Generalized Galois and Fibonacci Matrices in Cryptographic Applications

ANATOLY BELETSKY
Department of Electronics
National Aviation University,
Kyiv-03058, av. Kosmonavt Komarov, 1,
UKRAINE

Abstract: The terms of the Galois matrices $\boldsymbol{G}$, as well as those bijectively associated with them the Fibonacci matrices $\boldsymbol{F}$ connect by the operator of the right-hand transposition (that is, transposition to the auxiliary diagonal), are borrowed from the theory of cryptography, in which generators of pseudorandom number (PRN) widely use according to Galois and Fibonacci schemes (in configuration). A distinctive feature of both the $\boldsymbol{G}$ and $\boldsymbol{F}$ matrices is that the identical binary sequences can programmatically calculate the sequences formed by the PRN generators. The latter's constructions are based on linear feedback shift registers, implemented by software or hardware methods in Galois and Fibonacci architecture. The proposed generalized Galois matrices, discussed in the Chapter, significantly expand the variety of PRN generators. That is achieved both by increasing the number of generating elements θ (in the classical version used a single element $\theta = 10$) and since generalized generators can construct not only using PRN but also polynomials, not necessarily (as in classical generators), which are primitive. The listed features of generalized Galois matrices provide PRN generators with significantly higher cryptographic security than generators based on conventional matrices.

Key-Words: generators of pseudorandom numbers, linear feedback shift registers, Galois and Fibonacci matrices, Berlekamp–Massey algorithm.

Received: April 15, 2021. Revised: January 5, 2022. Accepted: January 25, 2022. Published: February 7, 2022.

1 Introduction

In the theory and practice of cryptographic information protection, one of the most critical problems is constructing generators of pseudorandom numbers (PRN) of the maximum length (period) with good statistical properties**.** There are two main types of PRN generators built using hardware or software. The first class of generators is made based on linear feedback shift registers (LFSR) in Galois or Fibonacci configurations (according to schemes) [1, 2]. The structural-logic diagrams of classical LFSR generator's PRN are uniquely defined by their generating primitive polynomials (PrP), through which the single-loop feedbacks in the shift registers are established [3, 4]. The software-implemented PRN generators, which make up the second class of generators, can also be built based on LFSR.

This Chapter focuses on constructing generalized matrix PRN generators in Galois and Fibonacci configurations [5, 6]. The terms of the Galois matrix $\boldsymbol{G}$ and those objectively associated with them by the operator of the right-hand transposition (i.e., transposition to the auxiliary diagonal [7]) of the Fibonacci matrix $\boldsymbol{F}$ borrowed from cryptography theory. The Galois and Fibonacci matrices will be called PRN generators.

In addition to the named base (initial) matrices $\boldsymbol{G}$ and $\boldsymbol{F}$, the so-called conjugate matrices $\boldsymbol{G}^*$ and $\boldsymbol{F}^*$ introduce in the Chapter, which forms by the classical (left-sided) transposition to the main diagonal of the corresponding initial matrices. For simplicity, the set of matrices $\{\boldsymbol{Q}\} = \{\boldsymbol{G}, \boldsymbol{F}, \boldsymbol{G}^*, \boldsymbol{F}^*\}$, which does not lead to ambiguity, will be called "Galois matrices". All Galois matrices can obtain by linear transformations of the left-sided and right-sided transposition of the Frobenius [8] standard form:

$$\boldsymbol{\Phi}_n = \begin{bmatrix} 0 & 0 & \dots & 0 & -c_0 \\ 1 & 0 & \dots & 0 & -c_1 \\ 0 & 1 & \dots & 0 & -c_2 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -c_{n-1} \end{bmatrix}, \qquad (1)$$

called in linear algebra the accompanying matrix of the unitary polynomial

$$\varphi_n(x) = x^n + c_{n-1}x^{n-1} + \ldots + c_k x^k \ldots + c_1 x^1 + c_0,$$
$$c_k \in GF(p).$$

The possibilities of using Frobenius matrices (1) for constructing a PRN generator based on the following properties $\boldsymbol{\Phi}_n$. First, if as a polynomial $\varphi_n(x)$ we choose a unitary irreducible polynomial f_n, represented by its vector form (by the set of polynomial coefficients)

$$\varphi_n(x) \Rightarrow f_n = 1\alpha_{n-1}\alpha_{n-2}\ldots\alpha_k\ldots\alpha_1\alpha_0,$$
$$\alpha_k = (-c_k) \bmod p,$$

then the matrix $\boldsymbol{\Phi}_n$ goes into the Fibonacci matrix

$$\boldsymbol{F}_n = \begin{bmatrix} 0 & 0 & \ldots & 0 & \alpha_0 \\ 1 & 0 & \ldots & 0 & \alpha_1 \\ 0 & 1 & \ldots & 0 & \alpha_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & \alpha_{n-1} \end{bmatrix}. \qquad (2)$$

And secondly, matrix (2) generates a linear recurrent $m-$sequence $\alpha_0, \alpha_1, \ldots \alpha_k, \ldots$ by transforming

$$(\alpha_k \alpha_{k+1} \ldots \alpha_{k+(n-1)}) \overset{p}{\otimes} \boldsymbol{F}_n = \\ = \alpha_{k+1}\alpha_{k+2}\ldots\alpha_{k+(n-1)}\alpha_{k+n}, \qquad (3)$$

for all $k \geq 0$.

Let's pay attention to that recursion feature (3). All high elements $\alpha_{k+1}\alpha_{k+2}\ldots\alpha_{k+(n-1)}$ of the output vector $\boldsymbol{V}_{\text{out}}$ are contained in the set of components of the input vector $\boldsymbol{V}_{\text{in}} = \alpha_k \alpha_{k+1} \ldots \alpha_{k+(n-1)}$. The only unknown part α_{k+n} of the vector $\boldsymbol{V}_{\text{out}}$ determined, according to relations (2), (3), by the scalar product of vectors $\boldsymbol{V}_{\text{in}}$ and $\boldsymbol{A} = \alpha_0 \alpha_1 \ldots \alpha_k \ldots \alpha_{n-2}\alpha_{n-1}$, i.e.

$$\alpha_{k+n} = (\alpha_k \alpha_0 + \alpha_{k+1}\alpha_1 + \ldots \\ \ldots + \alpha_{k+n-1}\alpha_{n-1}) \bmod p. \qquad (4)$$

Calculating a sequence of vectors $\boldsymbol{V}_{\text{out}}$ will be formative to illustrate the fourth-order Fibonacci matrix $\boldsymbol{F}_4$ generated by the binary PrP $f_4 = 10011$

$$\boldsymbol{F}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad (5)$$

generated by the binary PrP $f_4 = 10011$. As the initialization vector, let us designate it as $\overline{\boldsymbol{V}}_n$, on the left side of the expression (3), you can choose any non-zero binary vector of the fourth-order. Let this be the vector $\overline{\boldsymbol{V}}_4 = 1011$. The calculation results by formulas (3)-(5) of the recursive sequence are summarized in Table 1.

Table 1. The sequence of the state of the Fibonacci PRN generators generated PrP $f_4 = 10011$

Step (k)	The elements of $\boldsymbol{V}_{\text{out}}$				Step (k)	The elements of $\boldsymbol{V}_{\text{out}}$			
	0	1	2	3		0	1	2	3
0	1	1	0	1	8	1	0	0	0
1	1	0	1	0	9	0	0	0	1
2	0	1	0	1	10	0	0	1	0
3	1	0	1	1	11	0	1	0	0
4	0	1	1	1	12	1	0	0	1
5	1	1	1	1	13	0	0	1	1
6	1	1	1	0	14	0	1	1	0
7	1	1	0	0	15	1	1	0	1

The shading in Table 1 highlights the vector that coincides with the initialization vector. The number of non-repeating non-zero vectors generated by the Fibonacci generator turned out to be 15, as it should be for the selected parameters of the generation.

The vast majority of the generators of PRN are based on LFSR [9]. The main requirement for LFSR generators in cryptographic applications is to generate a sequence of cipher bits of maximum length (period). Its knowns that LFSR is a maximum period shift register if the corresponding feedback polynomial is primitive.

Along with LFSR generators, can use PRN generators based on shift registers with generalized feedback. Such generators include the Mersen vortex [10], which contains a modified Frobenius

matrix. Unfortunately, matrix PRN generators are not yet widely used in cryptography. Classic generators (both matrix and LFSR) do not provide the required level of cryptographic security. The noted disadvantage is that the output serial bits of the generator by the Berlecamp-Massey algorithm can be uniquely defined primitive polynomial, which generates the matrix of the PRN generator. And as a consequence, the generator turns out to be cracked [11].

