Variety of Matrix Galois-like Generators Pseudorandom Number Free from the Berlecamp-Messy Attack

ANATOLY BELETSKY
Department of Electronics,
National Aviation University,
1, Prospect Lyubomyra Guzara, Kyiv,
UKRAINE

Abstract: - Classical pseudorandom numbers generators (PRN) based on Galois and Fibonacci schemes are constructed, as a rule, using *n*-bit linear shift registers or corresponding *n*-order matrices and allowing both hardware and software implementation. The main disadvantage of such generators is their low crypto stability, the reason for which is that if by any means it is possible to obtain 2*n* bits of the generated sequence taken from any discharge of the generator, then with the help of the Berlecamp-Messy algorithm, it is possible to recover a primitive polynomial of *n*-degree f_n, generating the generator. To increase the cryptostability of the PRN matrix generators is proposed to replace the classical Galois and Fibonacci matrices, uniquely determined by the primitive polynomial f_n, at a fixed forming element ω, equal to 10, by the so-called generalized Galois or Fibonacci matrices. A distinctive feature of generalized matrices is that the polynomials f_n generating them need not be primitive. At the same time, the constituent elements ω must be chosen from the subset of primitive elements of the deduction field generated by the polynomial f_n. The generalized PRN generators are free from the Berlecamp-Messy attack. The latter property is obtained because the Berlecamp-Messy algorithm solves the problem of computing one single unknown - the primitive polynomial f_n, generating the generator. For variants of generalized matrix generators of PRN, there is a need to determine two unknown parameters: both the irreducible polynomial f_n and the forming element ω, jointly generating the generalized matrix, which turns out to be an unsolvable problem for the Berlecamp-Messy algorithm.

Key-Words: - Pseudorandom number generators, Galois and Fibonacci matrices, Berlecamp-Messy algorithm.

Received: June 26, 2022. Revised: August 14, 2023. Accepted: September 17, 2023. Published: October 23, 2023.

1 Introduction

In the theory and practice of noise-resistant coding, [1], [2], [3], [4], cryptographic information protection, [5], [6], [7], and in other areas of science and technology, pseudorandom sequence generators (PRNs) of maximum length with acceptable statistical characteristics are widely used. The most popular applications are two main methods for PRN generation. The first is based on using *n*-bit linear feedback shift registers (LFSR) according to Galois or Fibonacci schemes, [8], and the second one relies on *n*-order square matrices, which, by analogy with the names of register generators we will call Galois and Fibonacci matrices, [9], [10]. The matrix generators form the same PRNs as the corresponding register generators.

Structural and logical schemes of binary LFSR generators PRN are uniquely determined by *generating polynomials* f_n of *n*-degree (coinciding with the number of register digits), employing which the feedbacks in the shift registers establish. It is known, [11], that for a linear shift register to be a maximum period register equal to 2^n-1, the corresponding feedback polynomial f_n must be primitive (PrP).

Based on the construction of PRN register generator structural-logic circuits, we will use the natural ordering of register digits, in which the lowest digit is located on the right, as it is accepted when writing down numbers in positional number systems. The problem of developing structural-logic schemes of Galois generators of the PRN is most easily solved, the technology of construction of which we will illustrate in the example of the generator, the feedback circuit of which is defined by a PRP of the eighth degree $f_8 = 101100101$. The solution of the set task implies fulfilling such two stages, [9], [10].

Step 1. Form an eight-bit circular shift register (Figure 1) in the nodes of its feedback line and equidistantly arrange the digits of the selected primitive polynomial.

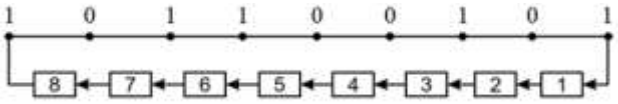

Fig. 1: To construct the circuit of the eight-digit Galois oscillator PRN

Step 2. Connecting the unit nodes of the feedback line with the *XOR* operator, as shown in Figure 2, we complete the construction of the classical LFSR generator PRN.

