Parallel Generators of Pseudo-random Numbers with Control of Calculation Errors

S. Dichenko[1], **O. Finko**[2]

Kuban State University of Technology.
Moskowskaya St., 2, Krasnodar 350072, Russia
[1]*dichenko.sa@yandex.ru*, [2]*ofinko@yandex.ru*

Abstract: ***Algorithms of parallel realization of classical "congruent" methods of generation of pseudo-random numbers are discussed. Algebraic (equational) and matrix interpretations of ways of realization of algorithms are presented. In order to achieve high level of performance of modern and perspective technical tools of cryptographic protection of data, as well as to provide contemporary stochastic methods of modeling, it is necessary to realize existing "congruent" methods in parallel. In developing solutions for the problems of high level of complexity, such as cryptographic protection of data, ensuring high levels of security of functioning in real time is of out-most importance. In this article the means of ensuring of functional diagnosis of the tools for pseudo-random numbers generation by means of application of classical methods of superfluous arithmetic coding (module control) are briefly discussed.***

Keywords: Pseudo-random numbers, congruent method, cryptography, fault diagnosis and tolerance in cryptography.

1. Introduction

Pseudo-random number (PRN) generators have important applied meaning for the implementation of most cryptographic algorithms and key material generation systems [1] – [6]. Stricter requirements for the encryption speed and increasing amount of data which needs to be protected cause the necessity to construct parallel algorithms of PRN generation, what should be supported by providing the required level of reliability of their operation [7], [8].

The purpose of the article is constructing parallel algorithms of PRN generation with control of calculation errors based on the congruent method.

2. Varieties of the congruent method

Varieties of the congruent method, in particular, are expressed by formulas (1) – (4):

$$x_{n+1} = \left|bx_n\right|_m, \tag{1}$$

$$x_{n+1} = \left|bx_n + c\right|_m, \tag{2}$$

$$x_{n+1} = \left|b^d x_n + c\right|_m, \tag{3}$$

$$x_{n+1} = \left|b_1^{d_1} x_n + b_2^{d_2} x_{n-1} + c\right|_m, \tag{4}$$

where m – module, b – multiple, $0 \le b < m$; c – increment, $0 \le c < m$; x_n – the initial value, $0 \le x_n < m$; $\left|\bullet\right|_m$ – smallest non-negative residue of number $\bullet$ module m.

In accordance with the purpose, we pose a problem to construct parallel algorithms of PRN generation with control of calculation errors. Algorithms of PRN generation are shown as algebraic and matrix formulas.

3. Development of parallel algorithms

3.1 Algebraic method

Formula (1) can take the following series of algebraic transformations:

$$\begin{cases} x_{n+1} = \left|bx_n\right|_m, \\ x_{n+2} = \left|bx_{n+1}\right|_m = \left| b\left(\underbrace{bx_n}_{x_{n+1}}\right)\right|_m = \left|b^2 x_n\right|_m, \\ x_{n+3} = \left|bx_{n+2}\right|_m = \left| b\left(\underbrace{b^2 x_n}_{x_{n+2}}\right)\right|_m = \left|b^3 x_n\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|bx_{n+k-1}\right|_m = \left|b^k x_n\right|_m, \end{cases}$$

with $k = 1, 2, \ldots$ or we have:

$$\begin{cases} x_{n+1} = \left|\beta_1 x_n\right|_m, \\ x_{n+2} = \left|\beta_2 x_n\right|_m, \\ \ldots\ldots\ldots\ldots \\ x_{n+k} = \left|\beta_k x_n\right|_m, \end{cases} \tag{5}$$

where $\beta_i = \left|b^i\right|_m$, $i = 1, 2, \ldots, k$.

For (2) it is possible to get:

$$\begin{cases} x_{n+1} = \left|bx_n + c\right|_m, \\ x_{n+2} = \left|bx_{n+1} + c\right|_m = \left| b\left(\underbrace{bx_n + c}_{x_{n+1}}\right) + c\right|_m = \left| b^2 x_n + c(b+1)\right|_m, \\ x_{n+3} = \left|bx_{n+2} + c\right|_m = \left| b^3 x_n + c\left(b^2 + b + 1\right)\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|bx_{n+k-1} + c\right|_m = \left| b^k x_n + c\left(\sum_{j=1}^{k-1} b^j + 1\right)\right|_m, \end{cases}$$

with $k = 2, 3, \ldots$ or we have:

$$\begin{cases} x_{n+1} = \left|\beta_1 x_n + \gamma_1\right|_m, \\ x_{n+2} = \left|\beta_2 x_n + \gamma_2\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|\beta_k x_n + \gamma_k\right|_m, \end{cases} \tag{6}$$

where $\beta_i = \left|b^i\right|_m$, $\gamma_1 = |c|_m$, $\gamma_{i+1} = \left| c\left(\sum_{j=1}^{k-1}\beta_j + 1\right)\right|_m$;