The main task of this study is to develop PRN generators of the maximum period based on generalized Galois matrices in the general case over fields of arbitrary characteristics p, free from the Berlecamp-Massey attack.

2 Galois Classical Hardware and Matrix Generators PRN

Definition 1. To subset of classical PRN generators of the maximum period will include generators built based on linear shift registers covered by single-loop feedback, which is exclusively a function of a primitive polynomial that plays the role of a generator polynomial.

Usually, D-triggers are used as LFSR bits. An example of a fourth-order Galois generator, feedbacks in which fourth-degree PrP $f_4 = 10011$ formed, is shown in Fig. 1. Using Fig. 1, let us develop a mnemonic rule for constructing classic LFSR generators in the Galois configuration. For this purpose, we will supplement the drawing with dotted strokes, placing them on those parts of the circuit in which there are no XOR operators.

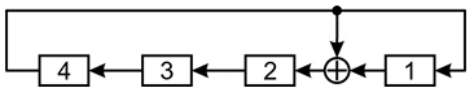

Fig. 1. Block diagram of the PRN generator in the Galois configuration generated by PrP $f_4 = 10011$

Then we put numbers 1 above the solid vertical lines (feedback lines) and numbers 0 above the dashed lines. Finally, we come to Fig. 2.

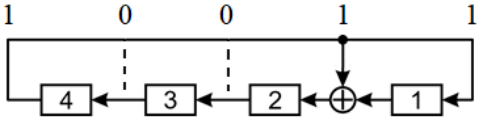

Fig. 2. To build a block diagram fourth-order Galois generator

As follows from Fig. 2, the ones of the primitive polynomial in vector form predetermine the position of the vertical lines in a single-loop feedback circuit in the classical LFSR Galois PRN generator.

Will illustrate the technology of applying formulated rules for drawing up a block diagram of the PRN generator of the maximum period in the Galois configuration will be illustrated by constructing a generator circuit generated by a PrP of the eighth-degree $f_8 = 101100101$. The solution to this problem involves the implementation of two stages of synthesis.

Stage 1. Form an eight-bit ring shift register (Fig. 3), in the nodes of the feedback line of which we equidistantly arrange the coefficients of the selected primitive polynomial

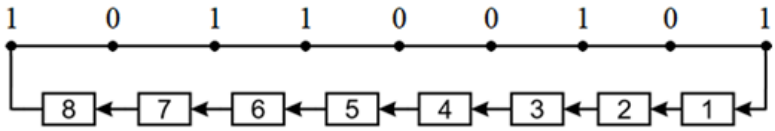

Fig. 3. To the construction of an eight-bit Galois generator circuit

Stage 2. Connecting, as shown in Fig. 4, the internal nodes of the feedback line, above which there are coefficients 1, with the XOR operator, we complete the construction of the classical LFSR Galois generator.

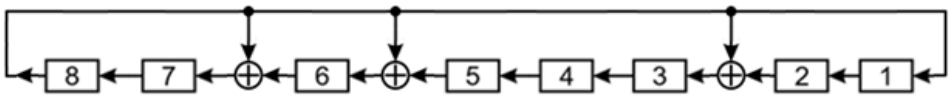

Fig 4. Block diagram of the Galois generator, generated PrP $f_8 = 101100101$

Similarly, by steps 1 and 2, it is possible to construct the block diagram of the classical LFSR generators in the Galois configuration for an arbitrary degree of the primitive polynomial that forms a feedback loop in the generator register. Classical Fibonacci LFSR generators PRN created from Galois generators by rotating the feedback loop relative to the vertical and horizontal axes. At the same time, the numbering of the shift register cells remains unchanged (Fig. 5).

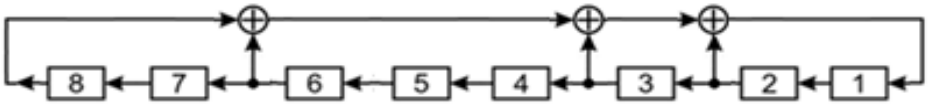

Fig 5. Block diagram of the Fibonacci generator, generated PrP $f_8 = 101100101$

Each LFSR generator (Galois or Fibonacci) corresponds to uniquely associated with the matrices, which we will denote by $\boldsymbol{G}$ and $\boldsymbol{F}$, respectively. A distinctive feature of the Galois and Fibonacci matrices is that it is possible to generate binary series similar to the m – sequences formed by the classical LFSR generators on their basis.

Let be $S(k)$ — the state vector of the n – discharge PRN generator in the Galois configuration after the k sync pulse (at the k step of the register shift), the calculation scheme of which is represented by the matrix expression

$$S(k+1) = S(k) \cdot \boldsymbol{G}_f^{(n)}, \; k = 0, 1, \ldots,$$
$$S(0) = \underbrace{00\ldots01}_{n \text{ bit}}. \qquad (6)$$

Our task is to calculate the Galois matrix for a given PrP $f = 1\alpha_{n-1}\alpha_{n-2}\ldots\alpha_k\ldots\alpha_1 1$, $\alpha_k \in GF(2) = \{0, 1\}$, with the help of which relation (6) forms the same sequence of pseudorandom numbers as the LFSR generator. Let us try first to deal with this problem for small orders of matrices. Then, let us turn to the analysis of the state of the triggers of the PRN generator (Fig. 6), previously shown in Fig. 1.

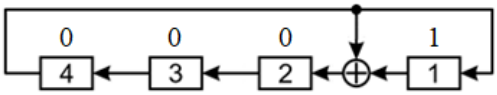

Fig. 6. Initial state illustration Generator PRN, according to Galois scheme

The numbers placed above the generator bits characterize the logic level of the signal at the output of the corresponding cell (trigger) of the register. As shown in Fig. 7, using sync pulses, 1 from the least significant bit of the register moved to its most essential bits.

From Fig.7, it follows that after the third synchrotact, logical ones arrive at the inputs of both the first and second flip-flops and, therefore, at the fourth step of the PRN generation (Fig. 8) appear at the outputs of these triggers.

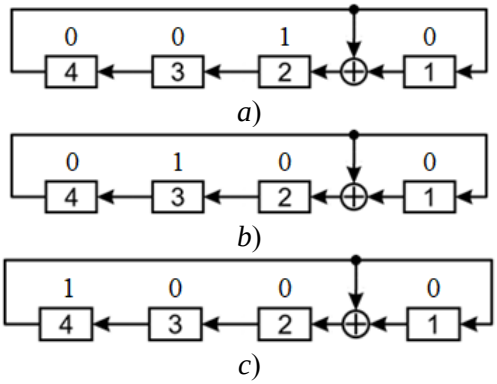

Fig. 7. PRN generator states after: *a*) - the first, *b*) - the second, *c*) - the third synchro tact

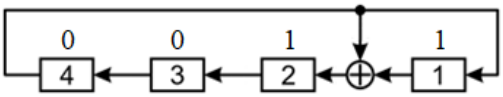

Fig. 8. State of the PRN generator after the fourth synchro title

Let us compose a matrix $\boldsymbol{G}_{13}^{(4)}$ from a set of state vectors $S(k)$, into which the Galois generator passes after the first four synchronizations, placing the vectors in the matrix, starting from its bottom row $i = 1$.

$$\boldsymbol{G}_{13}^{(4)} = \begin{array}{c} \uparrow i \\ \\ \\ \\ \leftarrow j \end{array} \begin{array}{c} \\ \\ \\ \\ 4\;3\;2\;1 \end{array} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{array}{c} \\ 4 \\ 3 \\ 2 \\ 1 \end{array}, \qquad (7)$$

Note that index 13 in the notation of the matrix $\boldsymbol{G}_f^{(n)}$ in (7) is the hexadecimal notation of PrP $f_4 = 10011$. We will continue to use the same form of presentation of the numerical values of the degree of polynomials in the future.

Thus, firstly, the matrix (7), when substituted into relation (6), forms a sequence of four-bit codes (Table 2), which include the multiplicative group of the field generated by PrP $f_4 = 10011$.