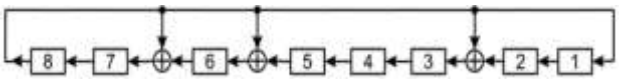

Fig. 2: Structural diagram of the Galois generator of PRN generated by PrP $f_8 = 101100101$

Each LFSR generator of the PRNs by the Galois scheme answers by uniquely related matrices, which we will call *classical Galois matrices* (CGMs) and denote by the symbol $\boldsymbol{G}_f^{(n)}$, where n is the order of the matrix, and f_n is the PrP of n-degree that generates the CGM. Based on the PRN generator circuit shown in Figure 1, we quickly arrive at the general form of the CGM matrix

$$\boldsymbol{G}_f^{(n)} = \begin{pmatrix} \alpha_{n-1} & \alpha_{n-2} & \dots & \alpha_2 & \alpha_1 & \mathbf{1} \\ 1 & 0 & \cdots & 0 & 0 & \mathbf{0} \\ 0 & 1 & \cdots & 0 & 0 & \mathbf{0} \\ \dots & \dots & \ddots & \dots & \dots & \dots \\ 0 & 0 & \cdots & 1 & 0 & \mathbf{0} \\ 0 & 0 & \cdots & 0 & 1 & \mathbf{0} \\ n-1 & n-2 & \cdots & 3 & 2 & 1 \end{pmatrix} \begin{matrix} n-1 \\ n-2 \\ n-3 \\ \dots \\ 2 \\ 1 \end{matrix}, \quad (1)$$

where $\alpha_i \in (0,1)$, $i = \overline{1, n-1}$, are the internal coefficients of PrP f_n, i.e., the coefficients located between the units enclosing the binary generating polynomial.

In the following, we will omit the numbering of rows and columns of matrices for simplicity.

By bolding the top row and the right column in (1), we represent the matrix $\boldsymbol{G}_f^{(n)}$ in a compact form

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

where $\boldsymbol{E}$ and $\mathbf{0}$ are the unit matrix, the zero-vector column of (n-1)-orders and the arrow indicates the position of the high internal coefficient α_{n-1} of the generating polynomial f_n.

Let us pay attention to such features of the matrix (1). First, the lower row of the matrix contains the *forming element* (FE) θ, equal to 10. Secondly, each subsequent row of the matrix, except for the top row, is obtained by shifting the previous row to the left by one digit. Third, the top row of the matrix (1) represents the PrP f_n, where the highest (left) unit is discarded. The above brief explanation makes it possible to formulate

Algorithm of CGM synthesis: In the right corner of the bottom line of the synthesized n-order CGM $\boldsymbol{G}_f^{(n)}$, the element forming it $\theta = 10$, which is the minimal primitive element of the field $GF(2^n)$, is generated by the binary PrP of n-degree f_n. Discharges in a string to the left of θ are filled with zeros. Subsequent rows of the matrix $\boldsymbol{G}_f^{(n)}$ (from bottom to top) are obtained by shifting the previous row one digit to the left. If, when moving a row, its highest unit digit goes outside the matrix (which is the upper row of the matrix $\boldsymbol{G}_f^{(n)}$), then the (n+1)-bit vector 100…0 corresponding to this row is reduced to the remainder modulo PrP f_n and, thus, the row becomes n-digit.

The matrices $\boldsymbol{G}_f^{(n)}$ are *primitive* in that the sequence of degrees of the matrices over the field $GF(2)$ forms a sequence of maximal length (m-sequence).

By involutional *right-side transposition* (rotation of a square matrix to the auxiliary diagonal) CGMs (1) transform into classical Fibonacci matrices (CFMs)

$$\boldsymbol{F}_f^{(n)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 1 & 0 & \cdots & 0 & 0 & \alpha_1 \\ 0 & 1 & \cdots & 0 & 0 & \alpha_2 \\ \dots & \dots & \ddots & \dots & \dots & \dots \\ 0 & 0 & \cdots & 1 & 0 & \alpha_{n-2} \\ 0 & 0 & \cdots & 0 & 1 & \alpha_{n-1} \end{pmatrix}, \quad (2)$$

a compact form of which is

$$\boldsymbol{F}_f^{(n)} = \begin{pmatrix} \theta & f \\ \boldsymbol{E} & \downarrow \end{pmatrix},$$

where θ is the zero-vector string of (n-1)-order.

Through classical Galois and Fibonacci matrices, it is possible to generate binary m-sequences of pseudorandom numbers (PRNs) similar to the sequences formed by classical LFSR generators in Galois or Fibonacci configurations. It's known that LFSRs are suitable generators of PRNs, but they have undesirable properties that reduce the efficiency of their use. For registers of length n, their internal state is a function of the n previous output bits of the generator. Even if the feedback scheme is kept secret, it can be determined from $2n$ generator output bits using the Berlecamp-Messy (BM) algorithm, [12], which reduces the crypto-resistance of the PRN generator.

The main goal of this study is to develop PRN generators based on generalized Galois and Fibonacci matrices free from the Berlecamp-Messy attack.

