Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.017898

ARTICLE

k-Order Fibonacci Polynomials on AES-Like Cryptology

Mustafa Asci and Suleyman Aydinyuz*

Pamukkale University, Kinikli, Denizli, 20160, Turkey

*Corresponding Author: Suleyman Aydinyuz. Email: aydinyuzsuleyman@gmail.com

Received: 15 June 2021 Accepted: 26 October 2021

ABSTRACT

The Advanced Encryption Standard (AES) is the most widely used symmetric cipher today. AES has an important place in cryptology. Finite field, also known as Galois Fields, are cornerstones for understanding any cryptography. This encryption method on AES is a method that uses polynomials on Galois fields. In this paper, we generalize the AES-like cryptology on 2×2 matrices. We redefine the elements of k-order Fibonacci polynomials sequences using a certain irreducible polynomial in our cryptology algorithm. So, this cryptology algorithm is called AES-like cryptology on the k-order Fibonacci polynomial matrix.

KEYWORDS

Fibonacci numbers; Fibonacci polynomials; k-order Fibonacci polynomials; Fibonacci matrix; k-order Fibonacci polynomial matrix; Galois field

1 Introduction

AES (Advanced Encryption Standard) is a standard offered for encryption of electronic data. AES, adopted by the American government, is also used as a defacto encryption standard in the international arena. It replaces DES (Data Encryption Standard). The encryption algorithm defined by AES is a symmetric-key algorithm in which the keys used in both encryption and decryption of encrypted text are related to each other. The encryption and decryption keys are the same for AES.

The algorithm standardized with AES was created by making some changes to the Rijndael algorithm, which was mainly developed by Vincent Rijmen and Joan Daeman. Rijndael is a name obtain using the developers' names: RIJmen and DAEmen.

AES is based on the design known as substitution-permutation. Its predecessor, DES, is an algorithm designed in Feistel structure. AES' software and hardware performance is high. The 128-bit input block has a key length of 128, 192 and 256 bits. Rijndael, on which AES is based, supports input block lengths that are multiples of 32 between 128 and 256 bits and key lengths longer than 128 bits. Therefore, in the standardization process, key and input block lengths were restricted. AES works on a 4×4 column-priority byte matrix called state. Operations in the matrix are also performed on a special finite field.

The algorithm consists of identical rounds that transform a certain number of repeating input open text into output ciphertext. Each cycle consists of four steps, except for the last cycle. These cycles are applied in reserve order to decode the encrypted text. The number of repetitions of cycles is a function of the key length according to Table 1.

Table 1: Key lengths and number of rounds for AES

Key lengths	Cycles
128 bits	10
192 bits	12
256 bits	14

These cycles include key addition, byte substitution, ShiftRow and MixColumn. We can see these cycles in Fig. 1. One can see detailed information about AES in Fig. 2 [1].

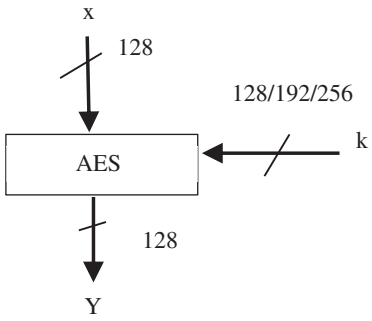

Figure 1: AES input/output parameters

A finite field, sometimes also called Galois field, is a set with a finite number of elements. Roughly speaking, a Galois field is a finite set of elements in which we can add, subtract, multiply and invert. Before we introduce the definition of a field, we first need the concept of a simple algebraic structure, a field.

1.1 Definition Field

A field F is a set of elements with the following properties:

- All elements of F form an additive group with the group operation "+" and the neutral element 0.
- All elements of F except 0 form a multiplicative group with the group operation "×" and the neutral element 1.
- When the two group operations are mixed, the distributivity law holds, i.e., for all $a, b, c \in F$: $a(b+c) = (ab) + (ac)$.

Galois field arithmetic is the most widely used field involving matrix operations. One can see detailed information about the Galois field and the operations performed on it in [2]. Also, you can find information on the classical cryptology benefit in [3].

In extension fields $GF(2^m)$ elements are not represented as integers but as polynomials with coefficients in $GF(2)$. The polynomials have a maximum degree of $m-1$, so that there are m

coefficients in total for every element. In the field $GF\left(2^8\right)$, which is used in AES, each element $A \in GF\left(2^8\right)$ is thus represented as

$$A(x) = a_7 x^7 + \ldots + a_1 x + a_0, \quad a_i \in GF(2) = \{0, 1\}.$$

Note that there are exactly $256 = 2^8$ such polynomials. The set of these 256 polynomials is the finite field $GF\left(2^8\right)$. It is also important to observe that every polynomial can simply be stored in digital form as an 8-bit vector