Table 2. The sequence of the state of the PRN generator from PrP $f_4 = 10011$

Step	LRS discharges				Step	LRS discharges			
(k)	4	3	2	1	(k)	4	3	2	1
0	0	0	0	1	8	0	1	0	1
1	0	0	1	0	9	1	0	1	0
2	0	1	0	0	10	0	1	1	1
3	1	0	0	0	11	1	1	1	0
4	0	0	1	1	12	1	1	1	1
5	0	1	1	0	13	1	1	0	1
6	1	1	0	0	14	1	0	0	1
7	1	0	1	1	15	0	0	0	1
6	1	1	0	0	14	1	0	0	1
7	1	0	1	1	15	0	0	0	1
6	1	1	0	0	14	1	0	0	1
7	1	0	1	1	15	0	0	0	1

And secondly, the top row of the matrix (7) is nothing but the fourth degree PrP $f_4 = 10011$, in which the leading unit remove, and the left element of the truncated polynomial is the coefficient α_{n-1}.

Based on the analysis of the matrix $\boldsymbol{G}_{13}^{(4)}$ in (7), we arrive at the following construction rule (synthesis algorithm) of the classical Galois matrix (CGM) $\boldsymbol{G}_f^{(n)}$ of the order n generated by a primitive polynomial f_n of degree n.

Algorithm for the synthesis of CGM: let f_n – a primitive binary polynomial of degree n and $\theta = 10$ – the minimal primitive element of the field $GF(2^n)$, generated by the polynomial f_n. Place θ in the lower right corner of the generated Galois matrix $\boldsymbol{G}_f^{(n)}$. All other digits of the bottom line $\boldsymbol{G}_f^{(n)}$, located to the element's left θ, are filled with zeros. Suppose the stage of formation of the next row its senior 1 goes beyond the left boundary of the matrix. In that case, the polynomial in this row reduces to the remainder modulo f_n. Thus, the row returns to the matrix, and the formation process of $\boldsymbol{G}_f^{(n)}$ continues further.

The right-hand side of the matrix (2) can represent in a more compact form [8]:

$$\boldsymbol{G}_f^{(n)} = \begin{pmatrix} \blacktriangleleft & f \\ \boldsymbol{E} & \boldsymbol{0} \end{pmatrix}, \qquad (8)$$

where $\boldsymbol{E}$ is the identity matrix of the $(n-1)-$order, the $\boldsymbol{0}-$zero column vector of length $(n-1)$, and $\blacktriangleleft$ — the pointer of the position of the highest PrP f_n coefficient α_{n-1}.

$$\boldsymbol{G}_f^{(n)} = \begin{pmatrix} \boldsymbol{\alpha}_{n-1} & \boldsymbol{\alpha}_{n-2} & \boldsymbol{\alpha}_{n-3} & \cdots & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_1 & \boldsymbol{1} \\ \boldsymbol{1} & 0 & 0 & \cdots & 0 & 0 & \boldsymbol{0} \\ 0 & \boldsymbol{1} & 0 & \cdots & 0 & 0 & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \boldsymbol{0} \\ 0 & 0 & 0 & \cdots & \boldsymbol{1} & 0 & \boldsymbol{0} \\ 0 & 0 & 0 & \cdots & 0 & \boldsymbol{1} & \boldsymbol{0} \\ n & n-1 & n-2 & \cdots & 3 & 2 & 1 \end{pmatrix} \begin{matrix} n \\ n-1 \\ n-2 \\ \cdots \\ 3 \\ 2 \\ 1 \end{matrix}. \qquad (9)$$

In matrix (9), for clarity, the elements of the main diagonal of the identity matrix $\boldsymbol{E}$ and the bordering elements of this matrix are highlighted in bold (on the right – the zero column $\boldsymbol{0}$, and on top – the row, which is a primitive polynomial f_n reduced by one digit on the left).

Compact forms of Fibonacci matrices $\boldsymbol{F}_f^{(n)}$ are interconnected with Galois matrices $\boldsymbol{G}_f^{(n)}$ in configuration (8) by the operator of right-hand transposition

$$\boldsymbol{G}_f^{(n)} \xleftrightarrow{\perp} \boldsymbol{F}_f^{(n)} = \begin{pmatrix} \boldsymbol{\Theta} & f \\ \boldsymbol{E} & \blacktriangledown \end{pmatrix}, \qquad (10)$$

where $\boldsymbol{\Theta}$ — is the zero-row vector of the $(n-1)-$ order.

Let us give expressions for the $\boldsymbol{G}$ and $\boldsymbol{F}$ matrices of the eight-order.

$$\boldsymbol{G}=\begin{array}{c}i\\ \begin{pmatrix}0&1&1&0&0&1&0&1\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\end{pmatrix}\\ j\quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\end{array}\begin{array}{c}8\\7\\6\\5\\4\\3\\2\\1\end{array},\quad \boldsymbol{F}=\begin{array}{c}i\\ \begin{pmatrix}0&0&0&0&0&0&0&1\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&1\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&1\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}\\ j\quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\end{array}\begin{array}{c}8\\7\\6\\5\\4\\3\\2\\1\end{array}. \tag{11}$$

Structural-logic diagrams of Galois and Fibonacci LFSR generators, corresponding to relations (1)), are shown above in Figures 4 and 5, respectively. Supplementing the symbolic forms (8), (10) of the Galois $\boldsymbol{G}$ and Fibonacci $\boldsymbol{F}$ matrices with the corresponding conjugate matrices $\boldsymbol{G}^*$ and $\boldsymbol{F}^*$ formed by the left-hand transposition of the base matrices,

$$\boldsymbol{G}(\boldsymbol{F})\overset{T}{\longleftrightarrow}\boldsymbol{G}^*(\boldsymbol{F}^*)= =\begin{pmatrix}\blacktriangle & \boldsymbol{E}\\ f & \Theta\end{pmatrix}\left(\begin{pmatrix}\mathbf{0} & \boldsymbol{E}\\ f & \blacktriangleright\end{pmatrix}\right), \tag{12}$$

we arrive at the interconnection scheme (Fig. 9) of the subset of matrices, which we denote $\{\boldsymbol{Q}\}=\{\boldsymbol{G},\boldsymbol{F},\boldsymbol{G}^*,\boldsymbol{F}^*\}$.

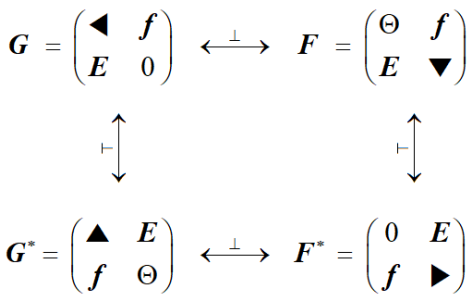

Fig. 9. The diagram of the relationship between primary and adjoint Galois and Fibonacci matrices

The conjugate eighth-order Galois $\boldsymbol{G}^*$ and Fibonacci $\boldsymbol{F}^*$ matrices generated by transformations (12) of matrices (11) have the form:

$$\boldsymbol{G}^*=\begin{array}{c}\begin{pmatrix}0&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0\\1&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\1&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\1&0&0&0&0&0&0&0\end{pmatrix}\\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\end{array}\begin{array}{c}8\\7\\6\\5\\4\\3\\2\\1\end{array};\quad \boldsymbol{F}^*=\begin{array}{c}\begin{pmatrix}0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\1&0&1&0&0&1&1&0\end{pmatrix}\\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\end{array}\begin{array}{c}8\\7\\6\\5\\4\\3\\2\\1\end{array}. \tag{13}$$

LFSR generators PRN, corresponding to matrices (13), are shown in Fig. 10.

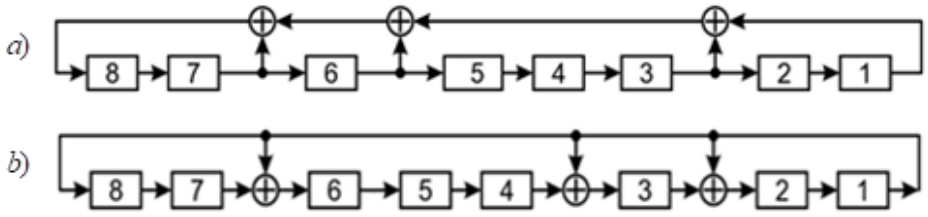

Fig. 10. Block diagrams of coupled PRN generator in configurations Galois $a)$ and Fibonacci $b)$ generated by PrP $f_8=101100101$

The term "feedback loop" in PRN generators can explain by their stylized graphical representation in Fig. 11. The PrP $f_8 = 101100101$ take as the generating polynomial.