2 Simple Galois Matrices

In the previous section, it noted that the classical Galois (1) and Fibonacci (2) matrices are interconnected by a right-side transpose, to denote which we use the symbol $\perp$, [13], i.e.

$$\boldsymbol{G}_f^{(n)} = \begin{pmatrix} \leftarrow & f \\ \boldsymbol{E} & \mathbf{0} \end{pmatrix} \xleftrightarrow{\perp} \boldsymbol{F}_f^{(n)} = \begin{pmatrix} \theta & f \\ \boldsymbol{E} & \downarrow \end{pmatrix}. \qquad (3)$$

The peculiarity of the involutive transformation (3) is manifested in the fact that, first, the PRN generators based on the classical Galois and Fibonacci matrices form sequences of maximal length and, second, the sequences of pseudorandom numbers taken from any discharge of the PRN matrix generator satisfy all three postulates of, [14]. The second involutional transformation that preserves the properties of the matrices the same as those delivered by right-side transposition is the classical (left-side) transposition operation since there is no objective reason why this should not be the case. The matrices $\boldsymbol{G}^*$ and $\boldsymbol{F}^*$ are formed by the left-side transposition of the matrices $\boldsymbol{G}$ and $\boldsymbol{F}$. We will call them *conjugate* matrices.

$$\boldsymbol{G}(\boldsymbol{F}) \xleftrightarrow{T} \boldsymbol{G}^*(\boldsymbol{F}^*).$$

The compact forms of conjugate Galois and Fibonacci matrices have the form

$$\boldsymbol{G}^* = \begin{pmatrix} \uparrow & \boldsymbol{E} \\ f & \theta \end{pmatrix}; \quad \boldsymbol{F}^* = \begin{pmatrix} \mathbf{0} & \boldsymbol{E} \\ f & \rightarrow \end{pmatrix}.$$

Finally, one more involutional transformation, preserving the properties of the original Galois and Fibonacci matrices, is the operation of matrices reversal

$$\boldsymbol{G}_f^{(n)}(\boldsymbol{F}_f^{(n)}) \xleftrightarrow{-} \bar{\boldsymbol{G}}_f^{(n)}(\bar{\boldsymbol{F}}_f^{(n)}).$$

The set of matrices $\{\boldsymbol{G}, \boldsymbol{F}, \boldsymbol{G}^*, \boldsymbol{F}^*\}$, augmented by the corresponding inverse matrices, let us call the complete set of *simple Galois-like matrices*. The completeness of the group should be understood in the sense that except by using the left- and right-side transpose operations and the inversion operation, there are no other involutive transformations that would lead to the appearance of new matrices that are not included in the set of simple Galois-like matrices.

Indeed, by numerical examples, it is easy to verify that such involutive transformations as turning by 180° the Galois matrices for their horizontal or vertical axes of symmetry are unacceptable since the transformed matrices lose the property of primitivity, i.e., their order becomes less than the maximum order. The involutional transformation of the type of rotation of Galois matrices clockwise (counterclockwise) by an angle equal to π is also unacceptable since it turns out to be redundant since

$$\boldsymbol{G}(\boldsymbol{F}) \xleftrightarrow{\pi} \boldsymbol{F}^*(\boldsymbol{G}^*).$$

The graph of the complete set of simple Galois-like matrices is shown in Figure 3.

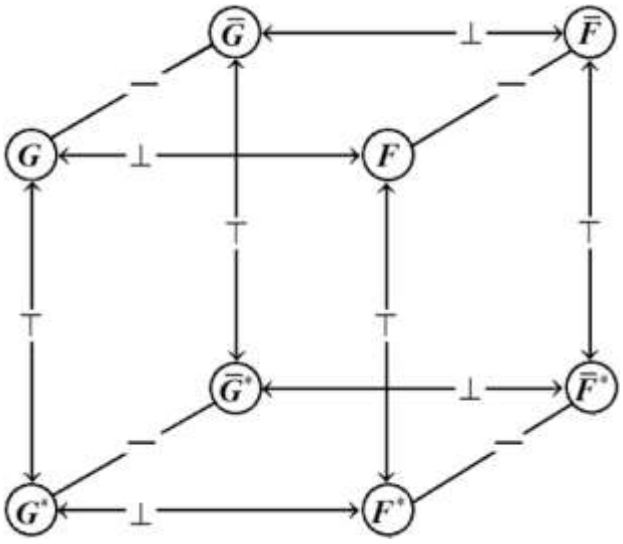

Fig. 3: Complete set involutive connected simple Galois-like matrices

For example, Table 1 gives the complete set of simple Galois-like matrices of order four generated by PrP $f_4 = 10011$.