$$A = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0).$$

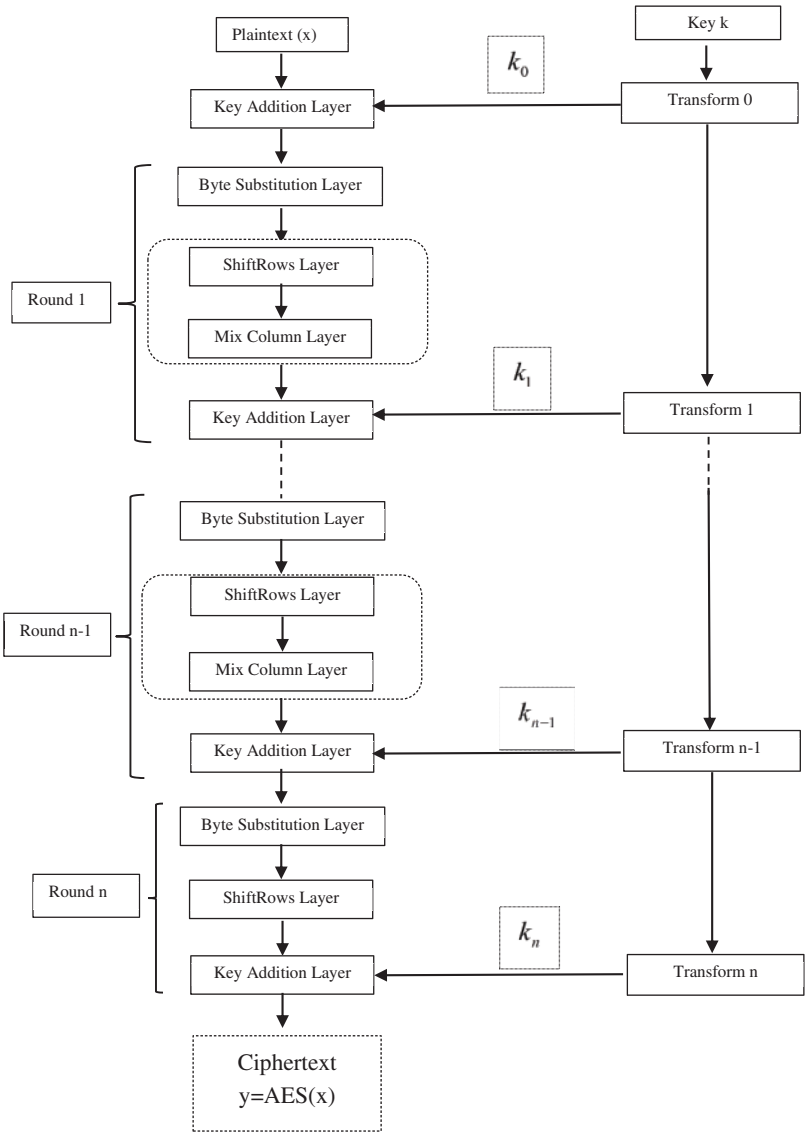

Figure 2: AES encryption block diagram

In particular, we do not have to store the factors x^7, x^6, etc. It is clear from the bit positions to which power x^i each coefficient belongs.

Fibonacci numbers are defined by the recurrence relation of $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with the initial conditions $F_0 = 0$ and $F_1 = 1$. There are a lot of generalizations of Fibonacci numbers satisfied and studied by some authors. For more information one can see in [4–8]. The Fibonacci Q-matrix is defined in [9,10] as follows:

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

and *nth* power of the Fibonacci $Q-$matrix is shown in [11–13] by

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

Fibonacci polynomials that belong to the large polynomial classes are defined by a recurrence relation similar to Fibonacci numbers. The Belgian mathematician Eugene Charles Catalan and the German mathematician E. Jacobsthal were studied Fibonacci polynomials in 1983. The polynomials $f_n(x)$ studied by Catalan are defined by the recurrence relation

$$f_n(x) = x f_{n-1}(x) + f_{n-2}(x)$$

where $f_0(x) = 0, f_1(x) = 1, f_2(x) = x$ and $n \geq 3$. Fig. 3 notice that for $x = 1$, $f_n(1) = F_n$, F_n is *nth* Fibonacci number.

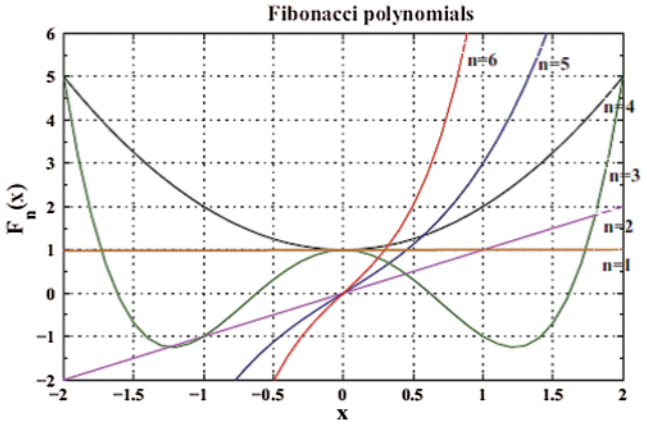

Figure 3: The behavior of the first six Fibonacci polynomials

In [14], the k-order Fibonacci polynomial is defined by A. N. Philippou, C. Georghiou and G. Philippou in 1983.

The sequence of polynomials $\left\{f_n^{(k)}(x)\right\}_{n=0}^{\infty}$ is said to be the sequel of Fibonacci polynomials of order k if $f_0^{(k)}(x)=0, f_1^{(k)}(x)=1$ and

$$f_n^{(k)}(x)=\begin{cases}\sum_{i=1}^{n} x^{k-i} f_{n-i}^{(k)}(x) & if \quad 2\le n\le k\\ \sum_{i=1}^{k} x^{k-i} f_{n-i}^{(k)}(x) & if \quad n\ge k+1\end{cases}.$$

Kizilates et al. studied a new generalization of convolved $(p,q)-$Fibonacci and $(p,q)-$Lucas polynomials in [4]. Also, Qi et al. gave a closed formula for the Horadam polynomials in terms of a tridiagonal determinant in 2019 in [15] and Kizilates et al. defined several determinantel expressions of generalized tribonacci polynomials and sequences in [5]. In [6], Kizilates et al. introduced new families of three-variable polynomials coupled with well-known polynomials and numbers in 2019. New families of Horadam numbers associated with finite operators and their applications were studied by Kizilates in [7].