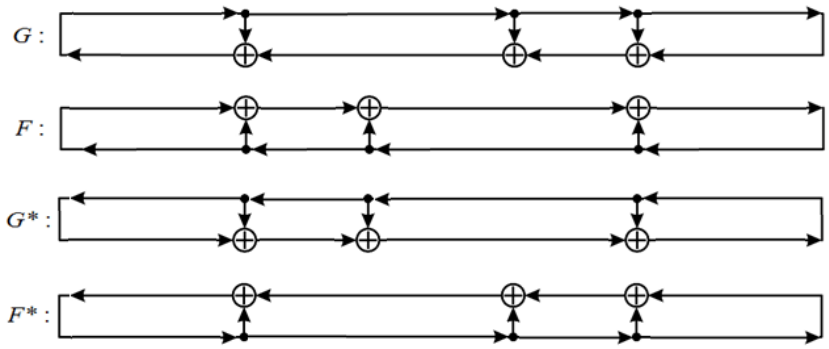

Fig. 11. A stylized representation of feedback schemes in PRN generators

Let's pay attention to such peculiarities of feedback. If the feedback loops of generators $\boldsymbol{G}$ and $\boldsymbol{F}$ wound clockwise, those of generators $\boldsymbol{G}^*$ and $\boldsymbol{F}^*$ wound counterclockwise. This fact can also be a sense in the block diagrams of the PRN generators shown in Figures 4, 5, and 10. The ways of transforming LFSR loops of PRN generators are present in Table 3. The table elements contain operators of loops rotation: $\leftrightarrow$ — relative to the vertical axis of symmetry and $\updownarrow$ — relative to the horizontal axis of symmetry.

Table 3: Relationship of $\{\boldsymbol{Q}\}$
LFSR feedback loops

	$\boldsymbol{G}$	$\boldsymbol{F}$	$\boldsymbol{G}^*$	$\boldsymbol{F}^*$
$\boldsymbol{G}$	–	$\leftrightarrow\updownarrow$	$\leftrightarrow$	$\updownarrow$
$\boldsymbol{F}$	$\leftrightarrow\updownarrow$	–	$\updownarrow$	$\leftrightarrow$
$\boldsymbol{G}^*$	$\leftrightarrow$	$\updownarrow$	–	$\leftrightarrow\updownarrow$
$\boldsymbol{F}^*$	$\updownarrow$	$\leftrightarrow$	$\leftrightarrow\updownarrow$	–

3 Efficient Algorithms for Calculating the States of Classical Matrix Galois

The algorithm's complexity for assessing the state of any of four PRN generators shown in Figure 9 is, according to relation (1), is $O(n^2)$, i.e., increases in quadratic dependence on the order of the classical Galois matrices. Based on the structures of the CGM (due to their components — the unit matrices $\boldsymbol{E}$ of $(n-1)-$order), it is possible to significantly reduce the computer time spent on assessing the state of the PRN generators at the next $(k+1)$ computation step.

For simplicity, let us introduce a notation system somewhat different from the one used earlier, assuming: $V_k = \{v_{n-1}, v_{n-2}, \ldots, v_1, v_0\}$ — the PRN vector at the k generation step, in the curly brackets of which the binary components of the vector indicated; $f_n = \{\alpha_n = 1, \alpha_{n-1}, \ldots, \alpha_1, \alpha_0 = 1\}$ — primitive polynomial generating CGM. The final relations that determine the vectors V_{k+1} for various CGMs summarize in Table 4. The arrows in Table 2, located to the right of the column vectors f_n and V_k, indicate the location of their senior element, and $\tilde{f}_n = \{\alpha_0, \alpha_1, \ldots, \alpha_n\}$.

Table 4. State vectors of classical matrix PRN generators

Matrices Galois	V_{k+1}		Matrices Fibonacci	V_{k+1}
$\boldsymbol{G}$	$\underbrace{\oplus^{v_{n-1}\cdot f_n \setminus \alpha_n, \alpha_0}_{V_k \setminus v_{n-1}}}_{n-1 \text{ bit}}, v_{n-1}$		$\boldsymbol{F}$	$\underbrace{V_k \setminus v_{n-1}}_{n-1 \text{ bit}}, \underbrace{\oplus\left(V_k \uparrow \otimes f_n \downarrow\right)}_{1 \text{ bit}}$
$\boldsymbol{G}^*$	$\underbrace{\oplus\left(V_k \uparrow \otimes f_n \uparrow\right)}_{1 \text{ bit}}, \underbrace{V_k \setminus v_0}_{n-1 \text{ bit}}$		$\boldsymbol{F}^*$	$v_0, \underbrace{\oplus^{V_k \setminus v_0}_{v_0 \cdot \tilde{f}_n \setminus \alpha_n, \alpha_0}}_{n-1 \text{ bit}}$

Let us confirm the correctness of the expressions given in Table 4. For example, let us calculate the vector V_{k+1} for the matrix $\boldsymbol{G}$. Let us write the general relation

$$V_{k+1} = V_k \otimes \boldsymbol{G}_f^{(n)}. \quad (14)$$

Substituting matrix (9) into (14), we have

$$V_{k+1} = (v_{n-1}, v_{n-2}, \ldots, v_1, v_0) \otimes \begin{pmatrix} \boldsymbol{\alpha}_{n-1} & \boldsymbol{\alpha}_{n-2} & \boldsymbol{\alpha}_{n-3} & \cdots & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_1 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \cdots & 0 & 0 & \mathbf{0} \\ 0 & \mathbf{1} & 0 & \cdots & 0 & 0 & \mathbf{0} \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & \cdots & \mathbf{1} & 0 & \mathbf{0} \\ 0 & 0 & 0 & \cdots & 0 & \mathbf{1} & \mathbf{0} \end{pmatrix}. \quad (15)$$

From formula (15), we uniquely arrive at the value of the vector-row, which locates at the intersection of row $\boldsymbol{G}$ and column V_{k+1} of Table 3. Similarly, expressions can determine for the remaining cells of Table 4.

Following the analysis of Table 4, we conclude that the proposed algorithm formation of the PRN is much simpler than those stated above. However, their computational complexity is $O(n)$, i.e., linearly depends on the order of Galois matrices forming generators of binary pseudorandom sequences.

4 Generalized Binary Hardware and Matrix PRN Generators

Definition 2. To the subset of generalized maximum period PRN generators, we will refer generators based on LFSR, covered by a multi-loop feedback circuit, which depends on the generating polynomial f_n (not necessarily primitive) and the generating element $\theta > 10$. One should choose a primitive element θ of the Galois field $GF(2^n)$ produced by an irreducible polynomial (IP) f_n as a generating element.

The Galois matrix $\boldsymbol{G}_{f,\theta}^{(n)}$, which use to programmatically form the same PRN as the sequence generated by the generalized LFSR generator, is called the generalized Galois matrix (GGM). Matrices synthesized by the rule similar to the regulation of CGM synthesis outlined in Section 2. Namely

Algorithm for the synthesis of GGM: Let f_n – an irreducible (not necessarily primitive) binary polynomial of degree n and $\theta > 10$ – the primitive element of the field $GF(2^n)$, generated by the polynomial f_n. Place θ in the lower right corner of the generated Galois matrix $\boldsymbol{G}_{f,\theta}^{(n)}$. All other digits of the bottom line $\boldsymbol{G}_{f,\theta}^{(n)}$, located to the element's left θ, are filled with zeros. Suppose that on the stage of formation of the following matrix row, its senior unit goes beyond the left boundary of the matrix. In that case, the polynomial located in this row gives by the remainder modulo f_n. Thus, the row returns to the matrix, and the formation process $\boldsymbol{G}_{f,\theta}^{(n)}$ continues further.

