ORIGINAL PAPER

A MIXED CODING/ DECODING ALGORITHM BASED ON COMPLEX RECURRENCE RELATIONS

MUSTAFA ASCI[1], TALAT AYTURK ADALIOGLU[1]

Manuscript received: 07.01.2025; Accepted paper: 23.11.2026;
Published online: 30.03.2026.

Abstract. *Nowadays, as information security is gaining vital importance, we aim to contribute to the development of coding/ decoding algorithms. To this end, we will apply innovative coding/decoding techniques using 2x2 matrices for Gauss-Fibonacci and Gauss-Lucas numbers, and 3x3 matrices for Leonardo and Gauss-Leonardo numbers. We will also strengthen our system by integrating the mixed model to maximise information security. These new algorithms will improve information security while providing high accuracy.*

Keywords: *coding/decoding algorithm; mixed model; cryptography; Fibonacci numbers; Lucas numbers; Leonardo numbers; Gauss Fibonacci numbers; Gauss-Lucas numbers; Gauss-Leonardo numbers; Fibonacci Q- Matrix; Leonardo Q- Matrix.*

Mathematics Subject Classification: *11B37, 11H71, 11T71, 68P30.*

1. INTRODUCTION AND PRELIMINARIES

Fibonacci and Lucas numbers are defined by the following recurrence relations (see [1-2] for more details and applications):

$$F_{n+1} = F_n + F_{n-1}, n \geq 1$$

with the initial conditions $F_0 = 0, F_1 = 1$ and

$$L_{n+1} = L_n + L_{n-1}, n \geq 1$$

with the initial conditions $L_0 = 2, L_1 = 1$, respectively. The Fibonacci Q −matrix is defined as:

$$Q = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

and the n −the power of the Fibonacci Q −matrix is given by:

$$Q^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}.$$

Gauss Fibonacci numbers GF_n are defined recursively in [3] as:

$$GF_n = GF_{n-1} + GF_{n-2}, n \geq 2$$

[1]Pamukkale University, Science and Arts Faculty, Department of Mathematics, 20160 Denizli, Turkey. E-mail: mustafa.asci@yahoo.com; talat.ayturk@gmail.com.

https://doi.org/10.46939/J.Sci.Arts-26.1-a01

with the initial values $GF_0 = i, GF_1 = 1$. Also, complex Fibonacci numbers are defined in [4]. It is clear that these numbers are closely related to Fibonacci numbers:

$$GF_n = F_n + iF_{n-1},$$

where $i = \sqrt{-1}$. Next, we give a matrix representation of the Gauss Fibonacci numbers [5-6]. Let R −matrix is defined as follows:

$$R = \begin{bmatrix} 1 & i \\ i & 1-i \end{bmatrix}, Q = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

and let R_n −matrix be:

$$\begin{aligned} R_n &= RQ^n \\ &= \begin{bmatrix} 1 & i \\ i & 1-i \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \\ &= \begin{bmatrix} GF_{n+1} & GF_n \\ GF_n & GF_{n-1} \end{bmatrix}. \end{aligned}$$

Similarly, Gauss-Lucas numbers GL_n are defined recursively as:

$$GL_n = GL_{n-1} + GL_{n-2}, n \geq 2$$

with the initial conditions $GL_0 = 2 - i, GL_1 = 1 + 2i$. Subsequently, we provide a matrix representation of the Gauss-Lucas numbers [7].

S matrix is defined as follows:

$$S = \begin{bmatrix} 1+2i & 2-i \\ 2-i & -1+3i \end{bmatrix}, Q = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

and S_n −matrix is given by:

$$\begin{aligned} S_n &= SQ^n \\ &= \begin{bmatrix} 1+2i & 2-i \\ 2-i & -1+3i \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \\ &= \begin{bmatrix} GL_{n+1} & GL_n \\ GL_n & GL_{n-1} \end{bmatrix}. \end{aligned}$$

Determinants of the Gauss Fibonacci matrix R_n and the Gauss-Lucas matrix S_n are given by:

$$\det(R_n) = GF_{n+1}GF_{n-1} - GF_n^2 = (-1)^n(2-i)$$

$$\det(S_n) = GL_{n+1}GL_{n-1} - GL_n^2 = 5(-1)^{n+1}(2-i).$$

Leonardo numbers were introduced and studied by Catarino and Burges in [8-9]. These numbers are defined by the second-order inhomogeneous recurrence relation:

$$Le_n = Le_{n-1} + Le_{n-2} + 1, n \geq 2$$

with the initial conditions $Le_0 = Le_1 = 1$. Also, these numbers can be defined as:

$$Le_{n+1} = 2Le_n - Le_{n-2}, n \geq 2.$$

Gauss-Leonardo numbers GLe_n are defined in [9]:

$$GLe_n = GLe_{n-1} + GLe_{n-2} + (1+i), n \geq 2$$

with initial conditions $GLe_0 = 1 - i, GLe_1 = 1 + i$.