In [16], Basu et al. introduced the generalized relations among the code elements for Fibonacci coding theory in 2009. In 2014, Basu et al. defined a new coding theory for Tribonacci matrices in [17] and they expended the coding theory on Fibonacci n-step numbers in [18]. Also, Basu et al. defined generalized Fibonacci n-step polynomials and stated a new coding theory called generalized Fibonacci n-step polynomials coding theory in [19].

In [19], for $k\ge 2$

$$Q_k(x)=\begin{bmatrix} x^{k-1} & x^{k-2} & x^{k-3} & \cdots & x & 1\\ 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & \cdots & 0 & 0\\ 0 & 0 & \cdots & 0 & 1 & 0\end{bmatrix}_{k\times k}$$

and

$$Q_k^n(x)=\begin{bmatrix} F_{n+k-1}^{(k)}(x) & x^{k-2}F_{n+k-2}^{(k)}(x)+x^{k-3}F_{n+k-3}^{(k)}(x)+\cdots+F_n^{(k)}(x)\\ F_{n+k-2}^{(k)}(x) & x^{k-2}F_{n+k-3}^{(k)}(x)+x^{k-3}F_{n+k-4}^{(k)}(x)+\cdots+F_{n-1}^{(k)}(x)\\ \vdots & \vdots\\ F_{n+1}^{(k)}(x) & x^{k-2}F_n^{(k)}(x)+x^{k-3}F_{n-1}^{(k)}(x)+\cdots+F_{n-k+2}^{(k)}(x)\\ F_n^{(k)}(x) & x^{k-2}F_{n-1}^{(k)}(x)+x^{k-3}F_{n-2}^{(k)}(x)+\cdots+F_{n-k+1}^{(k)}(x)\end{bmatrix}$$

$$\begin{bmatrix} x^{k-3}F_{n+k-2}^{(k)}(x)+x^{k-4}F_{n+k-3}^{(k)}(x)+\cdots+F_{n+1}^{(k)}(x) & \cdots & F_{n+k-2}^{(k)}(x)\\ x^{k-3}F_{n+k-3}^{(k)}(x)+x^{k-4}F_{n+k-4}^{(k)}(x)+\cdots+F_n^{(k)}(x) & \cdots & F_{n+k-3}^{(k)}(x)\\ \vdots & \vdots & \vdots\\ x^{k-3}F_n^{(k)}(x)+x^{k-4}F_{n-1}^{(k)}(x)+\cdots+F_{n-k+3}^{(k)}(x) & \cdots & F_n^{(k)}(x)\\ x^{k-3}F_{n-1}^{(k)}(x)+x^{k-4}F_{n-2}^{(k)}(x)+\cdots+F_{n-k+2\ldots}^{(k)}(x) & \cdots & F_{n-1}^{(k)}(x)\end{bmatrix} \tag{1}$$

where $F_n^{(k)}(x)$ is a k-order Fibonacci polynomials.

Diskaya et al. created a new encryption algorithm (known as AES-like) by using the AES algorithm in [20]. They created the encryption algorithm by splitting the message text into 2×2 block matrices using Fibonacci polynomials.

Fibonacci polynomials have many applications in algebra. In recent years, we see that these polynomials have many uses in the field of engineering. Also, Fibonacci polynomials are used in solving differential equations. These solutions are used in engineering and science, adding new approaches to the solution of engineering problems. Mirzae and Hoseini solved singularly perturbed differential-difference equations arising in science and engineering with Fibonacci polynomials in [21]. Also, in [22], Haq et al. studied approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials.

In this paper, we generalize the encryption algorithm given in [20] and study the encryption made with the 2×2 type block matrix operation to the $k \times k$ type in Galois field. We redefine the elements of k-order Fibonacci polynomials sequences using a certain irreducible polynomial in our cryptology algorithm. The algorithm consist of four steps as in the AES encryption algorithm. The encryption algorithm defined in this algorithm is a symmetric-key algorithm in which the keys used in both encryption and decryption of encrypted text are related to each other. The encryption and decryption keys are the same like AES. So, this cryptology algorithm is called AES-like cryptology algorithm on the k-order Fibonacci polynomials.

2 The k-Order Fibonacci Polynomials Blocking Algorithm

In this chapter, we redefine the elements of k-order Fibonacci polynomial sequences using a certain irreducible polynomial in our coding algorithm. In extension fields $GF(2^m)$ elements are not represented as integers but as polynomials with coefficients in $GF(2)$. Throughout this section, we take $m = 5$ for next process. Since $m = 5$, we consider the finite Galois field containing 32 elements in this algorithm and this Galois field is denoted as $GF\left(2^5\right)$. Note that there are exactly $2^5 = 32$ such polynomials. The set of these 32 polynomials is the finite field $GF\left(2^5\right)$. Each elements of this polynomials correspond to one letter of the alphabet.

The AES encryption algorithm uses the $P(x) = x^8 + x^4 + x^3 + x + 1$ polynomial as the irreducible polynomial.

The irreducible polynomials of $GF\left(2^5\right)$ are as follows:

$x^5 + x^2 + 1$

$x^5 + x^3 + 1$

$x^5 + x^3 + x^2 + x + 1$

$x^5 + x^4 + x^3 + x + 1$

$x^5 + x^4 + x^3 + x^2 + 1$

$x^5 + x^4 + x^2 + x + 1.$

In this paper, we consider the irreducible polynomials as $P(x) = x^5 + x^2 + 1$. We can also diversify our encryption algorithm by using other irreducible polynomials.