Let us consider examples of synthesis of a subset of primitive generalized Galois and Fibonacci matrices $\{\boldsymbol{Q}_g\} \in \{\boldsymbol{G}_g, \boldsymbol{F}_g, \boldsymbol{G}_g^*, \boldsymbol{F}_g^*\}$ and build on their basis the PRN generators of the maximum period. First, let's choose an irreducible binary polynomial of the fourth-degree $f_4 = 11111$, which is not primitive, and a primitive forming element (FE) equal to 111. Then, the matrices corresponding to the selected parameters have the form:

The block diagram of the generalized four-bit Galois generator corresponding to the GGM $\boldsymbol{G}_g$ shows in Fig. 12. The vertically arranged registers of the generators, marked at the top by the symbol $\otimes$, implement the operation of bitwise multiplication. In the registers is a saving symbol $\otimes$ — a function of adding the contents of the register modulo 2.

$$\boldsymbol{G}_g = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}; \qquad \boldsymbol{F}_g = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$\boldsymbol{G}_g^* = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}; \qquad \boldsymbol{F}_g^* = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{16}$$

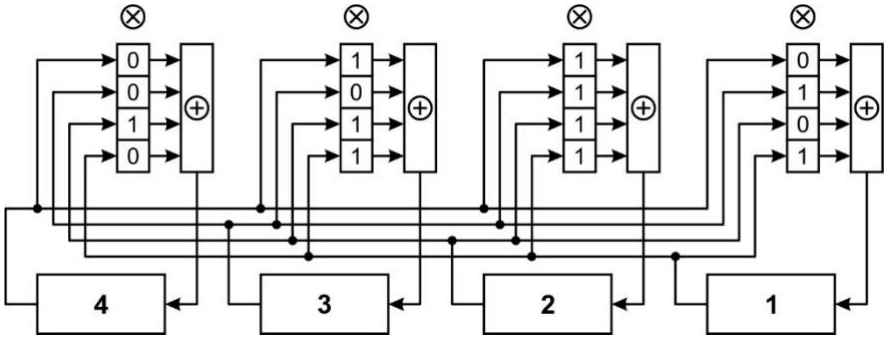

Fig. 12. Block diagram of the basic generalized Galois generator

Replacing in Figure 12 the contents of the cells of the vertical feedback registers by the elements of the matrix $\boldsymbol{G}_g^*$ from the system (16), we obtain the circuit (Fig. 13) of the conjugated generalized PRN generator in the Galois configuration. Block diagrams of the PRN generator shown in Fig. 12 and 13 are just examples of LFSR generators with multi-loop feedback.

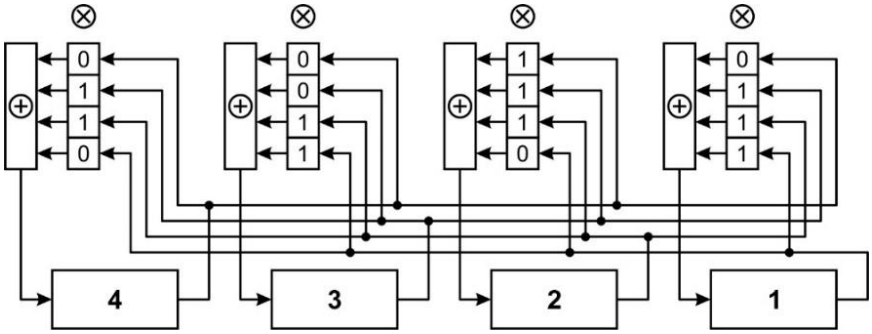

Figure 13. Block diagram of a conjugate generalized Galois generator

If in the graphs in Fig. 13 to replace the contents of the feedback register cells with matrix elements $\boldsymbol{F}$ and $\boldsymbol{F}^*$ from the (16), we come to the basic and conjugate generalized PRN generator schemes in the Fibonacci configuration.

The fundamental difference between generalized Galois matrices $\{\boldsymbol{Q}_g\}$ and classical matrices $\{\boldsymbol{Q}\}$ is as follows. If in CGM $\{\boldsymbol{Q}\}$ we can highlight the unit matrix $\boldsymbol{E}$ of (n-1)-order, the zero column-vector, and the row-vector, containing the bits of the generator polynomial f, that in generalized matrices $\{\boldsymbol{Q}_g\}$ does not have such features. It follows that there are no compact forms similar to the (8) of matrices $\{\boldsymbol{Q}_g\}$ for the set matrices. It follows that there are no compact forms similar to matrices $\{\boldsymbol{Q}_g\}$ represented by expression (8).

It is convenient to present a scheme of the interrelation of classical $\{\boldsymbol{Q}\}$ and generalized $\{\boldsymbol{Q}_g\}$ Galois matrices in Table 5.

Table 5. Interrelation of Galois and Fibonacci matrices

	G	F	G^*	F^*
G	–	$\perp$	T	$\perp T$
F	$\perp$	–	$\perp T$	T
G^*	T	$\perp T$	–	$\perp$
F^*	$\perp T$	T	$\perp$	–

5 Generalize Hardware and Matrix PNR Generators Over a Field of Odd Characteristics

The developed synthesis algorithms for binary matrix Galois PRN generators are easily generalized for constructing PRN generators over a field of odd characteristics p. The Galois matrices corresponding to such generators denote by $\boldsymbol{G}_{f,\theta,p}^{(n)}$. The matrix $\boldsymbol{G}_{f,\theta,p}^{(n)}$ synthesis algorithm coincides with the above algorithm for synthesizing binary GGMs $\boldsymbol{G}_{f,\theta}^{(n)}$. In this case, the algorithm is enough to perform only such simple replacements: $GF(2^n) \to GF(p^n)$ and $\boldsymbol{G}_{f,\theta}^{(n)} \to \boldsymbol{G}_{f,\theta,p}^{(n)}$.

Let us look at an example. Let $n=4$, $p=3$, $f=12121$ and $\theta=221$. The parameters include an irreducible polynomial f, the exponent of 10, and $\theta-$ a primitive element of the field $GF(3^4)$, generated by the IP f. The selected parameters correspond to the generalized primitive Galois and Fibonacci matrices over $GF(3)$

$$\boldsymbol{G}=\begin{bmatrix}0&1&2&2\\1&2&2&1\\2&2&1&0\\0&2&2&1\end{bmatrix},\quad \boldsymbol{F}=\begin{bmatrix}1&0&1&2\\2&1&2&2\\2&2&2&1\\0&2&1&0\end{bmatrix},$$

$$\boldsymbol{G}^*=\begin{bmatrix}0&1&2&0\\1&2&2&2\\2&2&1&2\\2&1&0&1\end{bmatrix},\quad \boldsymbol{F}^*=\begin{bmatrix}1&2&2&0\\0&1&2&2\\1&2&2&1\\2&2&1&0\end{bmatrix}, \tag{17}$$

in which letter indices are omitted for simplicity.

Using the matrix $\boldsymbol{G}$ of the system (17) and the generator circuit shown in Fig. 14, we will compose a generalized structural logic diagram (Figure 14) of a ternary four-bit register PRN generators in the Galois configuration. The numbers 3 located in the bitwise multiplication and addition operators mean that the calculations were modulo 3. It also assumed that the register $D-$ triggers transfer ternary numbers from the input to the output.

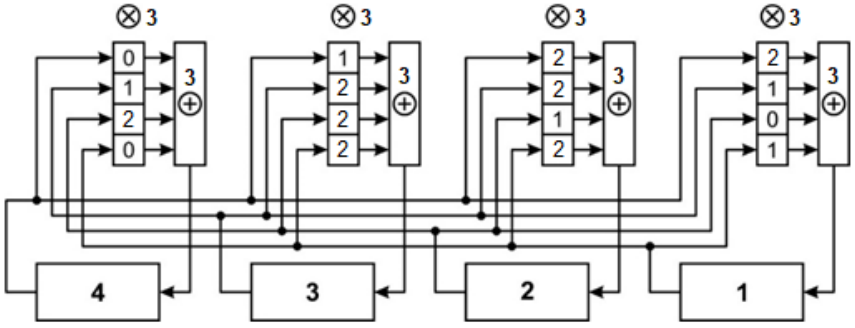

Fig. 14. Block diagram of the generalized Galois generator over IP $f=12121$

An alternative register generator shown in Fig. 14 is a matrix PRN generator, which generates the same sequence of pseudorandom ternary codes (Table 6) as a registered generator.

Table 6. A sequence of ternary vectors generated by the registered (Fig. 14) and matrix $\boldsymbol{G}_{f,\theta,p}^{(n)}$ ($\theta = 221$) generators of the PRN over the IP $f = 12121$

1		**0**	**0**	**0**	**1**		0	0	1	0		0	1	0	0		1	0	0	0		1	2	1	2
2		0	2	2	1		2	2	1	0		1	2	2	1		0	1	2	2		1	2	2	0
3		0	1	2	0		1	2	0	0		0	2	1	2		2	1	2	0		0	0	2	1
4		2	0	1	1		2	2	0	1		1	1	0	1		2	2	2	2		1	0	1	1
5		2	0	1	2		2	2	1	1		1	2	0	1		0	2	2	2		2	2	2	0
6		2	2	0	0		1	1	2	1		2	1	2	2		0	0	1	1		0	1	1	0
7		2	0	2	0		2	0	2	1		2	0	0	1		2	1	0	1		0	1	0	1
8		1	0	0	1		1	2	2	2		0	1	0	2		1	0	2	0		1	1	1	2
																						0	**0**	**0**	**2**

Table 6 contains only the first half of the sequence of the maximum period, consisting (for the selected values of the generator parameters) of 80 ternary four-digit codes. The second half of the series, starting with code 0002, is formed from codes of the first half due to their bitwise multiplication by 2 modulo 3.

