Secure pseudo-random linear binary sequences generators based on arithmetic polynoms

Oleg Finko[a], Sergey Dichenko

[a] *Computer Systems and Information Security of KubSTU, Krasnodar, Russia*

Abstract

We present a new approach to constructing of pseudo-random binary sequences (PRS) generators for the purpose of cryptographic data protection, secured from the perpetrator's attacks, caused by generation of masses of hardware errors and faults. The new method is based on use of linear polynomial arithmetic for the realization of systems of boolean characteristic functions of PRS' generators. "Arithmetizatio" of systems of logic formulas has allowed to apply mathematical apparatus of residue systems for multisequencing of the process of PRS generation and organizing control of computing errors, caused by hardware faults. This has guaranteed high security of PRS generator's functioning and, consequently, security of tools for cryptographic data protection based on those PRSs.

1. Introduction

PRS' generators play an important role in building of communication with cryptographic data protection [1, 2]. From the list of known attacks on information security is important type of attacks, based on the generation of hardware errors functioning of the nodes forming the binary PRS [3]. To ensure the required level of interference and fault tolerance of digital devices developed many methods, the most common of which are backup methods and methods of error-correcting coding [4]. However, allocation methods do not provide the required levels of fault tolerance for restrictions on hardware costs, and methods of error-correcting coding is not adapted to the specifics of construction and operation means of data protection (MDP), in particular, the generators of the PRS.

2. Analysis of attacks based on hardware faults generation

Currently, the following types of attacks on sites of formation of binary PRS are considered (attack on) [5]:

Email addresses: ofinko@yandex.ru (Oleg Finko), dichenko.sa@yandex.ru (Sergey Dichenko)

- analysis of results of power consumption measurements;
- analysis of results of operations performance duration;
- analysis of accidental hardware faults;
- analysis of intentionally generated hardware faults, etc.

The last two types of faults are not investigated enough currently and thus are threatening to the information security of the functioning of modern and perspective MDP. The origin of those attacks lies in the use of thermal, high frequency, ionizing and other types of external influences onto MDP for the purpose of creation of masses of faults in hardware functioning by initialising of computing errors.

Hardware attacks can be divided into two classes:

1. **Direct hardware attacks.** The consequences of those attacks are failures of data protection tools. There is a method of analysis of the consequences of those failures. These types of attacks mean that in distortion in the certain places algorithm of transformation, which results in computing errors. Those errors can lead, for example, to repeated generation of the elements of PRS or in generation of faulty elements of PRS, which is unacceptable
2. **Attacks on post failure recovery means.** Some systems do not recovery means. If the system protection is destroyed, it is impossible to restore the operational mode. That is why such systems need to have means of protection against attacks of the malefactor and to support the possibility of updating the security system without stopping the programme running.

Attacks, based on errors generation by means of external influence are highly efficient for the majority of currently known and used algorithms of PRS generation. It is known that probability of error generation is proportional to the time corresponding registers has been affected by the radiation, if the registers are in favourable condition for error occurrence, and to the quantity of bits, in which the error occurrence is expected. The most widely used and proven means of creating PRS are algorithms and structures — Linear feedback shift register (LFSR) — of PRS generation, based on the use of feedback functions of logic [1, 2].

The structure of LFSR is determined by the forming polynomial:

$$D(\chi) = \chi^{\tau} + \chi^{t_l} + \ldots + \chi^{t_2} + \chi^{t_1} + 1,$$

where $\tau,\ t_i \in N$ and characteristic equation based on it:

$$\begin{aligned} x_{p+\tau} &= x_p \oplus x_{p+t_1} \oplus x_{p+t_2} \oplus \ldots \oplus x_{p+t_l} \\ &= c_0 x_p \oplus c_1 x_{p+1} \oplus \ldots \oplus c_{\tau-2} x_{p+\tau-2} \oplus c_{\tau-1} x_{p+\tau-1}, \end{aligned} \tag{1}$$

where $x_p,\ c_i \in \{0,\ 1\}$; $p \in N$; $i = 0,\ 1,\ \ldots,\ \tau - 1$; $c_{i \in \{0,\ t_1,\ t_2,\ \ldots,\ t_l\}} = 1$.

In linear algebra the next element of PRS $x_{p+\tau}$ is calculated as the following multiplication:

$$\left\| \begin{matrix} x_{p+1} \\ x_{p+2} \\ \cdots\cdots \\ x_{p+\tau-1} \\ x_{p+\tau} \end{matrix} \right\|^{\top} = \left\| \begin{matrix} 0 & 1 & \ldots\ldots & 0 & 0 \\ & & \cdots\cdots\cdots & & \\ 0 & 0 & \ldots\ldots & 1 & 0 \\ 0 & 0 & \ldots\ldots & 0 & 1 \\ c_0 & c_1 & \ldots\ldots & c_{\tau-2} & c_{\tau-1} \end{matrix} \right\|^{\top} \cdot \left\| \begin{matrix} x_p \\ x_{p+1} \\ \cdots\cdots \\ x_{p+\tau-2} \\ x_{p+\tau-1} \end{matrix} \right\|^{\top}.$$

When the described attack is performed the conditions arise for PRS modification or its repeated generation. The effect of repeated generation of a site of PRS is explained by means of Fig. 1 (the forming polynomial: $D(\chi) = \chi^4+\chi+1$; the characteristic equation: $x_{p+4} = x_{p+1} \oplus x_p$; the initial conditions: $x_p = 1$, $x_{p+1} = 0$, $x_{p+2} = 1$, $x_{p+3} = 0$).