Table 1. Complete set of simple Galois-like matrices

$\boldsymbol{G} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\boldsymbol{F} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
$\bar{\boldsymbol{G}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$	$\bar{\boldsymbol{F}} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$
$\boldsymbol{G}^* = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$	$\boldsymbol{F}^* = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$
$\bar{\boldsymbol{G}}^* = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	$\bar{\boldsymbol{F}}^* = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

As well as classical matrices $\boldsymbol{G}_f^{(n)}$, all Galois-like matrices are primitive matrices (generators) using which the maximum length PRNs form, and the sequences of PRNs selected from any discharge of the generator support all three postulates of the Golomb.

Let us turn to the matrix $\bar{\boldsymbol{G}}$ (Table 1), the inverse of the simple matrix $\boldsymbol{G}$. The compact form of matrices $\bar{\boldsymbol{G}}$ can represent in the following form

$$\bar{\boldsymbol{G}} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{E} \\ f & \leftarrow \end{pmatrix},$$

where the bottom line, i.e., the combination $f \leftarrow$, is nothing but the FE $\bar{\theta}$ of the matrix $\bar{\boldsymbol{G}}$, the inverse of the forming element $\theta = 10$ of the simple matrix $\boldsymbol{G}$. Based on the matrix $\bar{\boldsymbol{G}}$ from Table 1, we arrive at a relatively simple way to determine the FE $\bar{\theta}$ of the matrix $\bar{\boldsymbol{G}}$, generated by the PrP f_n of arbitrary degree *n*. Namely

$$\bar{\theta} = f_n \setminus \alpha_0 = 1\alpha_{n-1}\alpha_{n-2}\ldots\alpha_1. \qquad (4)$$

In fact, by multiplying the right part of the expression (4) by FE $\theta = 10$, and, reducing the product to the remainder modulo f_n, we obtain

$$\left(\theta \cdot \bar{\theta}\right) \bmod f_n = 1,$$

which confirms the correctness of the calculation of the FE $\bar{\theta}$ of the matrix $\bar{\boldsymbol{G}}$.

The general form of the matrix $\bar{\boldsymbol{G}}_f^{(n)}$, the inverse of CGM (1), is as follows

$$\bar{\boldsymbol{G}}_f^{(n)} = \begin{pmatrix} \boldsymbol{0} & 1 & 0 & \cdots & 0 & 0 & 0 \\ \boldsymbol{0} & 0 & 1 & \cdots & 0 & 0 & 0 \\ \boldsymbol{0} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & 0 & 0 & \cdots & 0 & 1 & 0 \\ \boldsymbol{0} & 0 & 0 & \cdots & 0 & 0 & 1 \\ \boldsymbol{1} & \alpha_{n-1} & \alpha_{n-2} & \cdots & \alpha_3 & \alpha_2 & \alpha_1 \end{pmatrix}. \qquad (5)$$

By right-side transpose of the matrix (5), we arrive at the inverse CFM

$$\bar{\boldsymbol{F}}_f^{(n)} = \begin{pmatrix} \alpha_1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_2 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \alpha_3 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_{n-2} & 0 & 0 & \cdots & 0 & 1 & 0 \\ \alpha_{n-1} & 0 & 0 & \cdots & 0 & 0 & 1 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}, \qquad (6)$$

the compact form of which is

$$\bar{\boldsymbol{F}}_f^{(n)} = \begin{pmatrix} \downarrow & \boldsymbol{E} \\ f & \boldsymbol{\theta} \end{pmatrix}.$$

By left-side transpose of matrices (1), (2) and (5), (6), we obtain the corresponding simple conjugate Galois-like matrices.

Let $S(k)$ be the state of the matrix Galois-like PRN generator at the *k*-step. The state of the generator at the (*k*+1)-step is defined by the recurrence relation

$$S(k+1) = S(k) \cdot \boldsymbol{M}_f^{(n)}, \quad k = 0, 1, \ldots, \qquad \ldots, S(0) = \underbrace{00\ldots01}_{n \text{ bit}}, \qquad (7)$$

where $\boldsymbol{M}_f^{(n)}$ is one of the primitive Galois-like matrices of *n*-order generated by the PrP f_n.

Considering that simple Galois-like matrices contain unit matrices of (*n*-1)-order, we can significantly reduce the computer time required to estimate the oscillator's state at the following (*k*+1)-th computation step.