6. Isomorphism of Generalized Galois Matrices

The theory of polynomials of one variable x knows that multiplication of an arbitrary degree k polynomial $\omega_k(x)$ by the x equivalent of its shift by one digit to the left. Or, in other words,

$$x \cdot \omega_k(x) \to \omega_{k+1}(x). \qquad (18)$$

Using ratio (18) and taking into account how GGM formed, record the transformation chain

$$\boldsymbol{G}_{f,\omega}^{(n)} \Rightarrow \begin{bmatrix} x^{n-1}\cdot\omega \\ x^{n-2}\cdot\omega \\ \cdots \\ x\cdot\omega \\ 1\cdot\omega \end{bmatrix} \bmod f_n = \omega \cdot \begin{bmatrix} x^{n-1} \\ x^{n-2} \\ \cdots \\ x \\ 1 \end{bmatrix} \bmod f_n. \qquad (19)$$

The elements of the right vector-column of inequality (19) are monomials, which, when

$$\begin{bmatrix} x^{n-1} \\ x^{n-2} \\ \cdots \\ x \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \boldsymbol{E}_n \quad (20)$$

,

which makes it possible to formulate the following statement.

Affirmation. The GGM $\boldsymbol{G}_{f,\omega}^{(n)}$ of the order n above IP f_n isomorphous to its constitutive element ω, which belongs to the field $GF(2^n)$, i.e.

$$\boldsymbol{G}_{f,\omega}^{(n)} \simeq \omega, \qquad (21)$$

presented in binary form, transform the column into a unit matrix, i.e.
where $\simeq$ – the isomorphism symbol.

According to the expressions (19), (20), there is a mutually unambiguous correspondence (isomorphism) between GGM $\boldsymbol{G}_{f,\omega}^{(n)}$ and its forming element ω, which reflects by the ratio (21) and leads to such consequences:

Consequence 1. The generalized matrixes of Galois $\boldsymbol{G}_{f,\omega}^{(n)}$ are non-singular at any parameters f_n and ω, as are formed by linearly independent lines.

Consequence 2. For elevating the matrix $\boldsymbol{G}_{f,\omega}^{(n)}$ the degree k is enough to calculate forming element $\omega_k = \omega^k (\bmod f_k)$ and make a matrix $\boldsymbol{G}_{f,\omega}^{(n)}$.

Consequence 3. The minimum non-zero value of degree e providing equality $\left(\boldsymbol{G}_{f,\omega}^{(n)}\right)^e = \boldsymbol{E}$ coincides

with the order of the element ω, which forms the matrix $\boldsymbol{G}_{f,\omega}^{(n)}$

Consequence 4. The generalized matrix of Galois $\boldsymbol{G}_{f,\omega}^{(n)}$ is primitive if the element forming ω it is primitive, i.e., if $\omega = \theta$, there is θ — a primitive element of the field $GF(2^n)$.

Consequence 5. The operation of multiplication matrixes Galois $\boldsymbol{G}_{f,\omega_1}^{(n)}$ and $\boldsymbol{G}_{f,\omega_2}^{(n)}$, $\omega_1 \neq \omega_2$, is commutative because according to the ratio (21) of the product in the left and right parts of the equality $\boldsymbol{G}_{f,\omega_1}^{(n)} \cdot \boldsymbol{G}_{f,\omega_2}^{(n)} = \boldsymbol{G}_{f,\omega_2}^{(n)} \cdot \boldsymbol{G}_{f,\omega_1}^{(n)}$ is equivalent to the products of elements $(\omega_1 \cdot \omega_2)$ and $(\omega_2 \cdot \omega_1)$, calculated on the module of the IP f_n.

Consequence 6. Arbitrary modular arithmetic transformations over Galois matrixes are isomorphic to the same changes over the forming elements of these matrixes.

Consequence 7. The generalized matrixes of Galois $\bar{\boldsymbol{G}}_{f,\omega}^{(n)}$, inverse matrix $\boldsymbol{G}_{f,\omega}^{(n)}$, can be constructed according to the rule formulated in item 4. The forming element of the matrix $\bar{\boldsymbol{G}}_{f,\omega}^{(n)}$ is element $\bar{\omega}$, the inverse of the forming element matrix $\boldsymbol{G}_{f,\omega}^{(n)}$.

7 Calculating Inverse Elements of the Galois Field

For each "direct" matrix from subset $\{\boldsymbol{Q}\} \in (\boldsymbol{G}, \boldsymbol{F}, \boldsymbol{G}^*, \boldsymbol{F}^*)$, we will match the so-called "reverse" matrices, the set of which forms subset $\{\bar{\boldsymbol{Q}}\} \in (\bar{\boldsymbol{G}}, \bar{\boldsymbol{F}}, \bar{\boldsymbol{G}}^*, \bar{\boldsymbol{F}}^*)$. Let us supplement the internal matrix contour $\{\boldsymbol{Q}\}$, shown in Figure 9, another so-called external contour, placing the matrixes of the subset in its nodes $\{\bar{\boldsymbol{Q}}\}$. The posed problem has a trivial solution. Indeed, according to (21) Galois matrices $\boldsymbol{G}_f^{(n)}$ and their forming elements are connected by the isomorphism relation. And, as a consequence, two Galois matrices generated by the same non-acceptance (primitive for CGM) polynomial become mutually convertible if the elements forming them are mutually convertible. Therefore, to construct $\bar{\boldsymbol{G}}_{f,\theta}^{(n)}$ a reverse matrix $\boldsymbol{G}_f^{(n)}$, it is sufficient to replace forming element θ with its reverse value $\bar{\theta}$ at the stage of matrix synthesis, i.e., $\bar{\boldsymbol{G}}_{f,\theta}^{(n)} = \boldsymbol{G}_{f,\bar{\theta}}^{(n)}$.

For CGM, generated by PrP f_n, the forming element $\theta = 10$. By definition, some non-zero element ω of the Galois extended field is a reverse element $\bar{\omega}$ if and only if the condition met $(\omega \cdot \bar{\omega}) \bmod f = 1$. Let $f_n = 1\alpha_{n-1}\alpha_{n-2},\ldots,\alpha_1 1$ — the primitive binary polynomial and $(\theta = 10)$ — forming element matrix $\boldsymbol{G}_{f,\omega}^{(n)}$. Then $\bar{\theta} = 1\alpha_{n-1}\alpha_{n-2},\ldots,\alpha_1$ and $\bar{\theta} \cdot \theta = 1\alpha_{n-1}\alpha_{n-2},\ldots,\alpha_1 0$. The product $\bar{\theta} \cdot \theta$ by modulo f_n forms a subtraction of 1, required for a pair of reciprocal values. Thus, we come to the following general form of inverse CGM

$$\bar{\boldsymbol{G}}_f^{(n)} = \begin{pmatrix} \mathbf{0} & \mathbf{1} & 0 & \cdots & 0 & 0 & 0 \\ \mathbf{0} & 0 & \mathbf{1} & \cdots & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & 0 & 0 & \cdots & 0 & \mathbf{1} & 0 \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 & \mathbf{1} \\ \mathbf{1} & \boldsymbol{\alpha}_{n-1} & \boldsymbol{\alpha}_{n-2} & \cdots & \boldsymbol{\alpha}_3 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_1 \end{pmatrix} \begin{matrix} n \\ n-1 \\ n-2 \\ \cdots \\ 3 \\ 2 \\ 1 \end{matrix}, \quad (22)$$

$$n \quad n-1 \quad n-2 \quad \cdots \quad 3 \quad 2 \quad 1$$

which can present in such a compact form

$$\bar{\boldsymbol{G}}_f^{(n)} = \begin{pmatrix} \mathbf{0} & \boldsymbol{E} \\ f & \blacktriangleleft \end{pmatrix}.$$

In particular, for PrP $f_8 = 101100101$ under (22), we will receive

$$\bar{G} = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ j \end{matrix} \begin{matrix} i \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{matrix} \begin{matrix} \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{matrix}. \quad (23)$$

Hardware implementation of the LFSR generator's PRN, to which the matrix (23) responds, is shown in Fig. 15.