Definition: In [8], the Fibonacci polynomial sequence $\{f_n(x)\}_{n \geq 0}$ is $f_0(x) = 0, f_1(x) = 1$ and $f_{n+2}(x) = x f_{n+1}(x) + f_n(x)$.

For later use the first few terms of the sequence Fibonacci polynomials can be seen in the following Table 2 and a few the irreducible polynomials for Fibonacci polynomials are given as Table 3.

Table 2: Fibonacci polynomials

n	0	1	2	3	4	5	⋯
$f_n(x)$	0	1	x	x^2+1	x^3+2x	x^4+3x^2+1	⋯

Table 3: Irreducible polynomials for Fibonacci polynomials

n	$f_n(x)$	Z_2
0	0	mod 2
1	1	mod 2
2	x	mod 2
3	x^2+1	mod 2
4	x^3	mod 2
5	x^4+x^2+1	mod 2
6	x^2+x+1	mod 2
7	x^4+x^3+x+1	mod 2
8	x^4+x^2	mod 2
9	x^4+x^2+x	mod 2
⋮	⋮	⋮

Polynomials of the Galois field are equivalent of each alphabet in Table 4 is as following:

Table 4: Alphabet table

No.	Bit	Polynomial	Alphabet
0	00000	0	A
1	00001	1	B
2	00010	x	C
3	00011	$x+1$	Ç
4	00100	x^2	D
5	00101	x^2+1	E
6	00110	x^2+x	F
7	00111	x^2+x+1	G
8	01000	x^3	Ğ
9	01001	x^3+1	H
10	01010	x^3+x	I

(Continued)

Table 4 (continued)

No.	Bit	Polynomial	Alphabet
11	01011	x^3+x+1	İ
12	01100	x^3+x^2	J
13	01101	x^3+x^2+1	K
14	01110	x^3+x^2+x	L
15	01111	x^3+x^2+x+1	M
16	10000	x^4	N
17	10001	x^4+1	O
18	10010	x^4+x	Ö
19	10011	x^4+x+1	P
20	10100	x^4+x^2	R
21	10101	x^4+x^2+1	S
22	10110	x^4+x^2+x	Ş
23	10111	x^4+x^2+x+1	T
24	11000	x^4+x^3	U
25	11001	x^4+x^3+1	Ü
26	11010	x^4+x^3+x	V
27	11011	x^4+x^3+x+1	W
28	11100	$x^4+x^3+x^2$	X
29	11101	$x^4+x^3+x^2+1$	Y
30	11110	$x^4+x^3+x^2+x$	Z
31	11111	$x^4+x^3+x^2+x+1$	Q

Now, we obtain our encryption algorithm in line preliminary information we have given.

2.1 The k-Order Fibonacci Encryption Algorithm: The Coding Algorithm

- **Step 1:** We can consider the message text of length n and assume that each letter represents one length.
- **Step 2:** We can choose arbitrary value of k and n. The k-value we choose determine which order Fibonacci polynomials to use. We can create the matrix $Q_k^n(x)$ in Eq. (1) according to the $k$and n value we have chosen. Our message text is divided into blocks according to the value k. We get matrices of $k\times 1$ type. We get a new matrix by multiplying the $k\times 1$ type matrix with the $Q_k^n(x)$. Our new message is created by looking at the values in the matrix we obtained from the alphabet table.
- **Step 3:** We multiply the message matrix we just obtained by the invertible key matrix. In this paper, we accept the key matrix as follows:

1. $$KeyMatrix=\begin{bmatrix} B & B & C \\ Ç & E & \breve{G} \\ K & E & Y \end{bmatrix}=\begin{bmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \\ 13 & 5 & 29 \end{bmatrix}$$

If there is an ascending 2 letters in the text, it letters is multiplied by 2. Key matrix in 2×2:

2. $KeyMatrix = \begin{bmatrix} E & A \\ O & D \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 17 & 4 \end{bmatrix}$.

- **Step 4:** The text created in the 3th step is collected sequentially with the k-order Fibonacci polynomials by starting from left and our encrypted message is created.

$$\sum_{i=1}^{n} F_i^{(k)}(x) = F_1^{(k)}(x) + F_2^{(k)}(x) + \ldots + F_n^{(k)}(x).$$

2.2 The k-Order Fibonacci Decryption Algorithm: The Decoding Algorithm

- **Step 1:** We can consider encrypted a text of length n and assume that each letter represents one length.
- **Step 2:** The text created is addition sequentially with the k-order Fibonacci polynomials by starting from the left and our new message is created.

$$\sum_{i=1}^{n} F_i^{(k)}(x) = F_1^{(k)}(x) + F_2^{(k)}(x) + \ldots + F_n^{(k)}(x).$$

- **Step 3:** We multiply the message matrix we just obtained by inverse of the 1. key matrix.

$$InverseKeyMatrix = \begin{bmatrix} F & \check{C} & Z \\ S & \check{G} & N \\ V & T & G \end{bmatrix} = \begin{bmatrix} 6 & 3 & 30 \\ 21 & 8 & 16 \\ 26 & 23 & 7 \end{bmatrix}$$

If there is an ascending 2 letters in the text, it letters is multiplied by 2. Inverse key matrix in 2×2:

$$Inverse2.KeyMatrix = \begin{bmatrix} T & A \\ \check{G} & H \end{bmatrix} = \begin{bmatrix} 23 & 0 \\ 8 & 9 \end{bmatrix}.$$

- **Step 4:** We can obtain the matrix $\left(Q_k^n(x)\right)^{-1}$ according to the k and n we have chosen. Our text is divided into blocks according to the value k. We get matrices of $k \times 1$ type. We get a new matrix by multiplying the $k \times 1$ type matrix with the $\left(Q_k^n(x)\right)^{-1}$. Our new message is created by looking at the values in the matrix we obtained from the alphabet table. We can obtain our text message text.