Indeed, let us represent the state of $S(k)$ generators of PRN in the form of

$$S(k) \to S_k = (s_{n-1}, s_{n-2}, \ldots, s_1, s_0),$$

where s_i is the *i*-bit of the generator. Thus, equality (7) can be rewritten in the following form

$$S_{k+1} = (s_{n-1}, s_{n-2}, \ldots, s_1, s_0) \cdot \boldsymbol{M}_f^{(n)}. \quad (8)$$

For the Galois generator PRN in (8) instead of $\boldsymbol{M}_f^{(n)}$, we should substitute the matrix $\boldsymbol{G}_f^{(n)}$, given by expression (1). We have

$$S_{k+1}^{(G)} = \big((s_{n-1} \cdot \alpha_{n-1} \oplus s_{n-2}), (s_{n-1} \cdot \alpha_{n-2} \oplus s_{n-3}), \ldots, \ldots, (s_{n-1} \cdot \alpha_1 \oplus s_0, s_{n-1})\big)$$

where the upper index *G* means that the binary vector S_{k+1} is produced by the Galois generator.

For the Fibonacci generator, replacing in (8) $\boldsymbol{M}_f^{(n)}$ by the matrix (2), we obtain

$$S_{k+1}^{(F)} = \big(s_{n-2}, s_{n-3}, \ldots, s_1, s_0, s_{n-1} \oplus \ldots \ldots \oplus (s_{n-2} \cdot \alpha_1) \oplus (s_{n-3} \cdot \alpha_2) \oplus \ldots \oplus (s_0 \cdot \alpha_{n-1})\big).$$

Similarly, we arrive at expressions for the binary vectors formed by the generators generated by the remaining simple Galois-like matrices, namely

$$S_{k+1}^{(\bar{G})} = \big(s_0, (s_{n-1} \oplus \alpha_{n-1} \cdot s_0), (s_{n-2} \oplus \alpha_{n-2} \cdot s_0), \ldots, \ldots, (s_1 \oplus \alpha_1 \cdot s_0)\big);$$

$$S_{k+1}^{(\bar{F})} = \big((s_{n-1} \cdot \alpha_1) \oplus (s_{n-2} \cdot \alpha_2) \oplus \ldots \ldots \oplus (s_1 \cdot \alpha_{n-1}) \oplus s_0, s_{n-1}, s_{n-2}, \ldots, s_1\big);$$

$$S_{k+1}^{(G*)} = \big(\bigoplus_{i=0}^{n-1} \alpha_i \cdot s_i, s_{n-1}, s_{n-2}, \ldots, s_1\big);$$

$$S_{k+1}^{(F*)} = \big(s_0, s_{n-1} \oplus \alpha_1 \cdot s_0, s_{n-2} \oplus \alpha_2 \cdot s_0, \ldots, \ldots, s_1 \oplus \alpha_{n-1} \cdot s_0\big);$$

$$S_{k+1}^{(\bar{G}*)} = \big(s_{n-2}, s_{n-3}, \ldots, s_0, \bigoplus_{i=1}^{n} \alpha_i \cdot s_{i-1}\big);$$

$$S_{k+1}^{(\bar{F}*)} = \big(\alpha_1 \cdot s_{n-1} \oplus s_{n-2}, \alpha_2 \cdot s_{n-1} \oplus s_{n-3}, \ldots, \ldots, \alpha_{n-1} \cdot s_{n-1} \oplus s_0, s_{n-1}\big).$$

The computational complexity of the algorithms for generating PRNs based on the vectors $S_{k+1}^{(M)}$ is proportional to the order *n* of the $\boldsymbol{M}_f^{(n)}$ matrices. In contrast, the computational complexity of PRN generation by formula (7) is quadratically dependent on the order of these matrices.

3 Generalized Galois Matrices

Let us notice such features of the matrices $\bar{\boldsymbol{G}}_f^{(n)}$ inverse classical CGMs $\boldsymbol{G}_f^{(n)}$. First, the FE ω of the matrix $\bar{\boldsymbol{G}}_f^{(n)}$ must exceed the value θ of the forming element of the matrix, remaining a primitive element of the field generated by the PrP f_n. Secondly, the algorithm for forming the matrix $\bar{\boldsymbol{G}}_f^{(n)}$ remains similar to the algorithm for constructing the matrix $\boldsymbol{G}_f^{(n)}$. Third, the solution $\bar{f}_n$, produced by the BM algorithm based on the set of bits generated by the matrix $\bar{\boldsymbol{G}}_f^{(n)}$ generator, is inverse to the polynomial f_n. The matrices $\bar{\boldsymbol{G}}_f^{(n)}$ contain features that we will transfer to the *generalized Galois matrix* (GGM) notion, giving the term as follows

Definition. We will refer to generalized Galois matrices (GGM) $\boldsymbol{G}_{f,\omega}^{(n)}$ as square matrices of order *n* generated by irreducible over F_2 polynomials f_n and forming elements $\omega > 10$ belonging to the field $GF(2^n)$, and both f_n and ω need not be primitive.