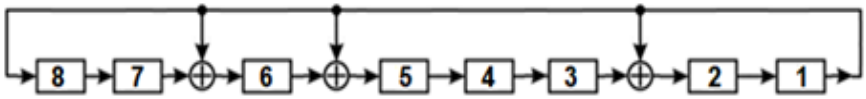

Fig. 15. Block diagram of the "reverse" generator PRN in the Galois configuration generated by the PrP $f_8 = 101100101$

The interrelation of Galois $\{\bar{Q}\}$ inverse matrices is determined by the same operators $\perp$ and T performed according to the same scheme as direct matrices $\{Q\}$. The scheme of the interrelation of the complete set of matrices $\{\bar{Q}\}$ is shown in Fig. 16.

$$\begin{array}{ccc} \bar{G} = \begin{pmatrix} \mathbf{0} & \boldsymbol{E} \\ f & \blacktriangleleft \end{pmatrix} & \xleftrightarrow{\perp} & \bar{F} = \begin{pmatrix} \blacktriangledown & \boldsymbol{E} \\ f & \Theta \end{pmatrix} \\ \Big\updownarrow \mathrm{T} & & \Big\updownarrow \mathrm{T} \\ \bar{G}^* = \begin{pmatrix} \Theta & f \\ \boldsymbol{E} & \blacktriangle \end{pmatrix} & \xleftrightarrow{\perp} & \bar{F}^* = \begin{pmatrix} \blacktriangleright & f \\ \boldsymbol{E} & \mathbf{0} \end{pmatrix} \end{array}$$

Fig. 16. The scheme of the interconnection of the set of classical Galois matrices

The matrix (23) is converted into a reverse Fibonacci matrix by right-hand transposition.

$$\bar{F} = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ j \end{matrix} \begin{matrix} i \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{matrix} \begin{matrix} \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{matrix}. \quad (24)$$

Hardware implementation of the LFSR of the PRN generator to which the matrix (24) responds shows in Fig. 17.

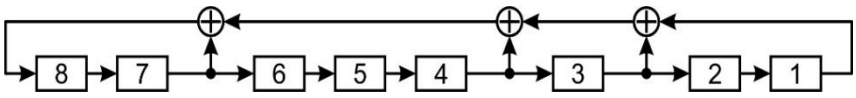

Fig. 17. Block diagram of the "reverse" generator PRN in the Fibonacci configuration generated by the PrP $f_8 = 101100101$

Reverse conjugate Galois $\bar{\boldsymbol{G}}^*$ and Fibonacci $\bar{\boldsymbol{F}}^*$ matrices of the eighth order are generated, according to transformations 13), by left-hand transposition of matrices (23), (24), and have a form:

$$\bar{\boldsymbol{G}}^* = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&1 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \begin{matrix} 8 \\ 7 \\ 6 \\ 5 \\ 4; \\ 3 \\ 2 \\ 1 \end{matrix} \quad \bar{\boldsymbol{F}}^* = \begin{pmatrix} 0&1&0&0&1&1&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \begin{matrix} 8 \\ 7 \\ 6 \\ 5 \\ 4. \\ 3 \\ 2 \\ 1 \end{matrix} \quad (25)$$

$$\begin{matrix} 8&7&6&5&4&3&2&1 \end{matrix} \qquad \begin{matrix} 8&7&6&5&4&3&2&1 \end{matrix}$$

Structural diagrams of inverse conjugate LFSR generators of PRN, corresponding to matrixes (25), are presented in Fig. 18.

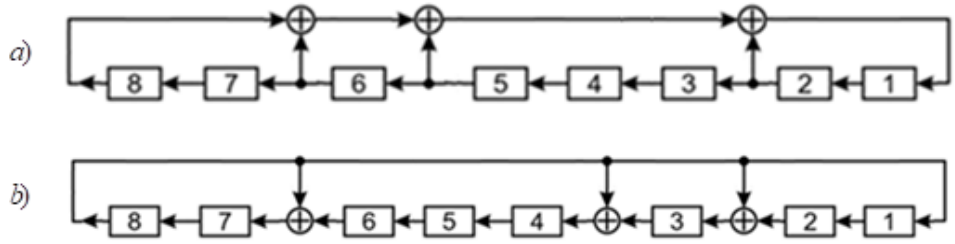

Fig. 18. in the of reverse conjugate generators of PRN in Galois (*a*) and Fibonacci (*b*) configurations, generated by PrP $f_8 = 101100101$

The main problem in the designated calculation chain is the definition of the element $\bar{\omega}$. There are different ways of finding the inverse elements of the Galois field [12]. The most frequently used method is based on the extended Euclidian algorithm [13]. Below is an alternative approach to calculations of $\bar{\omega}$ that is easier to implement than the Euclidian algorithm.

It is known that for any non-zero element ω of a binary Galois field, the equality

$$\left(\omega^{L_n}\right)_{f_n} = \left(\omega^{2^n-1}\right)_{f_n} = 1, \qquad (26)$$

where L_n is the order of the element ω, and $(a)_f = a \pmod f$.

Introducing (26) in the form

$$\left(\omega^{2^n-1}\right)_{f_n} = \left(\omega\left(\omega^{2^n-2}\right)\right)_{f_n}, \\ = \left(\omega\cdot\bar{\omega}\right)_{f_n} = 1$$

we'll get

$$\bar{\omega} = \left(\omega^{2^n-2}\right)_{f_n}. \qquad (27)$$

According to formula (27), the inverse element $\bar{\omega}$ is determined by the residue of the even degree $2^n - 2$ of the field $GF(2^n)$ element ω modulo f_n. These residues are placed in the odd lines of Table 7.

Table 7. The algorithm for calculating inverse elements of a binary Galois field

n	L_n	k	Residue		n	L_n	k	Residue
VI		1	$\Delta_1 = (\omega^2)_f$		6	63	8	$\Delta_8 = (\Delta_7 \cdot \omega)_f$
3	7	2	$\Delta_2 = (\Delta_1 \cdot \omega)_f$				9	$\Delta_9 = (\Delta_8^2)_f$
		3	$\Delta_3 = (\Delta_2^2)_f$		7	127	10	$\Delta_{10} = (\Delta_9 \cdot \omega)_f$
4	15	4	$\Delta_4 = (\Delta_3 \cdot \omega)_f$				11	$\Delta_{11} = (\Delta_{10}^2)_f$
		5	$\Delta_5 = (\Delta_4^2)_f$		8	255	12	$\Delta_{12} = (\Delta_{11} \cdot \omega)_f$
5	31	6	$\Delta_6 = (\Delta_5 \cdot \omega)_f$				13	$\Delta_{13} = (\Delta_{12}^2)_f$
		7	$\Delta_7 = (\Delta_6^2)_f$					

Table 7 it is indicated: $n-$ the degree of the irreducible polynomial f; k – step of iteration; L_n – the order of the multiplicative group of the field $GF(2^n)$, generated by IP f; *VI* – initialization vector equal to $(\omega^2)_f$.

Based on Table 7, we quickly come to the expression for the number of iterations k, performed when calculating the inverse field elements $\overline{\omega}$ over the IP degree n

$$k = 2n - 3.$$

Let's consider a numerical example. Suppose $n = 4$, $f = 10011$ and $\omega = 1101$. According to Table 7, the first step is the calculation

$$\Delta_1 = \left(\omega^2\right)_f = \left(1101 \cdot 1101\right)_{10011} = 1110$$

For the next step, we find

$$\Delta_2 = \left(\Delta_1 \cdot \omega\right)_f = \left(1110 \cdot 1101\right)_f = \left(1000110\right)_f = 1010;$$
$$\Delta_3 = \left(\Delta_2^2\right)_f = \left(1010 \cdot 1010\right)_f = \left(1000100\right)_f = 1000;$$
$$\Delta_4 = \left(\Delta_3 \cdot \omega\right)_f = \left(1000 \cdot 1101\right)_f = \left(1101000\right)_f = 10;$$
$$\Delta_5 = \left(\Delta_4^2\right)_f = \left(10 \cdot 10\right)_f = 100.$$

Residue $\Delta_5 = 100$ is the opposite of the element $\omega = 1101$.

The vector of initialization $VI = \Delta_1 = \left(\omega^2\right)_f$ starts the computational process. The further procedure consists of $n-2$ cycles, each of which includes two iteration steps. We find the auxiliary vector $\Delta_{2(n-2)}$ as the first (on the even step k) and the second (on the odd iteration stage) — the inverse element $\overline{\omega} = \Delta_{2n-3}$.

The algorithm for calculating the inverse elements of the field $GF(2^n)$ can easily be generalized to determine the inverse elements of an arbitrary characteristic p. The block scheme of the algorithm shows in Fig. 19.