Thus, those attacks, which are based on creating the conditions under which mass hardware errors occur, are threatening for MDP. One of the ways of solving this problem is development of methods for increasing the reliability of the functioning of sites of data protection tools, mostly subjected to attacks oft he described type, in particular the sites of forming of the encryption algorithm (cipher), based on PRS generation.

3. Analysis of methods for reliable binary PRS generation

Currently the required level of functional reliability of the sites of binary PRS generation is reached both by using of excessive devices (reservation) and timely excess by various repetitions of the calculations. In digital schemotechnics there are solutions known, based on use of methods of error-correction coding [4]. In order to use those methods for PRS generators it is necessary preliminary to solve the issue multisequencing the process of PRS calculations. The solution is based on the use of classic parallel algorithms of recursion [10].

For example, for the characteristic equation:

$$x_{p+\tau} = x_{p+t} \oplus x_p, \tag{2}$$

corresponding to treen $D(\chi) = \chi^\tau + \chi^t + 1$, it is possible to build a system of characteristic equations:

$$\begin{cases} x_{q,\ \tau-1} = x_{q-1,\ \tau-1} \oplus x_{q-1,\ \tau+t-1}, \\ x_{q,\ \tau-2} = x_{q-1,\ \tau-2} \oplus x_{q-1,\ \tau+t-2}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_{q,\ 1} = x_{q-1,\ 1} \oplus x_{q-1,\ t+1}, \\ x_{q,\ 0} = x_{q-1,\ 0} \oplus x_{q-1,\ t}. \end{cases}$$

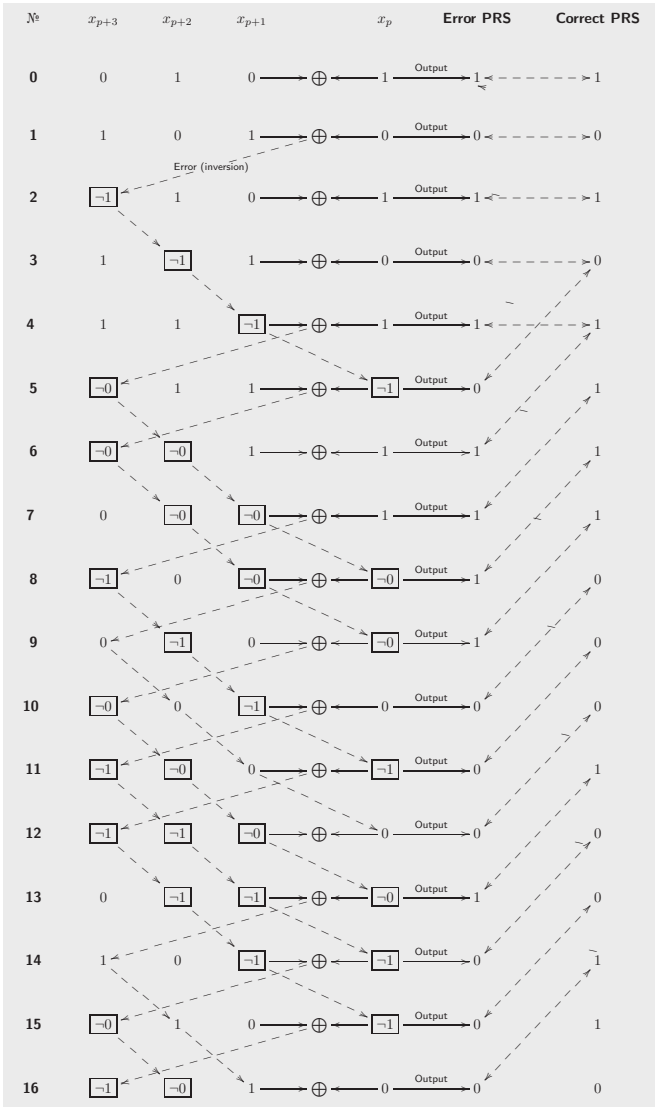


Figure 1: Example of operation of the LFSR when an error occurs ($\neg x$ — logical inversion x)