2.3 Illustrative Examples for AES-Like Cryptology on the k-Order Fibonacci Polynomial Matrix

Example 1: Let us consider the message text for the following:

"HELLO"

Application to the Coding Algorithm:

- **Step 1:** "HELLO" is 5 letters. In this example, we encrypt process by choosing $n = 5$ (We can choose n arbitrarily).

- **Step 2:** For $k=3$ and $n=5$, we can use Tribonacci polynomials for encryption. We can get as

$$Q_3^5(x)=\begin{bmatrix} x^2 & x & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^5=\begin{bmatrix} x^4+1 & x^4+x+1 & x^3+x^2 \\ x^3+x^2 & x^2 & x^3+x+1 \\ x^3+x+1 & x^2+1 & x^4+x \end{bmatrix}.$$

It is known that

$9=(01001)=x^3+1=H$

$5=(00101)=x^2+1=E$

$14=(01110)=x^3+x^2+x=L$

$17=(10001)=x^4+1=O$

So, it is

$$Q_3^5(x).\begin{bmatrix} H \\ E \\ L \end{bmatrix}=\begin{bmatrix} x^4+1 & x^4+x+1 & x^3+x^2 \\ x^3+x^2 & x^2 & x^3+x+1 \\ x^3+x+1 & x^2+1 & x^4+x \end{bmatrix}\begin{bmatrix} x^3+1 \\ x^2+1 \\ x^3+x^2+x \end{bmatrix}$$

$$=\begin{bmatrix} x^4+x^3+x \\ x^4+x^3+x \\ x^3+x^2+x+1 \end{bmatrix}=\begin{bmatrix} V \\ V \\ M \end{bmatrix}$$

Since the word "HELLO" has 5 letters, we divide it into blocks of 3×1 and 2×1. So now we encrypt the 2×1 block with the usual Fibonacci polynomial matrix.
We can get in Eq. (1) as

$$Q_2^5(x)=\begin{bmatrix} x & 1 \\ 1 & 0 \end{bmatrix}^5$$

$$=\begin{bmatrix} x^2+x+1 & x^4+x^2+1 \\ x^4+x^2+1 & x^3 \end{bmatrix}$$

So, it is

$$Q_2^5(x).\begin{bmatrix} L \\ O \end{bmatrix}=$$

$$=\begin{bmatrix} x^2+x+1 & x^4+x^2+1 \\ x^4+x^2+1 & x^3 \end{bmatrix}\begin{bmatrix} x^3+x^2+x \\ x^4+1 \end{bmatrix}$$

$$=\begin{bmatrix} x^3+x^2+1 \\ x^4+x^2 \end{bmatrix}$$

$$=\begin{bmatrix} K \\ R \end{bmatrix}$$

It results "$HELLO$" $\rightarrow$ "$VVMKR$".

- **Step 3:** We multiply the message matrix we just obtained by the invertible 1. Key matrix. Turn into blocks of 3s and multiply with the key matrix.

$$\begin{bmatrix} B & B & C \\ Ç & E & \breve{G} \\ K & E & Y \end{bmatrix}\begin{bmatrix} V \\ V \\ M \end{bmatrix} = \begin{bmatrix} 1 & 1 & x \\ x+1 & x^2+1 & x^3 \\ x^3+x^2+1 & x^2+1 & x^4+x^3+x^2+1 \end{bmatrix}\begin{bmatrix} x^4+x^3+x \\ x^4+x^3+x \\ x^3+x^2+x+1 \end{bmatrix}$$

$$= \begin{bmatrix} x^4+x^3+x^2+x \\ 1 \\ x^2 \end{bmatrix} = \begin{bmatrix} Z \\ B \\ D \end{bmatrix}$$

Since we have 2 letters left, we can use our 2. Key matrix,

$$\begin{bmatrix} E & A \\ O & D \end{bmatrix}\begin{bmatrix} K \\ R \end{bmatrix} = \begin{bmatrix} x^2+1 & 0 \\ x^4+1 & x^2 \end{bmatrix}\begin{bmatrix} x^3+x^2+1 \\ x^4+x^2 \end{bmatrix}$$

$$= \begin{bmatrix} x^4+x^3+x^2 \\ x^4+x^3+1 \end{bmatrix} = \begin{bmatrix} X \\ \ddot{U} \end{bmatrix}$$

It results "$VVMKR$" $\rightarrow$ "$ZBDX\ddot{U}$.

- **Step 4:** We get

$Z + T_1(x) = x^4 + x^3 + x^2 + x + 1 = Q$

$B + T_2(x) = 1 + x^2 = E$

$D + T_3(x) = x^4 + x^2 + x = S$

$X + T_4(x) = x^4 + x^2 + x + 1 = T$

$\ddot{U} + T_5(x) = x^4 + x^2 + 1 = S$

where $T_n(x)$ is a *nth* Tribonacci polynomial.

It results "$ZBDX\ddot{U}$" $\rightarrow$ "$QEŞTS$".

Application to the Decoding Algorithm:

- **Step 1:** We can get as

$Q + T_1(x) = x^4 + x^3 + x^2 + x = Z$

$E + T_2(x) = 1 = B$

$S + T_3(x) = x^2 = D$

$T + T_4(x) = x^4 + x^3 + x^2 = X$

$S + T_5(x) = x^4 + x^3 + 1 = \ddot{U}$

where $T_n(x)$ is a *nth* Tribonacci polynomial.