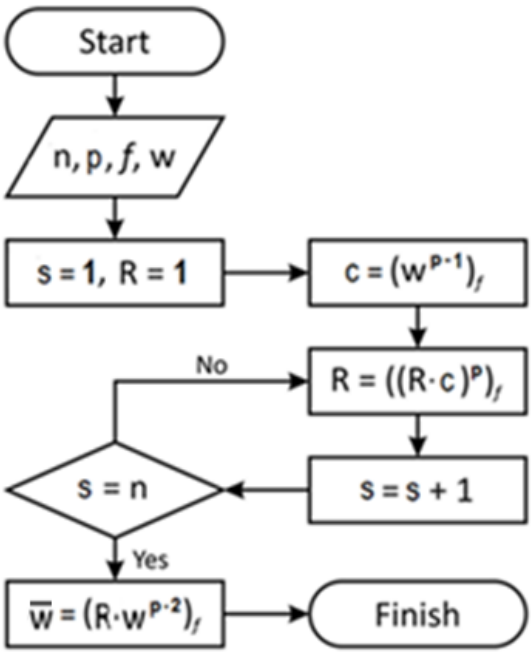

Fig. 19. Block scheme of the algorithm for calculating inverse field $GF(p^n)$ elements.

For example, let's $p=5$, $n=3$, $f=1032$, $\omega=234$. The sequence of results of calculations the inverse element $\bar{\omega}$ is as follows:

1. $s=1$, $R=1$, $c=\left(\omega^4\right)_f=(234)_f^4=24$, $R=\left(R\cdot c\right)_f^5=\left(24^5\right)_f=131$;
2. $s=2$, $R=\left(R\cdot c\right)_f^5=\left(131\cdot 24\right)_f^5=423$;
3. $s=3$, $\bar{\omega}=\left(R\cdot\omega^3\right)_f=\left(423\cdot 234^3\right)_f=202$; $\left(\omega\cdot\bar{\omega}\right)_f=\left(234\cdot 202\right)_f=\mathbf{1}$.

8 Hacking Problems of Galois PRN Generators

Its knowns that the classic Galois LFSR generators have lower crypto stability; the reason for which is that they quickly hacked using the Berlekamp-Massey (BM) algorithm [9]. This algorithm uses the known elements of the sequence $\boldsymbol{X}=\{x_1,x_2,\ldots,x_{2n}\}$ produced by a $n-$discharge oscillator to calculate the PrP f_n in the feedback circuit of a linear register of minimum length n. It should note that all primitive Galois matrices, both classical and generalized, can serve as generators of PRN length sequences $L=2^n-1$. Each of these sequences, removed from the output of an arbitrary discharge LFSR, satisfies all three postulates of Golomb [10]. For this reason, it may seem that the generalized Galois generators do not bring any new properties to the PRN formed by classical generators. But this is not entirely true. As established in [6], generators of PRN built based on generalized Galois matrices are free from the BM attack. The noted feature of the generators appears due to the following reason. The classical generators by mean of BM algorithm are successfully determining only one unknown – primitive polynomial f_n. For generalized generators, besides the polynomial f_n, the primitive FE θ of the Galois matrix is also unknown. But the classical algorithm of the BM is not intended for calculating two unknown parameters and therefore becomes unsuccessful in attacking the generalized generators. That, for one thing. And secondly, in any case (whether the conditions of applicability of the BM algorithm are satisfied or not) the processor implements the Berlekamp-Massey algorithm outputs as a solution that or the value of degree n primitive polynomial.

Below it will show that the transition from classical LFSR-generators of PRN to generators based on generalized matrixes of Galois and Fibonacci leads to the fact that the algorithm of BM loses the ability to determine the IP is produces the generator of PRN. The noted feature of such generators is that the series of bits formed by them

depends not only on the chosen IP f but also on the primitive element θ involved in the formation of the feedback chain of the generator.

For experimental confirmation of the stated statement and the basic theoretical positions concerning properties of matrixes of feedback, we shall address to results of computer modeling (reduced in Table 8) of the generalized eight-digit Galois generator of PRN. The PrP $f = 100011101$ was chosen as the polynomial forming the feedback loop of the generator.

Table 8. BM tester solutions on many primitive elements of the field generated by the PrP $f = 100011101$

№	**IP: 100011101**	Forming element							
	PrP	1	2	3	4	5	6	7	8
1	100011101	002	004	020	035	114	137	205	235
2	100101011	006	015	024	121	207	302	321	332
3	100101101	113	033	210	130	220	227	300	336
4	101001101	112	123	211	233	307	313	322	325
5	101011111	037	122	110	232	306	312	323	324
6	101100011	036	102	111	133	215	225	237	311
7	101100101	022	103	030	132	214	224	236	310
8	101101001	022	023	030	031	134	135	200	201
9	101110001	011	036	101	107	203	216	314	330
10	110000111	050	064	071	074	077	171	273	345
11	110001101	052	060	143	151	242	274	367	370
12	110101001	043	161	166	172	245	252	260	340
13	111000011	042	160	167	173	244	253	261	341
14	111001111	157	176	262	267	354	360	363	372
15	111100111	062	155	257	343	350	352	356	376
16	111110101	053	061	142	150	243	275	366	371

According to Table 8, the eight forming elements located in the top row of the table are such that each of them leads to the correct solution produced by the BM tester. We will call such forming elements "weak keys" of the flow code, the encrypting gamma formed by the analyzed PRN generator. It is quite easy to eliminate weak keys. For this purpose, it is enough to choose a polynomial f that is not primitive while keeping the forming element primitive θ.

9 Conclusion

The main results are as follows:

1. Different variants of construction of binary PRN generators based on the so-called generalized Galois and Fibonacci matrices developed. The transition from classical to generalized matrix generators PSF is accompanied by an expansion of diversity of generators and leads to a significant increase in their cryptographic strength. This effect is achieved by increasing the number of form elements and by the polynomials generating Galois matrices, which are not necessarily primitive.

2. It is shown that PRN generators based on generalized Galois matrices are not subject to BM attacks. The noted property is a consequence of this feature of the BM algorithm. The classical BM algorithm solves the problem of computing one unknown parameter: the minimal primitive polynomial f in the LFSR feedback circuit of the PRN generator. In the generalized matrix generators, two unknown parameters have to be determined: an irreducible polynomial f and a generating element θ, jointly forming Galois matrices. This problem becomes unsolvable for the BM algorithm.

3. Recurrence estimates of the states of classical matrix Galois generators of PRN have been proposed, significantly increasing the computational speed.

4. The developed algorithms of synthesis of generalized Galois and Fibonacci matrices allow to build cryptographically secure information protection systems and be useful in other applications.

References:

[1] Schneier B., *Applied cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York (1996).

[2] Chen L., Gong G. *Pseudorandom Sequence (Number) Generators, Communication Systems Security*, Appendix A, (2008).

[3] Ivanov M.A. *Cryptographic methods of information protection in computer systems and networks.* M.: KUDITS-OBRAZ, 2001. – 386 p. (In Russia)

[4] Jun Choi, Dukjae Moon, Seokhie Hong and Jaechul Sung. *The Switching Generator: New Clock-Controlled Generator with Resistance against the Algebraic and Side-Channel Attacks.* Entropy 2015, 17, pp. 3692-3709.

[5] Beletsky A. *Synthesis of Cryptoresistant Generators of Pseudorandom Numbers Based on Generalized Galois and Fibonacci Matrixes.* Radio Electronics, Computer Science, Control, (2019). Vol 3(50), pp. 86-98. (In Russia)

[6] Beletsky A., Beletsky E. *Generators of Pseudo Random Sequences of Galois.* Electronics and Control Systems, (2014, # 4(42). – P. 116-127. (In Russia)

[7] Mullajonov R.V. *Generalized transposition of matrices and structures of linear large-scale systems.* Reports of the National Academy of Sciences of Ukraine, 2009, №10. – P. 27-35. (In Russia)

[8] Gantmacher F.R. *Theory of Matrices*. — AMS Chelsea Publishing: Reprinted by American Math. Society, 2000. — 660 p.

[9] Beker H. and Piper F. *Cipher Systems: The Protection of Communication*, London: Northwood Books, 1982

[10] Matsumoto M. and Nishimura T. *Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator*. ACM Transactions on Modeling and Computer Simulation, 8:3–30, 1998.

[11] Meyer C.H. end Tuchman W.I. *Pseudorandom Codes Can be Cracked*, Electronic Design, v. 23, Nov 1972.

[12] Smart N. *Cryptography: An Introduction*, 3rd ed. McGraw-Hill College, 2013

[13] Van der Warden, B., L. *Mathematics Statistic*. Moscow, IL, 1960, 371 p. (In Russia)

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.