For (3) it is possible to get:

$$\begin{cases} x_{n+1} = \left|b^d x_n + c\right|_m, \\ x_{n+2} = \left|b^d x_{n+1} + c\right|_m = \left| b^{2d} x_n + c\left(b^d + 1\right)\right|_m, \\ x_{n+3} = \left|b^d x_{n+2} + c\right|_m = \left| b^{3d} x_n + c\left(b^{2d} + b^d + 1\right)\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|b^d x_{n+k-1} + c\right|_m = \left| b^k x_n + c\left(\sum_{j=1}^{k-1} b^j + 1\right)\right|_m, \end{cases}$$

with $k = 2, 3, \ldots$ or we have:

$$\begin{cases} x_{n+1} = \left|\beta'_1 x_n + \gamma_1\right|_m, \\ x_{n+2} = \left|\beta'_2 x_n + \gamma_2\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|\beta'_k x_n + \gamma_k\right|_m, \end{cases} \quad (7)$$

where $\beta'_i = \left|b^{di}\right|_m$, $\gamma_1 = |c|_m$, $\gamma_{i+1} = \left| c\left(\sum_{j=1}^{k-1}\beta'_j + 1\right)\right|_m$;

$i = 1, 2, \ldots, k$.

For (4) it is possible to get:

$$\begin{cases} x_{n+1} = \left|\lambda_1 x_n + \beta_1 x_{n-1} + \gamma_1\right|_m, \\ x_{n+2} = \left|\lambda_2 x_n + \beta_2 x_{n-1} + \gamma_2\right|_m, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{n+k} = \left|\lambda_k x_n + \beta_k x_{n-1} + \gamma_k\right|_m, \end{cases} \quad (8)$$

where $\lambda_2 = \left|\lambda_1^2 + \beta_1\right|_m$; $\lambda_i = \left|\lambda_1\lambda_{i-1} + \beta_1\lambda_{i-2}\right|_m$,

when $i = 3, 4, \ldots$;

$\beta_i = \left|\beta_1\lambda_{i-1}\right|_m$, when $i = 2, 3, \ldots$;

$\gamma_1 = c$; $\gamma_i = \left| c\left(\gamma_{i-1} + \lambda_{i-1}\right)\right|_m$, when $i = 2, 3, \ldots$.

3.2 Examples

Let's have $x_0 = 431$, $b = 756$, $m = 2047$.

Find x_1, x_2, x_3 using (5):

$x_1 = \left|bx_0\right|_m = \left|756 \cdot 431\right|_{2047} = 363,$

$x_2 = \left|b^2 x_0\right|_m = \left|756^2 \cdot 431\right|_{2047} = 130,$

$x_3 = \left|b^3 x_0\right|_m = \left|756^3 \cdot 431\right|_{2047} = 24.$

Let'shave $x_0 = 412$, $b = 531$, $c = 711$, $m = 4093$.

Find x_1, x_2, x_3, x_4 using (6):

$x_1 = \left| bx_0 + c\right|_m = \left| 531 \cdot 412 + 711\right|_{4093} = 2554,$

$x_2 = \left| b^2 x_0 + c(\beta_1 + 1)\right|_m$

$= \left| 531^2 \cdot 412 + 711(531 + 1)\right|_{4093} = 2102,$

$x_3 = \left| b^3 x_0 + c\left(\sum_{j=1}^{2}\beta_j + 1\right)\right|_m$

$= \left| 531^3 \cdot 412 + 711\left(531^2 + 531 + 1\right)\right|_{4093} = 3577,$

$x_4 = \left| b^4 x_0 + c\left(\sum_{j=1}^{3}\beta_j + 1\right)\right|_m$

$= \left| 531^4 \cdot 412 + 711\left(531^3 + 531^2 + 531 + 1\right)\right|_{4093} = 946.$

3.3 Matrix method

Let's present the considered algorithms of PRN generation in matrix form.