It results "$QEŞTS$" $\rightarrow$ "$ZBDX\ddot{U}$".

- **Step 2:** We multiply the message matrix we just obtained by inverse of the 1. Key matrix.

$$\begin{bmatrix} F & Ç & Z \\ S & Ğ & N \\ V & T & G \end{bmatrix} \begin{bmatrix} Z \\ B \\ D \end{bmatrix} = \begin{bmatrix} V \\ V \\ M \end{bmatrix}$$

Since we have 2 letters left, we can use our 2. Inverse key matrix.

$$\begin{bmatrix} T & A \\ Ğ & H \end{bmatrix} \begin{bmatrix} X \\ Ü \end{bmatrix} = \begin{bmatrix} K \\ R \end{bmatrix}$$

It results "$ZBDXÜ$" → "$VVMKR$".

- **Step 3:** We can obtain the matrix $\left(Q_k^n(x)\right)^{-1}$ according to the k and n value we have chosen. For $k=3$ and $n=5$; we get as

$$\left(Q_3^5(x)\right)^{-1} = \begin{bmatrix} 0 & x & x^3+1 \\ x^3+1 & 1 & x^4 \\ x^4 & x+1 & x^2 \end{bmatrix} = \begin{bmatrix} A & C & H \\ H & B & N \\ N & Ç & D \end{bmatrix}$$

So, it is

$$\left(Q_3^5(x)\right)^{-1} \begin{bmatrix} V \\ V \\ M \end{bmatrix} = \begin{bmatrix} 0 & x & x^3+1 \\ x^3+1 & 1 & x^4 \\ x^4 & x+1 & x^2 \end{bmatrix} \begin{bmatrix} x^4+x^3+x \\ x^4+x^3+x \\ x^3+x^2+x+1 \end{bmatrix}$$

$$= \begin{bmatrix} x^3+1 \\ x^2+1 \\ x^3+x^2+x \end{bmatrix} = \begin{bmatrix} H \\ E \\ L \end{bmatrix}$$

Since we have 2 letters left, we can get $\left(Q_2^5(x)\right)^{-1}$ for $k=2$ and $n=5$ as

$$\left(Q_2^5(x)\right)^{-1} = \begin{bmatrix} x^3 & x^4+x^2+1 \\ x^4+x^2+1 & x^2+x+1 \end{bmatrix} = \begin{bmatrix} Ğ & S \\ S & I \end{bmatrix}$$

So, it is

$$\left(Q_2^5(x)\right)^{-1} \begin{bmatrix} K \\ R \end{bmatrix} = \begin{bmatrix} x^3 & x^4+x^2+1 \\ x^4+x^2+1 & x^2+x+1 \end{bmatrix} \begin{bmatrix} x^3+x^2+1 \\ x^4+x^2 \end{bmatrix}$$

$$= \begin{bmatrix} x^3+x^2+x \\ x^4+1 \end{bmatrix} = \begin{bmatrix} L \\ O \end{bmatrix}$$

It results "$VVMKR$" → "$HELLO$".

We have handled the example given in [20] again with the algorithm we created. The correct result was obtained as a result of the operation we have done. In addition, the encryption process performed with 2×2 block matrices in the other study was performed faster and easier with this method.

Example 2: Let us consider the message text for the following:

"PUBLIC"

Application to the Coding Algorithm

- **Step 1:** "PUBLIC" is 6 letters. In his example, we encrypt process by choosing $n=6$ (We can choose n arbitrarily. We do not have to choose the same number of letters as the number of n in our message text to be encrypted).
- **Step 2:** For $k=4$ and $n=6$, we can use Tetranacci polynomials for encryption. We can get as

$$Q_4^6(x)=\begin{bmatrix}x^3 & x^2 & x & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}^6$$

$$=\begin{bmatrix}x^4+x^3+x & x^3+x & x^3+1 & x^4+x^3+x^2+x+1\\ x^4+x^3+x^2+x+1 & x^4+x^3+1 & x^4+x^3+1 & x^4+x\\ x^4+x & x^4+x^3+x+1 & x^4+x^3+x+1 & x^4+x^3\\ x^4+x^3 & x^3+x^2 & x^4+x^2 & x^3+x^2+x\end{bmatrix}$$

It is known that

$19=(10011)=x^4+x+1=P$

$24=(11000)=x^4+x^3=U$

$1=(00001)=1=B$

$14=(01110)=x^3+x^2+x=L$

$10=(01010)=x^3+x=I$

$2=(00010)=x=C$

So, it is

$$Q_4^6(x)\begin{bmatrix}P\\U\\B\\L\end{bmatrix}=Q_4^6(x)\begin{bmatrix}x^4+x+1\\x^4+x^3\\1\\x^3+x^2+x\end{bmatrix}$$

$$=\begin{bmatrix}x^4+x^3+x^2+x\\x+1\\x^3+x^2+x\\x^3+x^2+x+1\end{bmatrix}=\begin{bmatrix}Z\\Ç\\L\\M\end{bmatrix}$$

Since the word "PUBLIC" has 6 letters, we divide it into blocks of 4×1 and 2×1. So now:
We encrypt the 2×1 block with the usual Fibonacci polynomial matrix.
We can get as

$$Q_2^6(x)\begin{bmatrix}I\\C\end{bmatrix}=\begin{bmatrix}x^4+x^3+x+1 & x^2+x+1\\ x^2+x+1 & x^4+x^2+1\end{bmatrix}\begin{bmatrix}x^3+x\\x\end{bmatrix}$$

$$= \begin{bmatrix} x^4+x^3+x+1 \\ x^4+x^3+x^2 \end{bmatrix} = \begin{bmatrix} W \\ X \end{bmatrix}$$

It results '$PUBLIC$' → '$ZÇLMWX$'.