The Gauss-Leonardo numbers are closely related to the Leonardo numbers:

$$GLe_n = Le_n + iLe_{n-1},$$

where Le_n denotes the $n-$the Leonardo numbers and GLe_n denotes the $n-$th Gauss-Leonardo numbers [10].

Let us define the Leonardo $Q-$matrix as:

$$Q = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

and the Leonardo $T-$matrix defined as follows:

$$T = \begin{bmatrix} 5 & 3 & 1 \\ 3 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} Le_3 & Le_2 & Le_1 \\ Le_2 & Le_1 & Le_0 \\ Le_1 & Le_0 & Le_{-1} \end{bmatrix}$$

then T_n-matrix:

$$T_n = TQ^n = \begin{bmatrix} 5 & 3 & 1 \\ 3 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}^n = \begin{bmatrix} Le_{n+3} & Le_{n+2} & Le_{n+1} \\ Le_{n+2} & Le_{n+1} & Le_n \\ Le_{n+1} & Le_n & Le_{n-1} \end{bmatrix}.$$

The Gauss-Leonardo $K-$matrix is given by:

$$K = \begin{bmatrix} 5+3i & 3+i & 1+i \\ 3+i & 1+i & 1-i \\ 1+i & 1-i & -1+i \end{bmatrix} = \begin{bmatrix} GLe_3 & GLe_2 & GLe_1 \\ GLe_2 & GLe_1 & GLe_0 \\ GLe_1 & GLe_0 & GLe_{-1} \end{bmatrix}$$

and K_n-matrix is defined as:

$$K_n = KQ^n = \begin{bmatrix} 5+3i & 3+i & 1+i \\ 3+i & 1+i & 1-i \\ 1+i & 1-i & -1+i \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}^n = \begin{bmatrix} GLe_{n+3} & GLe_{n+2} & GLe_{n+1} \\ GLe_{n+2} & GLe_{n+1} & GLe_n \\ GLe_{n+1} & GLe_n & GLe_{n-1} \end{bmatrix}.$$

The determinants of the Leonardo matrices T_n and the Gauss Leonardo matrices K_n are given by:

$$\det(T_n) = {Le_n}^2 - Le_{n-1}Le_{n+1} = Le_{n-1} - Le_{n-2} + 4(-1)^n.$$

$$\det(K_n) = GLe_{n+1}GLe_{n-1} - (GLe_n)^2$$

$$= (-1)^{n+1}(8-4i) + 2(1+i)(2GF_{n+1} - GF_{n+2} - GF_n.$$

Many researchers have developed coding/decoding algorithms based on various number sequences and contributed to the literature [11-25]. Tas et al. introduced a coding/decoding algorithm using Fibonacci Q −matrix [11] that partitions the message matrix into 2×2 block matrices. Building on this idea, Ucar introduced the `Minesweeper Model' [12], which integrates multiple number sequences to establish a more robust coding system. This innovative approach offers a novel perspective on coding algorithms and their applications. Motivated by these studies, we propose a mixed coding/ decoding algorithm that extends Fibonacci, Lucas, and Leonardo numbers into the complex plane. Unlike previous approaches [11-12], the proposed method allows the parameter n to be chosen independently of the number of block matrices. This flexibility significantly enhances the reliability and effectiveness of the coding/ decoding algorithms.

In this study, we developed coding/decoding algorithms based on four different number sequences using a mixed model. The paper is organized into two main sections. In the first section, we present an algorithm that partitions the message matrix into 2×2 block matrices using R_n and S_n −matrix derived from Fibonacci Q −matrix. The second section introduces an algorithm that divides the message matrix into 3×3 block matrices using T_n and K_n −matrix derived from Leonardo Q −matrix. The encoding process uses the printable ASCII character set defined in decimal. This approach provides a standardized structure for the algorithms, ensuring operational consistency. Additionally, the Minesweeper model enhances information security while providing robust protection against data manipulation. Furthermore, the proposed method ensures reliable and highly accurate data transmission over communication channels.