Similarly, for the general equation (1):

$$\begin{cases} x_{q,\ \tau-1} = c_0^{(\tau-1)} x_{q-1,\ 0} \oplus c_1^{(\tau-1)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(\tau-1)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(\tau-1)} x_{q-1,\ \tau-1}, \\ x_{q,\ \tau-2} = c_0^{(\tau-2)} x_{q-1,\ 0} \oplus c_1^{(\tau-2)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(\tau-2)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(\tau-2)} x_{q-1,\ \tau-1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{q,\ 1} = c_0^{(1)} x_{q-1,\ 0} \oplus c_1^{(1)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(1)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(1)} x_{q-1,\ \tau-1}, \\ x_{q,\ 0} = c_0^{(0)} x_{q-1,\ 0} \oplus c_1^{(0)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(0)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(0)} x_{q-1,\ \tau-1}, \end{cases} \tag{3}$$

where $c_i^{(j)} \in \{0,\ 1\}$ $(i,\ j = 0,\ 1,\ \ldots,\ \tau - 1)$. The principle of parallel lasing elements PRS based on (3) is illustrated by a graph (see Fig. 2)

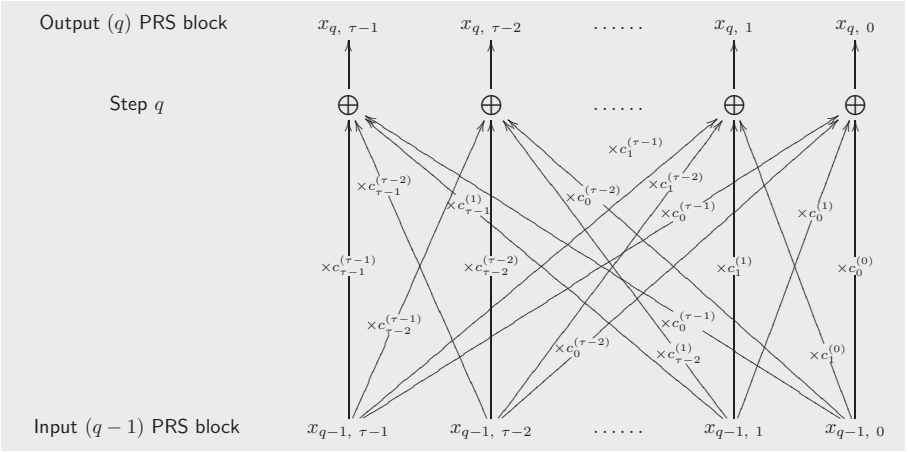


Figure 2: Graph generating elements parallel PRS based on (3)

System (3) forms an information matrix:

$$\mathbf{G_{Inf}} = \begin{Vmatrix} c_0^{(\tau-1)} & c_1^{(\tau-1)} & \ldots\ldots & c_{\tau-2}^{(\tau-1)} & c_{\tau-1}^{(\tau-1)} \\ c_0^{(\tau-2)} & c_1^{(\tau-2)} & \ldots\ldots & c_{\tau-2}^{(\tau-2)} & c_{\tau-1}^{(\tau-2)} \\ \ldots\ldots\ldots\ldots\ldots & & & & \\ c_0^{(1)} & c_1^{(1)} & \ldots\ldots & c_{\tau-2}^{(1)} & c_{\tau-1}^{(1)} \\ c_0^{(0)} & c_1^{(0)} & \ldots\ldots & c_{\tau-2}^{(0)} & c_{\tau-1}^{(0)} \end{Vmatrix}^{\top}.$$

Thus we obtain the q-th block of the PRS:

$$\mathbf{X}_q = \mathbf{G_{Inf}} \cdot \mathbf{X}_{q-1},$$

where

$$\mathbf{X}_q = \begin{bmatrix} x_{q,\ \tau-1} & x_{q,\ \tau-2} & \ldots & x_{q,\ 1} & x_{q,\ 0} \end{bmatrix}^{\top},$$
$$\mathbf{X}_{q-1} = \begin{bmatrix} x_{q-1,\ \tau-1} & x_{q-1,\ \tau-2} & \ldots & x_{q-1,\ 1} & x_{q-1,\ 0} \end{bmatrix}^{\top}.$$

To create the conditions for the application of separable linear redundant code will get form a matrix $\mathbf{G_{Gen}}$, consisting of the information and the check

matrix by adding (3) validation expressions:

$$\begin{cases} x_{q,\ \tau-1} = c_0^{(\tau-1)} x_{q-1,\ 0} \oplus c_1^{(\tau-1)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(\tau-1)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(\tau-1)} x_{q-1,\ \tau-1}, \\ x_{q,\ \tau-2} = c_0^{(\tau-2)} x_{q-1,\ 0} \oplus c_1^{(\tau-2)} x_{q-1,1} \oplus \ldots \oplus c_{\tau-2}^{(\tau-2)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(\tau-2)} x_{q-1,\ \tau-1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{q,\ 1} = c_0^{(1)} x_{q-1,\ 0} \oplus c_1^{(1)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(1)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(1)} x_{q-1,\ \tau-1}, \\ x_{q,\ 0} = c_0^{(0)} x_{q-1,\ 0} \oplus c_1^{(0)} x_{q-1,\ 1} \oplus \ldots \oplus c_{\tau-2}^{(0)} x_{q-1,\ \tau-2} \oplus c_{\tau-1}^{(0)} x_{q-1,\ \tau-1}, \\ x^*_{q,\ r-1} = a_0^{(r-1)} x_{q-1,\ 0} \oplus a_1^{(r-1)} x_{q-1,\ 1} \oplus \ldots \oplus a_{\tau-2}^{(r-1)} x_{q-1,\ \tau-2} \oplus a_{\tau-1}^{(r-1)} x_{q-1,\ r-1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x^*_{q,\ 0} = a_0^{(0)} x_{q-1,\ 0} \oplus a_1^{(0)} x_{q-1,\ 1} \oplus \ldots \oplus a_{\tau-2}^{(0)} x_{q-1,\ \tau-2} \oplus a_{\tau-1}^{(0)} x_{q-1,\ \tau-1}, \end{cases}$$

where r — the number of redundant symbols used linear code, $a_i^{(j)} \in \{0,\ 1\}$ $(i = 0,\ 1,\ \ldots,\ \tau-1;\ \ j = 0,\ \ldots,\ r-1)$.

A generator matrix takes the form:

$$\mathbf{G_{Gen}} = \left\| \begin{matrix} c_0^{(\tau-1)} & c_1^{(\tau-1)} & \ldots\ldots & c_{\tau-2}^{(\tau-1)} & c_{\tau-1}^{(\tau-1)} \\ c_0^{(\tau-2)} & c_1^{(\tau-2)} & \ldots\ldots & c_{\tau-2}^{(\tau-2)} & c_{\tau-1}^{(\tau-2)} \\ & \ldots\ldots\ldots\ldots\ldots & & & \\ c_0^{(1)} & c_1^{(1)} & \ldots\ldots & c_{\tau-2}^{(1)} & c_{\tau-1}^{(1)} \\ c_0^{(0)} & c_1^{(0)} & \ldots\ldots & c_{\tau-2}^{(0)} & c_{\tau-1}^{(0)} \\ a_0^{(r-1)} & a_1^{(r-1)} & \ldots\ldots & a_{\tau-2}^{(r-1)} & a_{\tau-1}^{(r-1)} \\ & \ldots\ldots\ldots\ldots\ldots & & & \\ a_0^{(0)} & a_1^{(0)} & \ldots\ldots & a_{\tau-2}^{(0)} & a_{\tau-1}^{(0)} \end{matrix} \right\|^{\top}.$$

Then the q-th block of the PRS with the control numbers (linear block code):

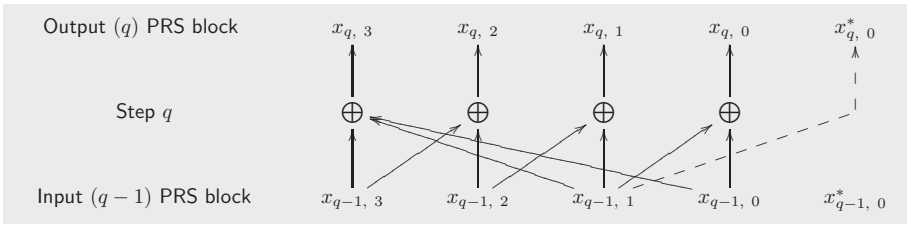
$$\mathbf{X}^*_q = \begin{bmatrix} x_{q,\ \tau-1} & x_{q,\ \tau-2} & \ldots & x_{q,\ 1} & x_{q,\ 0} & x^*_{q,\ r-1} & \ldots & x^*_{q,\ 0} \end{bmatrix}^{\top}$$

is calculated by:

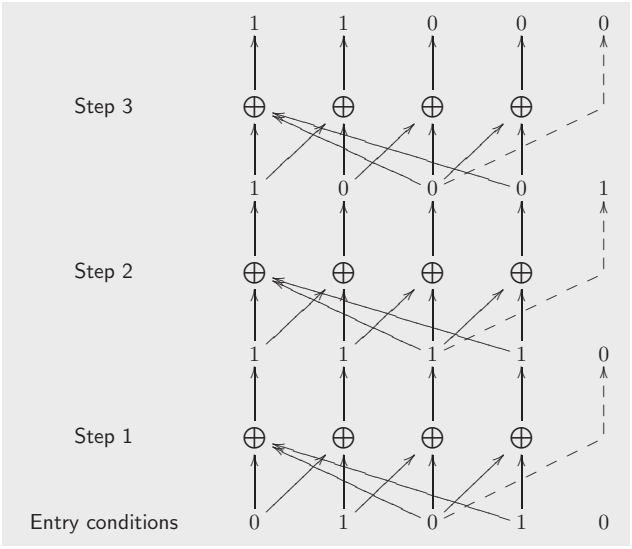
$$\mathbf{X}^*_q = \mathbf{G_{Gen}} \cdot \mathbf{X}_{q-1}.$$

Procedure error-correcting decoding is performed using the known rules [4]. The application of linear redundant codes and methods "hot" standby is not the only option for the implementation of functional diagnostics and fault tolerance of digital devices. Example graph parallel generation elements PRS error control computations is shown in Fig. 3.

