847}+x^{21}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{929}+x^{21}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{188}+x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{887}+x^{21}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{57}+x^{17}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$
	$x^{1024}+x^{700}+x^{15}+x^7+x^6+x^5+x^4+x^3+x^2+x^1+x^0$

Besides, we have conducted a collateral study of the possibility of practical realization of the *RFSR*s by means of digital signal processors (*DSP*s). Namely we have used dsPIC 30F4011 in the MPLAB environment and C-30. The results show the stream ciphers based on *RFSR*s:

- satisfy truly the properties P1, P2 and P3, mentioned in part 1 of our paper;
- they could be implemented in the cost-effective *DSP*s and successfully exploited for protecting the information in a large range of communication and computer networks.

5. Conclusions

In the paper a multicore computer system for design of stream ciphers based on the so - named random feedback shift registers (*RFSR*s). The system is built in the Faculty of Technical sciences of the University of Shumen “Bishop Konstatin Preslavsky”, shumen, Bulgaria.

The results, obtained during the exploitation, show that:

1) The stream ciphers, based on *RFSR*s, possess a far higher crypto – resistance in a comparison with the classical stream ciphers.

2) These stream ciphers could be implemented in the cost-effective *DSP*s and successfully exploited for protecting the information in a large range of communication and computer networks.
The most important directions of our future work are:
1) The improvement of the software for selecting primitive polynomials among the found irreducible polynomials.
2) The extension of the involved cores by including in the system the computers of all laboratories of the Faculty of Technical sciences.
3) The enlarging the set of models of stream ciphers, based on *RFSR*s.

Acknowledgments

The authors wish to thank Prof. PhD Halit Hami Oz with Istanbul Aydin University for his support of our participation in IWW2010 and Dr. Victor Shoup with New York University, who has kindly provided us the opportunity to use his library NTL.

Finally, the authors appreciate the thoughtful critique by the referees.

REFERENCES

[1] A.Menezes, P.van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC press (1996).
[2] Bruce Schneier, Applied Cryptography 2nd edition: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc., (1996).
[3] Bedzhev B. Y., Tasheva Zh. N., Stoyanov B. P., N-adic Summation-Shrinking Generator. Basic properties and empirical evidences, Cryptology ePrint Archive, http://eprint.iacr.org/2005/068.pdf
[4] I.V. Chugunkov, M.A. Ivanov, Theory, the use and evaluation of the quality of random sequences generators, Kudriz – Obraz, Moscow, (2003) – in Russian.
[5] Zhukov I. Yu., Ivanov M. A., Osmolovskiy S. A., Principles of design of crypto – resistible generators of pseudo – random codes, Problems of the information security. Computer systems, No1, 2001, pp. 55-65 – in Russian
[6] Osmolovskiy S. A., Stochastic methods for data transferring, Radio and communication, Moscow, 1991 – in Russian
[7] /NTL: A Library for doing Number Theory/ http://www.shoup.net/