2. MIXED CODING/ DECODING ALGORITHM

2.1. A MIXED CODING/ DECODING METHOD USING R −*MATRIX AND* S −*MATRIX*

In this section, we introduce a mixed coding/ decoding algorithm using Gauss Fibonacci and Gauss Lucas numbers. We embed the message into a square matrix M of size $2m \times 2m$ and partition it into 2×2 block matrices, denoted by $B_k (1 \leq k \leq m^2)$, proceeding from left to right.

We define the symbols and matrices utilized in the proposed coding method as follows. The matrices B_k, E_k, Q_n, R_n and S_n are given by:

$$B_k = \begin{bmatrix} b_1^k & b_2^k \\ b_3^k & b_4^k \end{bmatrix}, E_k = \begin{bmatrix} e_1^k & e_2^k \\ e_3^k & e_4^k \end{bmatrix}$$
$$Q_n = \begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \end{bmatrix}, R_n = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$$
$$S_n = \begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix}$$

In this context, n represents a positive integer, and the characters correspond to the printable characters defined in the decimal representation of ASCII.

MIXED MODEL

Coding Algorithm

Step 1. Divide the message matrix M into blocks $B_k(1 \leq k \leq m^2)$.

Step 2. Choose n.

Step 3. Determine the elements $b_k^j(1 \leq j \leq 4)$ for each block.

Step 4. Compute $\det(B_k) \rightarrow d_k$.

Step 5. Construct encoded matrix $F = [d_k, b_k^n]_{n \in 1,2,3}$.

Step 6. End of algorithm.

Decoding Algorithm

Step 1. Compute the matrix R_n.

Step 2. Compute the matrix S_n.

Step 3. Compute $r_1 b_1^k + r_3 b_2^k \rightarrow e_1^k$ for $k = 2l + 1, 0 \leq l \leq m$.

Step 4. Compute $s_1 b_1^k + s_3 b_2^k \rightarrow e_1^k$ for $k = 2l, 1 \leq l \leq 2m$.

Step 5. Compute $r_2 b_1^k + r_4 b_2^k \rightarrow e_2^k$ for $k = 2l + 1, 0 \leq l \leq m$.

Step 6. Compute $s_2 b_1^k + s_4 b_2^k \rightarrow e_2^k$ for $k = 2l, 1 \leq l \leq 2m$.

Step 7. Solve $(-1)^n(2 - i)d_k = e_1^k\left(q_2 b_3^k + q_4 x_k\right) - e_2^k(q_1 b_3^k + q_3 x_k)$ for $k = 2l + 1, 0 \leq l \leq m$.

Step 8. Solve $5(-1)^{n+1}(2 - i)d_k = e_1^k\left(r_2 b_3^k + r_4 x_k\right) - e_2^k(r_1 b_3^k + r_3 x_k)$ for $k = 2l, 1 \leq l \leq m$.

Step 9. Substitute $x_k = b_4^k$.

Step 10. Construct B_k.

Step 11. Reconstruct the original message matrix M.

Step 12. End of algorithm.

Example: Let us consider the message matrix for the following message text:

“CODING+DECODING#“

Using the message text, we get the following message matrix M:

$$M = \begin{bmatrix} C & O & D & I \\ N & G & + & D \\ E & C & O & D \\ I & N & G & \# \end{bmatrix}$$

Coding Algorithm

Step 1. We partition the 4×4 message matrix M into 2×2 block matrices, denoted by $B_k(1 \leq k \leq 4)$, proceeding from left to right:

$$B_1 = \begin{bmatrix} C & O \\ N & G \end{bmatrix}, B_2 = \begin{bmatrix} D & I \\ + & D \end{bmatrix}$$
$$B_3 = \begin{bmatrix} E & C \\ I & N \end{bmatrix}, B_4 = \begin{bmatrix} O & D \\ G & \# \end{bmatrix}$$

Step 2. For $n = 3$ we utilize the decimal values of the printable ASCII characters:

C	**O**	**D**	**I**	**N**	**G**	**+**	**D**
67	79	69	73	78	71	43	68
E	C	O	D	I	N	G	#
69	67	79	68	73	78	71	35

Step 3. The elements of the block matrix $B_k (1 \leq k \leq 4)$ are given by:

$b_1^1 = 67$	$b_2^1 = 79$	$b_3^1 = 78$	$b_4^1 = 71$
$b_1^2 = 68$	$b_2^2 = 73$	$b_3^2 = 43$	$b_4^2 = 68$
$b_1^3 = 69$	$b_2^3 = 67$	$b_3^3 = 73$	$b_4^3 = 78$
$b_1^4 = 79$	$b_2^4 = 68$	$b_3^4 = 71$	$b_4^4 = 35$

Step 4. Next, we compute the determinants d_k of each block matrix B_k:

$$d_1 = \det(B_1) = -1405$$
$$d_2 = \det(B_2) = 1485$$
$$d_3 = \det(B_3) = 491$$
$$d_4 = \det(B_4) = -2063$$

Step 5. Using *Steps 3* and *4*, we construct the encoded matrix *F:*

$$\mathrm{F} = \begin{bmatrix} -1405 & 67 & 79 & 71 \\ 1485 & 68 & 73 & 68 \\ 491 & 69 & 67 & 78 \\ -2063 & 79 & 68 & 35 \end{bmatrix}$$

Step 6. End of algorithm.

Decoding Algorithm

Step 1. First, we compute the matrix R_3:

$$R_3 = \begin{bmatrix} 3+2i & 2+i \\ 2+i & 1+i \end{bmatrix}.$$

Step 2. Then, we compute the matrix S_3:

$$S_3 = \begin{bmatrix} 7+4i & 4+3i \\ 4+3i & 3+i \end{bmatrix}.$$

Step 3. If k is an odd number, we use the Gauss Fibonacci R −matrix. We compute the element e_1^k, for $k = 1, 3$ in order to construct the matrix E_k:

$$e_1^1 = 359 + 213i$$
$$e_1^3 = 341 + 205i$$

Step 4. If k is an even number, we use the Gauss-Lucas S −matrix. We compute the element e_1^k, for $k = 2, 4$ in order to construct the matrix E_k:

$$e_1^2 = 768 + 491i$$
$$e_1^4 = 825 + 520i$$

Step 5. If k is an odd number, we use the Gauss Fibonacci R −matrix. We compute the element e_2^k, for $k = 1, 3$ in order to construct the matrix E_k:

$$e_2^1 = 213 + 146i$$
$$e_2^3 = 205 + 136i$$

Step 6. If k is an even number, we use the Gauss-Lucas S −matrix. We compute the element e_2^k, for $k = 2, 4$ in order to construct the matrix E_k:

$$e_2^2 = 491 + 277i$$
$$e_2^4 = 520 + 305i$$

Step 7. If k is an odd number, we use the Gauss Fibonacci R −matrix. We compute the element x_k, for $k = 1, 3$ to construct the matrix E_k:

$$(2-i)(-1)^3(1405) =$$
$$(359+213i)[(2+i)78+(1+i)x_1] - (213+146i)[(3+2i)78+(2+i)x_1$$
$$\Rightarrow x_1 = 71$$
$$(2-i)(-1)^3(491) =$$
$$(341+205i)[(2+i)73+(1+i)x_3] - (205+136i)[(3+2i)73+(2+i)x_3]$$
$$\Rightarrow x_3 = 78$$

Step 8. If k is an even number, we use the Gauss-Lucas S −matrix. Now we compute the element x_k, for $k = 2, 4$ to construct the matrix E_k:

$$5(2-i)(-1)^4(1485) =$$
$$(768+491i)[(4+3i)43+(3+i)x_2 - (491)+277i)[(7+4i)43+(4+3i)x_2]$$
$$\Rightarrow x_2 = 68$$
$$5(2-i)(-1)^4(-2063) =$$
$$(825+520i)[(4+3i)71+(3+i)x_4] - (520+305i)[(7+4i)71+(4+3i)x_4$$
$$\Rightarrow x_4 = 35$$

Step 9. We substitute the recovered values of x_k as follows:

$$x_1 = b_4^1 = 71$$
$$x_2 = b_4^2 = 68$$
$$x_3 = b_4^3 = 78$$
$$x_4 = b_4^4 = 35$$