GGM synthesis algorithm. The selected element ω of the field $GF(2^n)$, generated by the irreducible polynomial (IP) f_n, is placed in the lower right corner of the formed matrix $\boldsymbol{G}_{f,\omega}^{(n)}$. Element ω acts as a forming element of the matrix $\boldsymbol{G}_{f,\omega}^{(n)}$. All row bits to the left ω fill with zeros. Each subsequent matrix row in the bottom-up direction forms by shifting one position to the left of the previous row. Zero writes to the cell that freed after the line shift. If the row's highest nonzero digit exceeds the matrix's left border at a particular shift step, this row is reduced to the remainder modulo f_n, returning it to the limits of the formed matrix. Then the process continues according to the described scheme until all *n* rows of the synthesized matrix fill.

Following the above algorithm, let's compose a GGM, choosing, for example, the parameters of the synthesized matrix $\boldsymbol{G}_{f,\omega}^{(n)}$ as follows: $n = 8$; $f_8 = 100011011$; $\omega = 01011011$. We obtain

$$\boldsymbol{G}_{f,\omega}^{(8)} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

We come to generalized Fibonacci matrices $\boldsymbol{F}_{f,\omega}^{(n)}$ due to the right-side transposition of matrices

$\boldsymbol{G}_{f,\omega}^{(n)}$. Note that the matrices $\boldsymbol{G}_{f,\omega}^{(n)}$ and $\boldsymbol{F}_{f,\omega}^{(n)}$ do not allow their compact representation in the form used for the classical matrices $\boldsymbol{G}_{f}^{(n)}$ and $\boldsymbol{F}_{f}^{(n)}$.

From the theory of polynomials of one variable *x*, we know that multiplying an arbitrary polynomial $\omega_k(x)$ of *k*-degree is equivalent to shifting it one digit to the left. Or in other words,

$$x \cdot \omega_k(x) \rightarrow \omega_{k+1}(x). \quad (9)$$

Using relation (9) and taking into account the way of GGM formation, we write a chain of transformations

$$\boldsymbol{G}_{f,\omega}^{(n)} \Rightarrow \begin{bmatrix} x^{n-1}\cdot\omega \\ x^{n-2}\cdot\omega \\ \cdots \\ x^{1}\cdot\omega \\ x^{0}\cdot\omega \end{bmatrix} \bmod f_n = \omega \cdot \begin{bmatrix} x^{n-1} \\ x^{n-2} \\ \cdots \\ x \\ 1 \end{bmatrix} \bmod f_n. \quad (10)$$

The elements of the right vector-column inequality (10) are monomials which, being represented in binary form, turn the column into a unit matrix $\boldsymbol{E}$ of *n*-order, i.e.

$$\begin{bmatrix} x^{n-1} \\ x^{n-2} \\ \cdots \\ x \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \boldsymbol{E}, \quad (11)$$

which allows us to postulate such provisions.

Statement 1. The generalized Galois matrix $\boldsymbol{G}_{f,\omega}^{(n)}$, generated by IP f_n, is isomorphic to its forming element ω, which is a field element $GF(2^n)$, i.e.

$$\boldsymbol{G}_{f,\omega}^{(n)} \Leftrightarrow \omega, \quad (12)$$

where $\Leftrightarrow$ is the sign of isomorphism.

Thus, according to expressions (9)-(11), there is a one-to-one correspondence between the GGM $\boldsymbol{G}_{f,\omega}^{(n)}$ and its FE ω, which is represented by relation (12), leads to the results below in the form of consequences:

Consequence 1. A generalized Galois matrix $\boldsymbol{G}_{f,\omega}^{(n)}$ is primitive if its forming element ω is a primitive element of the field $GF(2^n)$, generated by an irreducible (not necessarily primitive) polynomial f_n.

Consequence 2. To raise the generalized Galois matrix to degree *k* is sufficient to calculate $\omega_k = \omega^k (\bmod f_n)$, which is just the generating element of the *k*-degree of the matrix $\boldsymbol{G}_{f,\omega}^{(n)}$.