For (1) it is possible to get:

$$\mathbf{X} = \left|\mathbf{B}\, x_n\right|_m = \left| \begin{pmatrix}\beta_1 \\ \beta_2 \\ \ldots \\ \beta_k\end{pmatrix} \cdot x_n \right|_m, \quad (9)$$

where $\beta_i = \left|b^i\right|_m$; $i = 1, 2, \ldots, k$.

For (2) it is possible to get:

$$\mathbf{X} = \left|\mathbf{B}\, x_n + \mathbf{G}\right|_m = \left| \begin{pmatrix}\beta_1 \\ \beta_2 \\ \ldots \\ \beta_k\end{pmatrix} \cdot x_n + \begin{pmatrix}\gamma_1 \\ \gamma_2 \\ \ldots \\ \gamma_k\end{pmatrix} \right|_m, \quad (10)$$

where

$\beta_i = \left|b^i\right|_m$, $\gamma_1 = |c|_m$, $\gamma_{i+1} = \left| c\left(\sum_{j=1}^{k-1}\beta_j + 1\right)\right|_m$; $i = 1, 2, \ldots, k$.

For (3) it is possible to get:

$$\mathbf{X} = \left|\mathbf{B}\, x_n + \mathbf{G}\right|_m = \left| \begin{pmatrix}\beta_1 \\ \beta_2 \\ \ldots \\ \beta_k\end{pmatrix} \cdot x_n + \begin{pmatrix}\gamma_1 \\ \gamma_2 \\ \ldots \\ \gamma_k\end{pmatrix} \right|_m,$$

where $\beta'_i = \left|b^{di}\right|_m$, $\gamma_1 = |c|_m$, $\gamma_{i+1} = \left| c\left(\sum_{j=1}^{k-1}\beta'_j + 1\right)\right|_m$;

$i = 1, 2, \ldots, k$.

For (4) it is possible to get:

$$\mathbf{X} = \left|\mathbf{A}\, x_n + \mathbf{B}\, x_{n-1} + \mathbf{G}\right|_m = \left| \begin{pmatrix}\lambda_1 \\ \lambda_2 \\ \ldots \\ \lambda_k\end{pmatrix} \cdot x_n + \begin{pmatrix}\beta_1 \\ \beta_2 \\ \ldots \\ \beta_k\end{pmatrix} \cdot x_{n-1} + \begin{pmatrix}\gamma_1 \\ \gamma_2 \\ \ldots \\ \gamma_k\end{pmatrix} \right|_m,$$

where $\lambda_2 = \left|\lambda_1^2 + \beta_1\right|_m$; $\lambda_i = \left|\lambda_1\lambda_{i-1} + \beta_1\lambda_{i-2}\right|_m$,

when $i = 3, 4, \ldots$;

$\beta_k = \left|\beta_1\lambda_{k-1}\right|_m$, when $k = 2,3, \ldots$;

$\gamma_1 = c$; $\gamma_k = \left| c\left(\gamma_{k-1} + \lambda_{k-1}\right)\right|_m$, when $k = 2, 3, \ldots$.

3.4 Examples

Let's have $x_n = 431$, $b = 756$, $m = 2047$.

Find x_1, x_2, x_3 using (9):

$$\mathbf{X} = \left|\mathbf{B}\, x_n\right|_m = \left| \begin{pmatrix} 756 \\ 756^2 \\ 756^3 \end{pmatrix} \cdot 431 \right|_{2047} = \begin{pmatrix} 363 \\ 130 \\ 24 \end{pmatrix}.$$

Let'shave $x_n = 412$, $b = 531$, $c = 711$, $m = 4093$.

Find x_1, x_2, x_3, x_4 using (10):

$$\mathbf{X} = \left|\mathbf{B}\, x_n + \mathbf{G}\right|_m$$

$$= \left| \begin{pmatrix} 531 \\ 531^2 \\ 531^3 \\ 531^4 \end{pmatrix} \cdot 412 + \begin{pmatrix} 711 \\ 711 \cdot (531+1) \\ 711 \cdot (531^2 + 531 + 1) \\ 711 \cdot (531^3 + 531^2 + 531 + 1) \end{pmatrix} \right|_m$$

$$= \begin{pmatrix} 2554 \\ 2102 \\ 3577 \\ 946 \end{pmatrix}.$$

4. Control of calculation errors

Let's use $M = m \cdot q$, as a composite module in calculating any of the obtained systems of expressions (5) – (8), where $q \le m$ module introduced for the implementation of control of calculation errors.