Step 10. We reconstruct the block matrices B_k:

$$B_1 = \begin{bmatrix} 67 & 79 \\ 78 & 71 \end{bmatrix}, B_2 = \begin{bmatrix} 68 & 73 \\ 43 & 68 \end{bmatrix}$$
$$B_3 = \begin{bmatrix} 69 & 67 \\ 73 & 78 \end{bmatrix}, B_4 = \begin{bmatrix} 79 & 68 \\ 71 & 35 \end{bmatrix}$$

Step 11. Finally, we obtain the original message matrix M:

$$M = \begin{bmatrix} 67 & 79 & 68 & 73 \\ 78 & 71 & 43 & 68 \\ 69 & 67 & 79 & 68 \\ 73 & 78 & 71 & 35 \end{bmatrix}$$
$$= \begin{bmatrix} C & O & D & I \\ N & G & + & D \\ E & C & O & D \\ I & N & G & \# \end{bmatrix}$$

Step 12. End of algorithm.

2.2. A MIXED CODING/DECODING METHOD USING T −*MATRIX AND* K −*MATRIX*

In this section, we introduce a coding/ decoding algorithm using Leonardo and Gauss-Leonardo numbers. We embed our message into a square matrix M of size $3m \times 3m$. Then, we divide the message matrix M into block matrices, denoted by B_k for $(1 \leq k \leq m^2)$, of size 3×3, proceeding from left to right.

We define the symbols and matrices utilized in the proposed coding methods as follows. The matrices B_k, E_k, T_n and K_n are given by:

$$B_k = \begin{bmatrix} b_1^k & b_2^k & b_3^k \\ b_4^k & b_5^k & b_6^k \\ b_7^k & b_8^k & b_9^k \end{bmatrix}, E_k = \begin{bmatrix} e_1^k & e_2^k & e_3^k \\ e_4^k & e_5^k & e_6^k \\ e_7^k & e_8^k & e_9^k \end{bmatrix}$$
$$T_n = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{bmatrix}, K_n = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{bmatrix}$$

In this context, n represents any positive integer, and the characters used correspond to the printable characters defined in the decimal representation of ASCII.

MIXED MODEL

Coding Algorithm

Step 1. Divide the message matrix M into blocks matrices $\mathrm{B_k} (1 \leq \mathrm{k} \leq \mathrm{m}^2)$.
Step 2. Choose n.
Step 3. Numerical representations $\mathrm{B_k}$ block matrices.
Step 4. Compute $\det(\mathrm{B_k}) \longrightarrow \mathrm{d_k}$.
Step 5. Construct the coded matrix $\mathrm{C} = \left[\mathrm{d_k}, \mathrm{b_n^k}\right]_{\mathrm{n} \in 1,2,3,4,6,7,8,9}$.
Step 6. End of algorithm.

Decoding Algorithm

Step 1. Compute the matrix $\mathrm{T_n}$.
Step 2. Compute the matrix $\mathrm{K_n}$.
Step 3. Compute $\mathrm{E_k}$ for $\mathrm{k} = 2\mathrm{l} + 1, 0 \leq \mathrm{l} \leq \mathrm{m}$.

$$\begin{aligned}
e_1^k &= t_1 b_1^k + t_2 b_4^k + t_3 b_7^k \\
e_2^k &= t_1 b_2^k + t_3 b_8^k \\
e_3^k &= t_1 b_3^k + t_2 b_6^k + t_3 b_9^k \\
e_4^k &= t_4 b_1^k + t_5 b_4^k + t_6 b_7^k \\
e_5^k &= t_4 b_2^k + t_6 b_7^k \\
e_6^k &= t_4 b_3^k + t_5 b_6^k + t_6 b_9^k \\
e_7^k &= t_7 b_1^k + t_8 b_4^k + t_9 b_7^k \\
e_8^k &= t_7 b_2^k + t_9 b_8^k \\
e_9^k &= t_7 b_3^k + t_8 b_6^k + t_9 b_9^k
\end{aligned}$$

Step 4. Compute $\mathrm{E_k}$ for $\mathrm{k} = 2\mathrm{l}, 1 \leq \mathrm{l} \leq \mathrm{m}$.