- **Step 3:** We multiply the message matrix we just obtained by the invertible 1. Key matrix. Turn into blocks of 3s and multiply with the key matrix.

$$\begin{bmatrix} B & B & C \\ Ç & E & \breve{G} \\ K & E & Y \end{bmatrix} \begin{bmatrix} Z \\ Ç \\ L \end{bmatrix} = \begin{bmatrix} 1 & 1 & x \\ x+1 & x^2+1 & x^3 \\ x^3+x^2+1 & x^2+1 & x^4+x^3+x^2+1 \end{bmatrix} \begin{bmatrix} x^4+x^3+x^2+x \\ x+1 \\ x^3+x^2+x \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ x^4+x^2+x+1 \\ x^4+x^3+x \end{bmatrix} = \begin{bmatrix} B \\ T \\ V \end{bmatrix}$$

Since we have 3 letters left, we can use our 1. Key matrix again.

$$\begin{bmatrix} B & B & C \\ Ç & E & \breve{G} \\ K & E & Y \end{bmatrix} \begin{bmatrix} M \\ W \\ X \end{bmatrix} = \begin{bmatrix} 1 & 1 & x \\ x+1 & x^2+1 & x^3 \\ x^3+x^2+1 & x^2+1 & x^4+x^3+x^2+1 \end{bmatrix} \begin{bmatrix} x^3+x^2+x+1 \\ x^4+x^3+x+1 \\ x^4+x^3+x^2 \end{bmatrix}$$

$$= \begin{bmatrix} x^3+1 \\ x^4+x \\ x^4+x \end{bmatrix} = \begin{bmatrix} H \\ \ddot{O} \\ \ddot{O} \end{bmatrix}$$

It results "$ZÇLMWX$" → "$BTVH\ddot{O}\ddot{O}$".

- **Step 4:** We get

$B + F_1^{(4)} = 1 = B$

$T + F_2^{(4)} = x^4+x^2+x+1 = T$

$V + F_3^{(4)} = x^4+x^3+x+1 = W$

$H + F_4^{(4)} = 1 = B$

$\ddot{O} + F_5^{(4)} = x^4+x^3+x^2 = X$

$\ddot{O} + F_6^{(4)} = x^3+x = I$

where $F_n^{(4)}(x)$ is a Tetranacci polynomial.

It results "$BTVH\ddot{O}\ddot{O}$" → "$BTWBXI$".

Application to the Decoding Algorithm

- **Step 1:** We can get as

$$B + F_1^{(4)} = 1 = B$$

$$T + F_2^{(4)} = x^4 + x^2 + x + 1 = T$$

$$W + F_3^{(4)} = x^4 + x^3 + x = V$$

$$B + F_4^{(4)} = 1 + x^3 = H$$

$$X + F_5^{(4)} = x^4 + x = \ddot{O}$$

$$I + F_6^{(4)} = x^4 + x = \ddot{O}$$

where $F_n^{(4)}(x)$ is a Tetranacci polynomial.

It results "$BTWBXI$" $\rightarrow$ "$BTVH\ddot{O}\ddot{O}$".

- **Step 2:** We multiply the message matrix we just obtained by inverse of the 1. Key matrix.

$$\begin{bmatrix} F & \c{C} & Z \\ S & \breve{G} & N \\ V & T & G \end{bmatrix} \begin{bmatrix} B \\ T \\ V \end{bmatrix} = \begin{bmatrix} Z \\ \c{C} \\ L \end{bmatrix}$$

Since we have 3 letters left, we can use our 1. Inverse key matrix again.

$$\begin{bmatrix} F & \c{C} & Z \\ S & \breve{G} & N \\ V & T & G \end{bmatrix} \begin{bmatrix} H \\ \ddot{O} \\ \ddot{O} \end{bmatrix} = \begin{bmatrix} M \\ W \\ X \end{bmatrix}$$

It results "$BTVH\ddot{O}\ddot{O}$" $\rightarrow$ "$Z\c{C}LMWX$".

- **Step 3:** We can obtain the matrix $\left(Q_k^n(x)\right)^{-1}$ according to the k and n value we have chosen. For $k = 4$ and $n = 6$; we get as

$$\left(Q_4^6(x)\right)^{-1} \begin{bmatrix} Z \\ \c{C} \\ L \\ M \end{bmatrix} = \begin{bmatrix} P \\ U \\ B \\ L \end{bmatrix}$$

and for $k = 2$ and $n = 6$

$$\left(Q_2^6(x)\right)^{-1} \begin{bmatrix} W \\ X \end{bmatrix} = \begin{bmatrix} I \\ C \end{bmatrix}.$$

It results "$Z\c{C}LMWX$" $\rightarrow$ "$PUBLIC$".

3 Conclusion

AES (Advanced Encryption Standard) is a standard offered for encryption of electronic data. The AES cipher is almost identical to the block cipher Rijndael. The Rijndael block and key size vary between 128, 192 and 256 bits. However, the AES standard only calls for a block size of 128 bits. Hence, only Rijndael with a block length of 128 bits is known as the AES algorithm. In the remainder of this page, we only discuss the standard version of Rijndael with a block length of 128 bits.