Statement 2. A PRN generator based on a Galois matrix $\boldsymbol{G}_{f,\omega}^{(n)}$ such that f_n is not primitive and $\omega > 10$ is a primitive element of the field $GF(2^n)$ generated by the polynomial of f_n found to be free from the BM attack.

Let us prove Statement 2 with simple numerical examples. Let the nonprimitive IP of the eighth-degree $f_8 = 100011011$ and the FE $\omega = 11$, a primitive element of the field generated by the polynomial f_8, be chosen. Let us define the first 16 eight-bit elements of the multiplicative group caused by *k*-degrees of FE ω modulo f_8, which we place in Table 2. The sequence of the multiplicative group elements repeats the PRN of binary vectors calculated by the formula (9) if the parameters f and ω of the matrix $\boldsymbol{G}_{f,\omega}^{(8)}$ coincide with the corresponding parameters of the example under consideration.

A set of bits of any column in Table 2, fed to the input of the BM algorithm, leads to the output of the PrP $f' = 100011101$. If, for example, while keeping the generating polynomial f_8, we choose $\omega = 110$ as a primitive FE, then the solution of the BM algorithm is PrP $f'' = 100101011$. Both f' and f'' are different from f_8. Thus, we have confirmed that the generalized matrix generators of PRNs are free from the Berlecamp-Messy attack.

Table 2. Fragment of the multiplicative group

k	# of binary vector discharges							
	8	7	6	5	4	3	2	1
1	0	0	0	0	0	0	1	1
2	0	0	0	0	0	1	0	1
3	0	0	0	0	1	1	1	1
4	0	0	0	1	0	0	0	1
5	0	0	1	1	0	0	1	1
6	0	1	0	1	0	1	0	1
7	1	1	1	1	1	1	1	1
8	0	0	0	1	1	0	1	0
9	0	0	1	0	1	1	1	0
10	0	1	1	1	0	0	1	0
11	1	0	0	1	0	1	1	0
12	1	0	1	0	0	0	0	1
13	1	1	1	1	1	0	0	0

14	0	0	0	1	0	0	1	1
15	0	0	1	1	0	1	0	1
16	0	1	0	1	1	1	1	1

The noted feature of the generalized matrix generators of PRN appears for the following reasons. The BM algorithm successfully copes with defining only one unknown parameter — PrP f_n, generating matrix generators. In generalized PRN generators, in addition to the primitive FE θ, the unknown is also the irreducible polynomial f_n, which, together with θ, generates the matrix $\boldsymbol{G}_{f,\omega}^{(n)}$. However, the BM algorithm is not designed to calculate the two unknown parameters and, therefore, becomes invalid when organizing an attack on the generalized PRN generators. That is first. Secondly, in any case (whether the conditions of applicability of the BM algorithm are satisfied or not), the processor implementing the BM algorithm always gives as a solution that or the value of PrP of *n*-degree. At the same time, it can build the generalized matrix generators of PRN based on IPs, not necessarily primitive.

Can easily extend the results obtained to the space of objects (IPs and GGMs) over a simple Galois field of arbitrary odd characteristics *p*. For illustration, let us give the generalized matrix $\boldsymbol{G}_{f,\omega}^{(n)}$ of the fourth order over the field F_5, generated by the IP $f_4 = 13201$ and the primitive FE $\omega = 3402$.

$$\boldsymbol{G}_{f,\omega}^{(4)} = \begin{pmatrix} 0 & 3 & 0 & 1 \\ 4 & 2 & 2 & 0 \\ 0 & 4 & 2 & 2 \\ 3 & 4 & 0 & 2 \end{pmatrix}.$$

The matrix $\boldsymbol{G}_{f,\omega}^{(4)}$ is primitive, and the period of the multiplicative group that compiles from it is 624.

4 Key Scientific Findings and Future

The study results hold significant importance from both scientific and practical standpoints due to the development of algorithms for constructing crypto-resistant matrix generators of pseudorandom numbers. These generators are based on generalized Galois matrices and offer reliable protection against Berlekamp-Massey attacks. What factors contribute to the enhanced cryptographic strength of the proposed pseudorandom number generators when compared to PRN generators using classical Galois matrices? Two key factors should be noted.

Firstly, highly sparse matrices, which CGMs fall under, may exhibit specific structural patterns that compromise the randomness of the PRNG sequence, rendering it more susceptible to predictive attacks. Secondly, the pronounced sparsity of CGMs simplifies the application of Berlekamp-Massey attacks, which aim to recover the linear feedback structure within the generator.