Important advantages for these purposes have redundant arithmetic codes, in particular, so-called AN-codes and residue number systems (RNS) codes. The application of these codes to monitor logical data types and fault tolerance implementing devices became possible with the introduction of logical operations arithmetic expressions [11], in particular linear numerical polynomials (LNP) and modular forms [12].



a)



b)

Figure 3: a) Example graph parallel generation elements PRS (the characteristic equation: $x_{p+4} = x_{p+1} \oplus x_p$) error control computations (parity control); b) Numerical example

4. Error control operation of the PRS generators, based on "arithmetization" logical account

At the end of the last century there was formed a new direction parallel logic computation by the arithmetic (numeric) polynomials [11]. In particular received position "Modular arithmetic parallel logic computation" of the unification of the theoretical foundations of RNS [13, 14, 15] and theoretical foundations of parallel logic computation by the arithmetic of polynomials. The objective of the Association is to use advantages of RNS, i.e. parallelization arithmetic, error control calculations [16] in real time and ensure high availability of computing equipment, in the field of parallel logical account. In the following, these provisions were developed in various aspects, in particular, to-

wards the implementation of cryptographic functions [17, 18]. In particular, it was considered parallel generators PRS based, in General, nonlinear (canonical) arithmetic polynomials. Using use of LNP proposed by prof. V.D. Malyugin [11] for the construction of parallel generators PRS possible to reduce the maximum length of realizing polynomial to a value of $n+1$, where n — number of arguments of a Boolean function implemented [18]. In this paper, this method is used as the basis for the construction of safe (self-checking, fault-tolerant) generators on the basis of the excess bandwidth RNS.

It is known [19] that the q-th block of land PRS can be represented by a single LNP. The system of characteristic equations (3) must submit, as a system of Boolean functions, which in turn must be converted into a system:

$$\begin{cases} L_{\tau-1}(\mathbf{X}_{q-1}) = g_1^{(\tau-1)} x_{q-1,\ 0} + g_2^{(\tau-1)} x_{q-1,\ 1} + \ldots + g_\tau^{(\tau-1)} x_{q-1,\ \tau-1}, \\ L_{\tau-2}(\mathbf{X}_{q-1}) = g_1^{(\tau-2)} x_{q-1,\ 0} + g_2^{(\tau-2)} x_{q-1,\ 1} + \ldots + g_\tau^{(\tau-2)} x_{q-1,\ \tau-1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ L_0(\mathbf{X}_{q-1}) = g_1^{(0)} x_{q-1,\ 0} + g_2^{(0)} x_{q-1,\ 1} + \ldots + g_\tau^{(0)} x_{q-1,\ \tau-1}, \end{cases}$$

where $g_j^{(i)}$ (here and then) takes the value "0" or "1" depending on the entry in the i-th LNP $x_{q-1,\ j}$; $i,\ j = 0,\ 1,\ \ldots,\ \tau-1$.

The result of the calculation of i-LNP system appears to be a binary word of length $l_i = \lfloor \log(\sum_{j=\tau-1}^{0} g_j^{(i)}) \rfloor + 1$, where $\lfloor a \rfloor$ — the largest integer. Calculated total LNP:

$$\begin{aligned} L(\mathbf{X}_{q-1}) &= L_{\tau-1}(\mathbf{X}_{q-1}) + 2^{\gamma_1} L_{\tau-2}(\mathbf{X}_{q-1}) + \ldots + 2^{\gamma_{\tau-1}} L_0(\mathbf{X}_{q-1}) = \\ &= g_1^{(\tau-1)} x_{q-1,\ 0} + g_2^{(\tau-1)} x_{q-1,\ 1} + \ldots + g_\tau^{(\tau-1)} x_{q-1,\ \tau-1} + \\ &+ 2^{\gamma_1} (g_1^{(\tau-2)} x_{q-1,\ 0} + g_2^{(\tau-2)} x_{q-1,\ 1} + \ldots + g_\tau^{(\tau-2)} x_{q-1,\ \tau-1}) + \ldots \\ &\ldots + 2^{\gamma_{\tau-1}} (g_1^{(0)} x_{q-1,\ 0} + g_2^{(0)} x_{q-1,\ 1} + \ldots + g_\tau^{(0)} x_{q-1,\ \tau-1}) = \\ &= h_1 x_{q-1,\ 0} + h_2 x_{q-1,\ 1} + \ldots + h_\tau x_{q-1,\ \tau-1}, \end{aligned}$$

where $\gamma_k = \sum_{i=0}^{k-1} (l_i + 1)$, $k = 1,\ 2,\ \ldots,\ \tau-1$; $h_j \in Z$, or

$$L(\mathbf{X}_{q-1}) = \sum_{i=1}^{\tau} h_i x_{q-1,\ i-1}. \tag{4}$$

The final result is formed by implementing operator masking $\Xi^{\varphi}\{U\}$, which is used to determine the values of the φ-th Boolean function representation $U = (b_v \ldots b_\varphi \ldots b_2 b_1)_2$ (record $(\ldots)_2$ means representing a non-negative U in a binary number), that is $\Xi^{\varphi}\{U\} = b_\varphi$. Count calculation LNP (4) is represented in figure 3.