The Rijndael algorithm perform encryption with the help of polynomials in Galois fields. We have obtained a new encryption algorithm by generalizing the previous studies. In this paper, we generalized the encryption algorithm given in [20] and studied the encryption made with the 2×2 type block matrix operation to the $k \times k$ type in Galois field. We redefined the elements of k-order Fibonacci polynomials sequences using a certain irreducible polynomial in our cryptology algorithm. The algorithm consist of four steps as in the AES-like encryption algorithm. The encryption algorithm defined in this algorithm is a symmetric-key algorithm in which the keys used in both encryption and decryption of encrypted text are related to each other. The encryption and decryption keys are the same like AES. So, this cryptology algorithm is called AES-like cryptology algorithm on the k-order Fibonacci polynomials. In this way, researchers can perform the encryption process based on arbitrary choices.

In this paper, we present the mathematical basis for understanding the design rationale and the features that follow the description itself. Then, we define AES-like encryption by giving the encryption method and its implementation.

Funding Statement: This work is supported by the Scientific Research Project (BAP) 2020FEBE009, Pamukkale University, Denizli, Turkey.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this article.

References

1. Avaroglu, E., Koyuncu, I., Ozer, A. B., Turk, M. (2015). Hybrid pseudo-random number generator for cryptographic systems. *Nonlinear Dynamics, 82(1–2),* 239–248. DOI 10.1007/s11071-015-2152-8.
2. Paar, C., Pelzl, J. (2009). *Understanding cryptography: A textbook for students and practitioners*. London: Springer Science, Business Media.
3. Klima, R. E., Sigmon, N. P. (2012). *Cryptology: Classical and modern with maplets*. New York: Chapman and Hall/CRC.
4. Kizilates, C., Tuglu, N. (2017). A new generalization of convolved (p,q)-Fibonacci and (p,q)-Lucas polynomials. *Journal of Mathematics and Computer Science, 7,* 995–1005. DOI 10.28919/jmcs/3476.
5. Kizilates, C., Du, W. S., Fi, Q. (2022). Several determinantal expressions of generalized tribonacci polynomials and sequences. *Tamkang Journal of Mathematics, 53,* 17–35. DOI 10.5556/j.tkjm.53.2022.3743.
6. Kizilates, C., Cekim, B., Tuglu, N., Kim, T. (2019). New families of three-variable polynomials coupled with well-known polynomials and numbers. *Symmetry, 11(264),* 1–13. DOI 10.3390/sym11020264.
7. Kizilates, C. (2021). New families of Horadam numbers associated with finite operators and their applications. *Mathematical Methods in the Applied Science, 3(4),* 161. DOI 10.1002/mma.7702.
8. Koshy, T. (2001). *Fibonacci and Lucas numbers with applications*. A Wiley-Interscience Publication, John Wiley & Sons, Inc.
9. Gould, H. W. (1981). A history of the Fibonacci Q-matrix and a higher-dimensional problem. *The Fibonacci Quarterly, 19(3),* 250–257. DOI 10.1177/001316448104100337.

10. Hoggat, V. E. (1969). *Fibonacci and Lucas numbers*. Palo Alto: Houghton-Mifflin.
11. Stakhov, A. P. (1999). A generalization of the Fibonacci Q-matrix. *Reports of the National Academy of Sciences of Ukraine, 9,* 46–49.
12. Stakhov, A. P., Mssinggue, V., Sluchenkov, A. (1999). *Introduction into Fibonacci coding and cryptography*. Kharkov: Osnova.
13. Vajda, S. (1989). *Fibonacci and Lucas numbers and the golden section theory and applications*. Lancashire, UK: Ellis Harwood Limitted.
14. Philippou, A. N., Geoughiou, C., Philippou, G. (1983). Fibonacci polynomials of order k, multinomial expansions and probability. *International Journal of Mathematics and Mathematical Sciences, 6(3),* 545–550. DOI 10.1155/S0161171283000496.
15. Qi, F., Kizilates, C., Du, W. S. (2019). A closed formula for the Horadam polynomials in terms of a tridiagonal determinant. *Symmetry, 11(6),* 8. DOI 10.3390/sym11060782.
16. Basu, M., Prasad, B. (2009). The generalized relations among the code elements for Fibonacci coding theory. *Chaos Solitons and Fractals, 41(5),* 2517–2525. DOI 10.1016/j.chaos.2008.09.030.
17. Basu, M., Das, M. (2014). Tribonacci matrices and a new coding theory. *Discrete Mathematics Algorithms and Applications, 6(1),* 1450008. DOI 10.1142/S1793830914500086.
18. Basu, M., Das, M. (2014). Coding theory on Fibonacci n-step numbers. *Discrete Mathematics Algorithms and Applications, 6(2),* 1450017. DOI 10.1142/S1793830914500177.
19. Basu, M., Das, M. (2017). Coding theory on generalized Fibonacci n-step polynomials. *Journal of Information & Optimization Sciences, 38(1),* 83–131. DOI 10.1080/02522667.2016.1160618.
20. Diskaya, O., Avaroglu, E., Menken, H. (2020). The classical AES-like cryptology via the Fibonacci polynomial matrix. *Turkish Journal of Engineering, 4(3),* 123–128. DOI 10.31127/tuje.646926.
21. Mirzaee, F., Hoseini, S. F. (2013). Solving singularly perturbed differential-difference equations arising in science and engineering with Fibonacci polynomials. *Results in Physics, 3(5),* 134–141. DOI 10.1016/j.rinp.2013.08.001.
22. Haq, S., Ali, I. (2021). Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials. *Engineering with Computers, 21,* 366–378. DOI 10.1007/s00366-021-01327-5.