Then each of the expressions (5) – (8) can be rewritten as a system of expressions on module M, for example (7):

$$\begin{cases} x^*_{n+1} = \left|\beta^*_1 x^*_n + \gamma^*_1\right|_M, \\ x^*_{n+2} = \left|\beta^*_2 x^*_n + \gamma^*_2\right|_M, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ x^*_{n+k} = \left|\beta^*_k x^*_n + \gamma^*_k\right|_M, \end{cases} \qquad (11)$$

where $\beta^*_i = \left|b^{di}\right|_M$, $\gamma^*_1 = |c|_M$, $\gamma^*_{i+1} = \left| c\left(\sum_{j=1}^{k-1} \beta^*_j + 1\right)\right|_M$;

$i = 1, \ldots, k$.

Control digits are calculated by the system:

$$\begin{cases} r_{n+1} = \left|\beta^{**}_1 r_n + \gamma^{**}_1\right|_q, \\ r_{n+2} = \left|\beta^{**}_2 r_n + \gamma^{**}_2\right|_q, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ r_{n+k} = \left|\beta^{**}_k r_n + \gamma^{**}_k\right|_q, \end{cases} \qquad (12)$$

where $\beta^{**}_i = \left|b^{di}\right|_q$, $\gamma^{**}_1 = |c|_q$, $\gamma^{**}_{i+1} = \left| c\left(\sum_{j=1}^{k-1} \beta^{**}_j + 1\right)\right|_q$;

$i = 1, \ldots, k$.

Thus the result of (11), (12) is a redundant arithmetic code, represented by informational $\left[x^*_{n+1}, \ldots, x^*_{n+k}\right]$ and control vectors $\left[r_{n+1}, \ldots, r_{n+k}\right]$. If the minimal values of the arithmetic code distance $d_{\min} = q - 1$ control arithmetic expressions are as follows:

$$\begin{cases} \text{error is no, if } \left|x^*_{n+j}\right|_q - r_{n+j} = 0, \\ \text{error is, if } \left|x^*_{n+j}\right|_q - r_{n+j} \neq 0, \end{cases}$$

where $1 \le j \le k$.

The final result is:

$$x_{n+1} = \left|x^*_{n+1}\right|_m,\ x_{n+2} = \left|x^*_{n+2}\right|_m, \ldots,\ x_{n+k} = \left|x^*_{n+k}\right|_m.$$

5. Conclusion

Developed algorithms of accurate parallel PRN generation provide the required level of perspective highly productive cryptographic means of protection of data.

References

[1] A.V.Babash, G.P.Shankin, "Cryptology". SOLON-PRESS. 1997. (book style)

[2] Henk C.A. van Tilborg, "Fundamentals of Cryptology". KLUWER ACADEMIC PUBLISHERS.2000. (book style)

[3] B.Schneier, "Applied Cryptography". John Wiley & Sons, Inc. 1996. (book style)

[4] B.A.Forouzan, "Cryptography and Network Security". McGraw Hill. 2008. (book style)

[5] D.E.Knuth,"The Art of Computer Programming. Volume 2. Seminumerical Algorithms". Addison-Wesley. Redwood City. 1981. (book style)

[6] C.P.Schonorr, "On the Construction of Random Number Generators and Random Function Generators". EUROCRUPT. 1988. (journal style)

[7] "Cryptographic Hardware and Embedded Systems – CHES 2004". 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings

[8] "Fault Diagnosis and Tolerance in Cryptography". Third International Workshop, FDTC 2006, Yokohama, Japan, October 10, 2006, Proceedings.

Sergey Dichenko- Is a Graduate student. Research interests are development of parallel algorithms for generating pseudo-random numbers and binary sequences. Error control algorithms.

Oleg Finko - Professor, Doctor of Technical Sciences. Professor of Department of computer technologies and information security of the Kuban State University of Technology. Research interests -a residue number system, the use of error-correcting coding techniques in cryptography, multi-biometric encryption, digital signature algorithms improve, secure electronic document systems, Parallel computing logic by modular numerical polynomials.URL: *http://www.mathnet.ru/eng/person/40004*