In RNS a nonnegative coefficient LNP (4) h_j is uniquely represented by a set of residues on the grounds RNS ($m_1,\ m_2,\ \ldots,\ m_n < m_{n+1} < \ldots < m_k$ —

pairwise simple):

$$h_j = (\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n,\ \alpha_{n+1},\ \ldots,\ \alpha_k)_{\mathrm{MA}}, \tag{5}$$

where $\alpha_t = |h_j|_{m_t}$; $t = 1,\ 2,\ \ldots,\ n,\ \ldots,\ k$; $|\,\centerdot\,|_m$ — the smallest non-negative deduction number $\centerdot$ on the modulo m. Operating range $M_n = m_1 m_2 \ldots m_n$ must meet $M_n > 2^s$, where $s = \sum\limits_{1 \le \varepsilon \le \tau} l_\varepsilon$ — the number of binary bits required to represent the result of a calculation LNP (4).

The remains $\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n$ are informational, and $\alpha_{n+1},\ \ldots,\ \alpha_k$ — are control. RNS in this case is called the extended and covers the complete set of states represented all k residues. This area is full range RNS $[0,\ M_k)$, where $M_k = m_1 m_2 \ldots m_n m_{n+1} \ldots m_k$, and consists of the operating range $[0,\ M_n)$, defined information bases RNS, and range identified redundant bases $[M_n,\ M_k)$, unacceptable region for the results of a calculation. This means that operations on numbers h_j are in the range $[0,\ M_k)$. Therefore, if the result of the operation RNS beyond M_n, it should output error calculation.

Consider RNS specified grounds $m_1,\ m_2,\ \ldots,\ m_n,\ m_{n+1}$. Each coefficient LNP h_j can be written as (5) and get redundant code RNS represented by the LNP system:

$$\begin{cases} U^{(1)} = L^{(1)}(\mathbf{X}_{q-1}) = \alpha_1^{(1)} x_{q-1,0} + \alpha_2^{(1)} x_{q-1,1} + \ldots + \alpha_\tau^{(1)} x_{q-1,\tau-1}, \\ U^{(2)} = L^{(2)}(\mathbf{X}_{q-1}) = \alpha_1^{(2)} x_{q-1,0} + \alpha_2^{(2)} x_{q-1,1} + \ldots + \alpha_\tau^{(2)} x_{q-1,\tau-1}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ U^{(n)} = L^{(n)}(\mathbf{X}_{q-1}) = \alpha_1^{(n)} x_{q-1,0} + \alpha_2^{(n)} x_{q-1,1} + \ldots + \alpha_\tau^{(n)} x_{q-1,\tau-1}, \\ U^{(n+1)} = L^{(n+1)}(\mathbf{X}_{q-1}) = \alpha_1^{(n+1)} x_{q-1,0} + \alpha_2^{(n+1)} x_{q-1,1} + \ldots \\ \ldots + \alpha_\tau^{(n+1)} x_{q-1,\tau-1}. \end{cases} \tag{6}$$

Substituting in (6) values of RNS residue on the appropriate grounds for each coefficient (4) and the values of the variables $x_{q-1,\,0},\ \ldots,\ x_{q-1,\,\tau-1}$, get the values of LNP system (6), where $U^{(1)},\ U^{(2)},\ \ldots,\ U^{(n)},\ U^{(n+1)}$ — nonnegative integer. In accordance with the Chinese remainder theorem solve the system of equations:

$$\begin{cases} U^* = |U^{(1)}|_{m_1}, \\ U^* = |U^{(2)}|_{m_2}, \\ \ldots\ldots\ldots\ldots\ldots \\ U^* = |U^{(n)}|_{m_n}, \\ U^* = |U^{(n+1)}|_{m_{n+1}}. \end{cases} \tag{7}$$

Since $m_1,\ m_2,\ \ldots,\ m_n,\ m_{n+1}$ are pairwise prime, then the only solution of (7) gives the expression:

$$U^* = \left| \sum_{s=1}^{n+1} M_{s,\ n+1} \mu_{s,\ n+1} U^{(s)} \right|_{M_{n+1}}, \tag{8}$$

where $M_{s,\ n+1} = \dfrac{M_{n+1}}{m_s}$, $\mu_{s,\ n+1} = |M^{-1}_{s,\ n+1}|_{m_s}$, $M_{n+1} = \prod\limits_{s=1}^{n+1} m_s$.