$$\begin{aligned}
e_1^k &= k_1 b_1^k + k_2 b_4^k + k_3 b_7^k \\
e_2^k &= k_1 b_2^k + k_3 b_8^k \\
e_3^k &= k_1 b_3^k + k_2 b_6^k + k_3 b_9^k \\
e_4^k &= k_4 b_1^k + k_5 b_4^k + k_6 b_7^k \\
e_5^k &= k_4 b_2^k + k_6 b_7^k \\
e_6^k &= k_4 b_3^k + k_5 b_6^k + k_6 b_9^k \\
e_7^k &= k_7 b_1^k + k_8 b_4^k + k_9 b_7^k
\end{aligned}$$

$$e_8^k = k_7 b_2^k + k_9 b_8^k$$
$$e_9^k = k_7 b_3^k + k_8 b_6^k + k_9 b_9^k$$

Step 5. Solve d_k for $k = 2l + 1, 0 \leq l \leq m$.

$$d_k = \left(e_2^k + t_2 x_k\right) \begin{vmatrix} e_6^k & e_4^k \\ e_9^k & e_7^k \end{vmatrix} + \left(e_5^k + t_5 x_k\right) \begin{vmatrix} e_1^k & e_7^k \\ e_3^k & e_9^k \end{vmatrix} + \left(e_8^k + t_8 x_k\right) \begin{vmatrix} e_3^k & e_1^k \\ e_6^k & e_4^k \end{vmatrix}$$

Step 6. Solve d_k for $k = 2l, 1 \leq l \leq m$.

$$d_k = \left(e_2^k + k_2 x_k\right) \begin{vmatrix} e_6^k & e_4^k \\ e_9^k & e_7^k \end{vmatrix} + \left(e_5^k + k_5 x_k\right) \begin{vmatrix} e_1^k & e_7^k \\ e_3^k & e_9^k \end{vmatrix} + \left(e_8^k + k_8 x_k\right) \begin{vmatrix} e_3^k & e_1^k \\ e_6^k & e_4^k \end{vmatrix}$$

Step 7. Substitute for $x_k = b_5^k$.
Step 8. Construct B_k.
Step 9. Construct M.
Step 10. End of algorithm.

Example: Let us consider the message matrix for the following message text:

"MINESWEEPERfor:GaussLeonardoNumbers"

Using the message text, we get the following message matrix M:

$$M = \begin{bmatrix} M & I & N & E & S & W \\ E & E & P & E & R & f \\ o & r & : & G & a & u \\ s & s & - & L & e & o \\ n & a & r & d & o & N \\ u & m & b & e & r & s \end{bmatrix}$$

Coding Algorithm

Step 1. We divide the message matrix M of size 6×6 into block matrices, denoted by $B_k(1 \leq k \leq 4$, proceeding from left to right, where each block is of size is 3×3:

$$B_1 = \begin{bmatrix} M & I & N \\ E & E & P \\ o & r & : \end{bmatrix}, B_2 = \begin{bmatrix} E & S & W \\ E & R & f \\ G & a & u \end{bmatrix}$$
$$B_3 = \begin{bmatrix} s & s & - \\ n & a & r \\ u & m & b \end{bmatrix}, B_4 = \begin{bmatrix} L & e & o \\ d & o & N \\ e & r & s \end{bmatrix}$$

Step 2. $n = 4$.
Step 3. The numerical representations can be shown as follows:

$$B_1 = \begin{bmatrix} 77 & 73 & 78 \\ 69 & 69 & 80 \\ 111 & 114 & 58 \end{bmatrix}, B_2 = \begin{bmatrix} 69 & 83 & 87 \\ 69 & 82 & 102 \\ 71 & 97 & 117 \end{bmatrix}$$
$$B_3 = \begin{bmatrix} 115 & 115 & 45 \\ 110 & 97 & 114 \\ 117 & 109 & 98 \end{bmatrix}, B_4 = \begin{bmatrix} 76 & 101 & 111 \\ 100 & 111 & 78 \\ 101 & 114 & 115 \end{bmatrix}$$

Step 4. Next, we calculate the determinants d_k of the blocks B_k:

$$d_1 = \det(T_4 B_1) = -87384$$
$$d_2 = \det(K_4 B_2) = -166752 - 55584i$$
$$d_3 = \det(T_4 B_3) = -51140$$
$$d_4 = \det(K_4 B_4) = -605940 - 201980i$$

Step 5. Using Step 3 and Step 4, we obtain the encoded Matrix C:

$$C = \begin{bmatrix} -87384 & 77 & 73 & 78 & 69 & 80 & 111 & 114 & 58 \\ -166752 - 55584i & 69 & 83 & 87 & 69 & 102 & 71 & 97 & 117 \\ -51140 & 115 & 115 & 45 & 110 & 114 & 117 & 109 & 98 \\ -605940 - 201980i & 76 & 101 & 111 & 100 & 78 & 101 & 114 & 115 \end{bmatrix}$$

Step 6. End of algorithm.