Now, let's highlight the distinctive features of generalized Galois matrices and the PRN generators based on them. Firstly, GGMs contain a higher density of random elements than CGMs, resulting in the increased cryptographic resilience of the PRN generators. Secondly, effective algorithms for breaking GGM-based PRN generators, which maintain polynomial complexity in calculations, have not yet been developed. Any attempt to launch a frontal attack on the generator is practically unfeasible due to the challenge posed by the "nightmare of large numbers," significantly when the order of GGM exceeds 30.

5 Conclusion

The main results of this work are:

1. The variants of construction of binary generators of PRN based on the so-called generalized Galois and Fibonacci matrices, by which the identical binary sequences as the sequences formed by the corresponding register generators can generate programmatically, have been developed. The transition from classical to generalized Galois and Fibonacci matrices, accompanied by the expansion of the manifold of matrix generators of PRN, is provided in two ways. Firstly, the expansion of the manifold is achieved by increasing the number of primitive elements forming generalized matrices since the classical PRN generators use only the element equal to 10 in the matrices. Secondly, if the matrices of classical PRN matrix generators are constructed based on primitive polynomials, the IPs, which are not necessarily primitive, can be used in the matrices of generalized generators.

2. It is shown that the generalized PRN matrix generators are free from the Berlecamp-Messy attack. The noted property is a consequence of this peculiarity of the Berlecamp-Messy algorithm. When classical PRN matrix generators cracked using the Berlecamp-Messy algorithm, the problem of computing the only unknown, the primitive polynomial generating the generating matrix is solved. In generalized PRN matrix generators, there is a need to determine two unknown parameters: both the irreducible polynomial and the generating

element that jointly generate the generalized matrices, i.e., a problem arises that is intractable for the Berlecamp-Messy algorithm by definition.

3. One of the most promising directions of applying generalized Galois and Fibonacci matrices is cryptographic applications, particularly the construction of crypto-stable systems of stream information encryption.

References:

[1] Blahut R. E. *Theory and Practice of Error Control Codes.* - Addison-Wesley Publishing Company Reading, (1984). pp.500, ISBN: 0201101025

[2] Berlekamp E. R. *Algebraic Coding Theory*, New York: McGraw-Hill, 1968. Revised ed., Aegean Park Press, (1984). ISBN: 0-89412-063-8

[3] Peterson, W.W., Weldon, E.J. *Error Correcting Codes*, MIT Press, Cambridge, MA (1972). pp.560, ISBN: 10: 0262527316 ISBN: 13: 9780262527316

[4] Shu Lin, Daniel Costello Jr. *Error Control Coding*. Fundamentals and Applications. Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1983). pp.604, ISBN: 0-13-283796-X

[5] Schneier, B. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York (1996). pp.1027, ISBN: 13: 978-0471117094

[6] Smart N. *Cryptography: An Introduction*, 3rd ed. McGraw-Hill College, (2013). pp.436, ISBN: 13: 978-0077099879

[7] Simon Edwards. *Modern Cryptography*. (1996). pp.170, ISBN: 13: 979-8622477546

[8] Stream Ciphers. *The results of the open foreign cryptology*. - (1997). http//www/ssl/stu/neva/ ru/psw/crypto/potok/str_ciph.htm (Accessed Date: 4/8/2023)

[9] Beletsky A. *Generalized Galois-Fibonacci Matrix Generators Pseudorandom Sequences*. I. J. Computer Network and Information Security, 2021, 6, pp.57-69. DOI: 10.5815/ijcnis.2021.06.05

[10] Beletsky A. *Generalized Galois and Fibonacci Matrices in Cryptographic Applications*. WSEAS Transactions on Circuits and Systems, Vol. 21, 2022, Art. #1, pp.1-19. DOI: 10.37394/23201.2022.21.1

[11] Lidl R., Niederreiter H. *Finite Fields*. Cambridge University Press, (1996). pp.755, ISBN: 9780511525926

[12] Erin Casey. *Berlekamp-Massey Algorithm*. University of Minnesota, (2000). pp.10.

[13] Mullajonov R.V. *Generalized Transposition of Matrices and Structures of Linear Large-Scale Systems*. Reports of the National Academy of Sciences of Ukraine, (2009), No. 10. pp.27-35.

[14] Golomb S.W. *Shift Register Sequences*, Holden-Day, San Francisco CA, (1967).

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

The author contributed to the present research at all stages, from the problem formulation 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 authors have no conflict of interest to declare.