Graph parallel generation PRS based on (8) is shown in Fig. 4. The occurrence of the result of the calculation (8) in the range (control expression):

$$0 \le U^* < M_n,$$

means the absence of detectable errors of calculations.

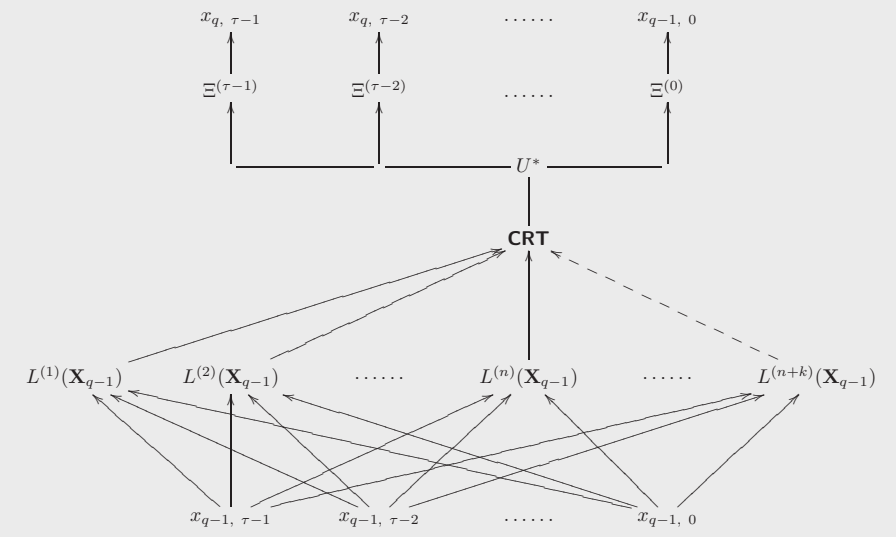


Figure 4: Graph of parallel generation PRS based on the Chinese remainder theorem (CRT)

5. Reconfiguration of equipment

Restore reliable operation of the generator of the PRS in the case of long-term failure is possible by correcting an error or reconfiguration of equipment generator (active redundancy). The first option is unacceptable because it does not guarantee no penetration of undetectable errors in the result of the encryption. By methods of modular redundant coding is made possible to apply a variant of the reconfiguration of the equipment by excluding from the operation of the failed equipment.

After localization of the faulty equipment — for example — a single channel operation RNS, the reconfiguration operation is performed by the calculation

U^* from the system:

$$\begin{cases} U^* = |U^{(1)}|_{m_1}, \\ \ldots\ldots\ldots\ldots\ldots \\ U^* = |U^{(n)}|_{m_n}, \\ U^* = |U^{(n+1)}|_{m_{n+1}}, \\ U^* = |U^{(n+2)}|_{m_{n+2}} \end{cases}$$

on the "right" reasons of residue number systems:

$$U^* = |\widetilde{U}^{(1)} B_{1,\ j} + \widetilde{U}^{(2)} B_{2,\ j} + \ldots + \widetilde{U}^{(n+2)} B_{n+2,\ j}|_{M_j},$$

where $\widetilde{U}^{(i)}$ — wrong balance; $B_{i,\ j}$ — orthogonal bases; $i,\ j = 1,\ 2, \ldots,\ n+2$; $i \neq j$; $B_{i,\ j} = \dfrac{M_j \mu_{i,\ j}}{m_i}$; $M_j = \dfrac{M_{n+2}}{m_j}$; $\mu_{i,\ j}$ is calculated from the comparison: $\dfrac{M_j \mu_{i,\ j}}{m_i} \equiv 1 \pmod{m_i}$. Compiled table 1 contains the values of the orthogonal bases and modules of the system for the occurrence of a single error for each base RNS.

Table 1: Calculation table orthogonally bases and modules RNS

j	$B_{1,\ j}$	$B_{2,\ j}$	$\cdots$	$B_{n+2,\ j}$	M_j
1	0	$\dfrac{M_1 \mu_{2,\ 1}}{m_2}$	$\cdots$	$\dfrac{M_1 \mu_{n+2,\ 1}}{m_{n+2}}$	$m_2 m_3 \ldots m_{n+2}$
2	$\dfrac{M_2 \mu_{1,\ 2}}{m_1}$	0	$\cdots$	$\dfrac{M_2 \mu_{n+2,\ 2}}{m_{n+2}}$	$m_1 m_3 \ldots m_{n+2}$
$\ldots\ldots$	$\ldots\ldots\ldots$	$\ldots\ldots\ldots$	$\cdots$	$\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots$
$n+2$	$\dfrac{M_{n+2} \mu_{1,\ n+2}}{m_1}$	$\dfrac{M_{n+2} \mu_{2,\ n+2}}{m_2}$	$\cdots$	0	$m_1 m_2 \ldots m_{n+1}$