Decoding Algorithm

Step 1. First, we compute the matrix T_4:

$$T_4 = \begin{bmatrix} 41 & 25 & 15 \\ 25 & 15 & 9 \\ 15 & 9 & 5 \end{bmatrix}.$$

Step 2. Then, we compute the matrix K_4:

$$K_4 = \begin{bmatrix} 41 + 25i & 25 + 15i & 15 + 9i \\ 25 + 15i & 15 + 9i & 9 + 5i \\ 15 + 9i & 9 + 5i & 5 + 3i \end{bmatrix}.$$

Step 3. E_k for $k = 2l + 1, 0 \leq l \leq m$.

$$E_1 = \begin{bmatrix} 6547 & 4703 & 6068 \\ 3959 & 2851 & 3672 \\ 2331 & 1665 & 2180 \end{bmatrix}$$
$$E_3 = \begin{bmatrix} 9220 & 6350 & 6165 \\ 5578 & 3856 & 3717 \\ 3300 & 2270 & 2191 \end{bmatrix}$$

Step 4. $\mathrm{E_k}$ for $\mathrm{k} = 2\mathrm{l}, 1 \leq \mathrm{l} \leq \mathrm{m}$.

$$E_2 = \begin{bmatrix} 5619 + 3399i & 4858 + 2948i & 7872 + 4758i \\ 3399 + 2011i & 2948 + 1730i & 4758 + 2808i \\ 2011 + 1179i & 1730 + 1038i & 2808 + 1644i \end{bmatrix}$$
$$E_4 = \begin{bmatrix} 7131 + 4309i & 5851 + 3551i & 8226 + 4980i \\ 4309 + 2545i & 3551 + 2085i & 4980 + 2942i \\ 2545 + 1487i & 2085 + 1251i & 2942 + 1734i \end{bmatrix}$$

Step 5. Solve d_k for $k = 2l + 1, 0 \leq l \leq m$.

$$-87384 =$$
$$(4703 + 25x_1)(-71188) + (2851 + 15x_1)(127952) + (1665 + 9x_1)(-17372)$$
$$\Rightarrow x_1 = 69.$$
$$-51140 =$$
$$(6350 + 25x_3)(44702) + (3856 + 15x_3)(-143480) + (2270 + 9x_3)(117630)$$
$$\Rightarrow x_3 = 97.$$

Step 6. Solve d_k for $k = 2l, 1 \leq l \leq m$.

$$
\begin{gathered}
-166752-55584i \\
=[(4858+2948i)+(25+15i)x_2]\times(19398+21726i) \\
+[2948+1730i)+(15+9i)x_2]\times(-30714-67398i) \\
+[(1730+1038i)+(9+5i]x_2]\times(-2220+52440i) \\
\Rightarrow x_2=82. \\
-605940-201980i \\
=[(5851+3551i)+(25+15i)x_4]\times(35294-66546i) \\
+[(2085+1251i)+(15+9i)x_4]\times(-22314+136070i) \\
+[(1730+1038i)+(9+5i)x_4]\times(-63568-44232i) \\
\Rightarrow x_4=111.
\end{gathered}
$$

Step 7. We substitute the recovered values of x_k as follows:

$$
\begin{gathered}
x_1=b_5^1=69 \\
x_2=b_5^2=82 \\
x_3=b_5^3=97 \\
x_4=b_5^4=111
\end{gathered}
$$

Step 8. We reconstruct the block matrices B_k:

$$
B_1=\begin{bmatrix}77&73&78\\69&69&80\\111&114&58\end{bmatrix},B_2=\begin{bmatrix}69&83&87\\69&82&102\\71&97&117\end{bmatrix}
$$
$$
B_3=\begin{bmatrix}115&115&45\\110&97&114\\117&109&98\end{bmatrix},B_4=\begin{bmatrix}76&101&111\\101&111&78\\101&114&115\end{bmatrix}
$$

Finally, we obtain the original message matrix $\boldsymbol{M}$:

$$
M=\begin{bmatrix}77&73&78&69&83&87\\69&69&80&69&82&102\\111&114&56&71&97&117\\115&115&45&76&101&111\\110&97&114&100&111&78\\117&109&98&101&114&115\end{bmatrix}=\begin{bmatrix}M&I&N&E&S&W\\E&E&P&E&R&f\\o&r&:&G&a&u\\s&s&-&L&e&o\\n&a&r&d&o&N\\u&m&b&e&r&s\end{bmatrix}
$$

Step 9. End of algorithm.