6. Conclusion

It is known that the use of RNS already with two redundant bases allows us to provide a level of fault tolerance modular transmitter exceeds the tolerance provided by the method of rorovana equipment. This redundant hardware costs are reduced from 200% (triple) up to 30-40% (when using RNS) [20]. At the same time it should be noted that the amount of hardware, PRS generator operating in accordance obtained by the method, may exceed the hardware failover LFSR, built in accordance with traditional solutions. So you should made a fundamentally new level of functional flexibility of the designed generator PRS is able to implement and many other cryptographic functions, time-varying, without rebuilding the structure. This allows for the implementation of the device not only programmable logic integrated circuit, but also high-tech large

custom integrated circuits, in particular used for the implementation of number theoretic transformations in the field of digital signal processing.

The implementation of the PRS generators using LNP and redundant RNS allows to obtain a new class of solutions aimed at the safe implementation of the logical cryptographic functions, in particular parallel generators PRS. This is provided as a functional control equipment (in real time), and its fault tolerance through reconfiguration of the structure of the evaluator in the process of its degradation. Classic LFSR considered in the present work, is the basis and more complex, for example, combining generators PRS. Use for the implementation of the PRS generator modular arithmetic provides the possibility of applying the proposed solutions in the hybrid cryptosystems (including asymmetric) [18]. When this arithmetic calculator that supports the implementation of asymmetric cryptographic algorithms may be used to implement systems of Boolean functions (elements PRS).

References

[1] B. Schneier. Applied Cryptography. John Wiley & Sons, Inc. 1996.

[2] B.A. Forouzan. Cryptography and Network Security. McGraw Hill. 2008.

[3] B. Yang, K. Wu, R. Karri. Scan Based Side Channel Attack on Data Encryption Standard. Report 2004/324, pages 114–116, 2004.

[4] J.A. Hetagurov, Y.P. Prudnaya. Improving the reliability of digital devices redundant coding methods. Energiya, Moscow, 1974.

[5] J. Kelsey. Protocol Interactions and the Chosen Protocol Attack. Security Protocols, 5th Int'l Workshop, pages 91–104, New York, 1996. Springer-Verlag.

[6] C. Canovas, J. Clediere. What do DES S-boxes Say in Differential Side Channel Attacks? Report 2005/311, pages 191–200, 2005.

[7] V. Carlier, H. Chabanne, E. Dottax. Electromagnetic Side Channels of an FPGA Implementation of AES. Report 2004/145, pages 111–124, 2004.

[8] D. Page. Partitioned Cache Architecture as a Side-Channel Defence Mechanism. Report 2005/280, pages 213–225, 2005.

[9] P. Gutmann. Software Generation of Random Numbers for Cryptographic Purposes. Usenic Security Symp., Usenix Assoc., Berkeley, pages 243–257, Calif, 1998.

[10] J.M. Ortega. Introduction to Parallel & Vector Solution of Linear Systems. Plenum Press New York, NY, 1988.

[11] V.P. Shmerko. Malyugin's Theorems: A New Concept in Logical Control, VLSI Design, and Data Structures for New Technologies. Automation and Remote Control, volume 65, issue 6, pages 893–912, june 2004.

[12] O.A. Finko. Large Systems of Boolean Functions: Realization by Modular Arithmetic Methods. Automation and Remote Control, volume 65, issue 6, pages 871–892, june 2004.

[13] H.L. Garner. Number systems and arithmetic. Advances in Computers, volume 6, pages 131–194, 1965.

[14] M.A. Soderstrand, W.K. Jenkins, G.A. Jullien and F.J. Tailor. Residue Number System Arithmetic: Modern Application in Digital Signal Processing, NY, IEEE Press, 1986.

[15] A. Omondi, B. Premkumar. Residue Number System: Theory and Implementation. Imperial Collegt Press, London, 2007.

[16] W.K. Jenkins. The design of error checkers for self-checking residue number arithmetic, volume 4, pages 388-396, IEEE Trans. on Computers, 1983.

[17] O.A. Finko, A.K. Vishnevsky. Parallel realization of systems of substitutions by numerical polynoms. Papers of the Fifth International Conference "Parallel Computing and Control Problems", pages 935–943, Moscow, 2010.

[18] O.A. Finko, A.K. Vishnevsky. Standard function hybrid cryptosystem arithmetic and logical multinomial realization. Theory and Techniques of radio, pages 32–38, Voronezh, 2011.

[19] O.A. Finko, S.A. Dichenko, N.I. Eliseev. Error Function generator binary PRS control implemented on arithmetic polynomials. St. Petersburg state polytechnical university journal "Computer Science. Telecommunications and Control Systems", 4(176), pages 142–149, St. Petersburg, 2013.

[20] V.A. Krasnobaev. Reliable model in the computer residue number system. Electronic modeling, volume 7, number 4, pages 44–46, 1985.