3. COMPARISONS AND CONCLUSION

In this section, we discuss the differences between the proposed method and existing approaches. First, in previous studies [11-12], the value of the parameter n depends on the number of block matrices, whereas in our method, n can be chosen independently. Additionally, while other methods generate custom character tables based on n, our approach utilizes the decimal structure of ASCII. This provides the algorithms with a standardized framework and ensures more consistent operations. Furthermore, extending the recurrence relations to the complex plane has significantly strengthened the coding/ decoding algorithm, making it more flexible and robust.

Acknowledgement: *The authors thank the anonymous referees for their comments and valuable suggestions that improved the presentation of the manuscript.*

REFERENCES

[1] Koshy, T., *Fibonacci and Lucas Numbers with Applications*, John Wiley & Sons, New York, 2019.

[2] Kızılates, C., Yachani, S. T., *Computational Mathematics and Mathematical Physics*, **65**, 674, 2025.

[3] Jordan, J. H., *Fibonacci Quarterly*, **3**, 315, 1965.

[4] Horadam, A. F., *American Math Monthly*, **70**, 289, 1963.

[5] Asci, M., Aydinyuz, S., *Journal of Information and Optimization Sciences*, **42**(8), 1765, 2021.

[6] Kızılates, C., *Contributions to Discrete Mathematics*, **19**(4), 71818, 2024.

[7] Gurel, E., Asci, M., *Ars Combinatoria*, **135**, 345, 2017.

[8] Catarino, P., Borges, A., *Journal of Information and Optimization Sciences*, **39**, 1363, 2018.

[9] Catarino, P., Borges, A., *Acta Mathematica Universitatis Comenianae*, **89**, 75, 2020.

[10] Tasci, D., *Contributions to Mathematics*, **7**, 34, 2023.

[11] Tas, N., Ucar, S., Ozgur, N. Y., Kaymak, O. O., *Discrete Mathematics Algorithms and Applications*, **10**(2), 1850028, 2018.

[12] Ucar, S., Tas, N., Ozgur, N. Y., *Mathematical Sciences and Applications E-Notes*, **7**(1), 62, 2019.

[13] Oztunc Kaymak, O., *Balıkesir University Journal of Science Institute*, **26**(1), 226, 2024.

[14] Kuloglu, B., Ozkan, E., *Mathematica Montisnigri*, **57**, 1, 2023.

[15] Eser, E., Kuloglu, B., Ozkan, E., *Applied Mathematics E-Notes*, **24**, 274, 2024.

[16] Shtayat, J., Al-Kateeb, A., arXiv preprint, arXiv:1907.02007, 2019.

[17] Ozcevik, S. B., Dertli, A., *Journal of Science and Arts*, **23**(4), 897, 2023.

[18] Alenezi, M. N., Alabdulrazzaq, H., Mohammad, N. Q., *International Journal of Communication Networks and Information Security*, **2**, 256, 2020.

[19] Babu, V. M., Ganesh, T. S., Ramraj, K., *International Journal of Scientific and Research Publications*, **4**(4), 2014.

[20] Flaut, C., *Journal of Difference Equations and Applications*, **25**(7), 905, 2019.

[21] Irge, V., Soykan, Y., *Discrete Mathematics Algorithms and Applications*, **16**(8), 2350113, 2024.

[22] Stakhov, A. P., *Chaos Solitons Fractals*, **32**, 1138, 2007.

[23] Stakhov, A. P., *Chaos Solitions Fractals*, **30**, 56, 2006.

[24] Ozgur N., Ucar, S., Tas, N., Oztunc Kaymak, O., *Discrete Mathematics Algorithms and Applications*, **17**(4), 2450068, 2024.

[25] Tas, N., Ucar, S., Ozgur, N. Y., *Journal of Balıkesir University Institute of Science and Technology*, **21**, 306, 2019.