I. J. Computer Network and Information Security, **2023, 4, 48-60**
Published Online on August 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.04.05

Encryption Using Binary Key Sequences in Chaotic Cryptosystem

Vishwas C. G. M.*
J.N.N. College of Engineering / Department of Information Science & Engineering, Shivamogga-577204, Karnataka, India
E-mail: vishwascgm@jnnce.ac.in
ORCID iD: https://orcid.org/0000-0003-2184-6075
*Corresponding author

R. Sanjeev Kunte
J.N.N. College of Engineering / Department of Information Science & Engineering, Shivamogga-577204, Karnataka, India
E-mail: sanjeevkunte@jnnce.ac.in
ORCID iD: https://orcid.org/0000-0001-8424-582X

Varun Yarehalli Chandrappa
Central Queensland University / School of Engineering and Technology, Melbourne-3008, Australia
E-mail: varun.chandrappa@cqumail.com
ORCID iD: https://orcid.org/0000-0002-9542-3297

Received: 25 February 2023; Revised: 04 May 2023; Accepted: 20 June 2023; Published: 08 August 2023

Abstract: Transmission of images on the network is considered insecure which has security-related issues. In this paper, to provide security to digital images, an encryption system that uses four chaotic maps for binary key sequence generation is proposed. The system consists of stages namely, confusion, generation of random binary chaotic key sequence, pseudo-random number generation (*RN*), and diffusion. Keys for encryption are chosen randomly based on a pseudo random generator from the selected chaotic maps by the linear feedback shift register (LFSR). The algorithm achieves good results in terms of NPCR, UACI, and entropy values. The developed cryptosystem resists differential attacks, is sensitive to minor alterations in the keys, and has a large key space.

Index Terms: Linear Feedback Shift Register, Random Binary Key Sequence, Chaotic Image Cryptosystem, Confusion, Diffusion.

1. Introduction

Providing storage and protecting multimedia data in communication has gained much attention. Cryptography, steganography, and watermarking are three efficient ways to provide security for digital information from attacks [1,2]. Among these three ways, cryptography plays an important role in providing security while transmitting digital information on an insecure medium. Considering traditional algorithms such as DES, IDEA, and RSA [3] are appropriate for encrypting text but are inappropriate for images [4,5] mainly because of two reasons. First, the size of the images is larger than the text; second, the traditional encryption systems consume more time to encrypt the images. High redundancy and bulk data capacity are the factors in which conventional algorithms fail to provide security. There exist differences between cryptography and chaotic cryptography. Cryptography uses keys for generating cipher images unlike chaos takes some initial parameters which are considered equal to keys. In providing security for information, implementing efficient encryption algorithms is an important criterion and the key generation process must be unpredictable. Therefore, key sequence generation plays a vital role in any encryption system. Chaotic encryption schemes exhibit certain properties which are appropriate for the encryption of images. Chaos is extremely sensitive to initial conditions, pseudo random, non-periodic, and ergodicity, and accurate in the generation of a large number of chaotic sequences [6]. In this work, a novel hybrid key generator that generates a binary key sequence combining four chaotic maps is proposed. The generated key sequence encrypts and decrypts the digital image in a chaotic cryptosystem.

In this work, the focus is on the application of the generated random binary key sequences for encryption. The novelty of this implementation is in the generation of binary key sequences using all four chaotic maps which are used for encryption. The selection of bits to form the selected random binary key is dependent on the pseudo random generator. Row and column shuffling and at last shuffling of columns randomly minimize the correlation among the pixels. In diffusion, 8 bits of the image are xored with the randomly selected 8 bits of the key from any four chaotic maps to generate the cipher, which increases randomness and makes the system unpredictable.

In section 2, a review of the implementation of chaotic encryption is done. Next, in section 3, we provide the basics of chaotic maps used to develop an encryption cryptosystem. The proposed chaotic system is presented in section 4. The results analysis is done in section 5. Section 6 presents the summary of the work.

2. Literature Review

Recently, chaotic cryptosystems are considered a prime source for the design and implementation of pseudorandom generators. Many papers exist in the literature.

Ali Soleymani et.al., [7] use two maps, the Arnold Cat Map (ACM) and Henon chaotic map, and proposed a novel encryption scheme to secure images. Initially, ACM is used for bit and pixel-level permutations on plain and secret images respectively. Then, the Henon map is applied to the secret image and the specific parameters for the permutation. In the first step, a secret image is generated, and the pixel of the secret image is permuted. Simultaneously, the bit-level permutation of the plain image is carried out for r rounds. Pixel modification is carried out on the two obtained images and xor of consecutive pixels is done. The obtained result of this stage is given as feedback to the bit permutation function to carry out $(p-1)$ rounds. The algorithm resists statistical as well as the chosen plaintext attack and attains large key space.

Asia Mahdi Naser Alzubaidi [8] proposed an efficient algorithm in which iterative scrambling of image pixels is carried out by dividing it into 64 blocks. Then, 2D ACM is applied to distort the relationship among adjacent pixels of the input. Encryption is performed using a Henon map which diffuses the correlation between the input and the cipher. Performance analysis shows that the system is resistant to both statistical and differential attacks.

Guodong Ye and Xiaoling Huang [9] proposed an encryption algorithm that performs permutation and diffusion in a single round. A mathematical model is used in the permutation stage. The SHA-3 algorithm is used in the generation of the keystream in diffusion. This keystream is combined with newly evaluated keys. The generated keystream has a dependency on the binary input. The system has high security and resists known plaintext and chosen plaintext attacks.

Yueping Li et.al., [10] proposed an algorithm using both pixel-level and bit-level permutation. The chaotic system generates chaotic sequences which are related to the elements of the input. Permutation at the pixel level is applied which shuffles the input. Next, a bit-level permutation is applied which makes the system secure. Finally, diffusion generates the cipher. This scheme is secure and considered reliable.

Dasari Sravanthi et.al., [11] proposed an algorithm that performs a bit-plane operation using Piece-wise Linear Chaotic Map (PWLCM) and 2-D Logistic-adjusted-Sine map. Initially, bit plane diffusion is carried out by applying the PWLCM. Next, the 2-D Logistic-adjusted-Sine map is applied in row-shuffling and column-shuffling. The SHA-256 generates the secret keys which resist known-plaintext attack and chosen-plaintext attack. The bit-plane operation performs confusion and diffusion of the pixels simultaneously. This scheme resists common attacks.

Sameh S. Askar et.al., [12] proposed an encryption method that initially alters the pixel positions. Row shuffling followed by column shuffling operations minimizes the correlation between the pixels. The algorithm has a large key space and good information entropy. Also, the algorithm resists differential attacks.

Yingchun Hu et.al., [13] used a 1D chaotic map for a bit-level encryption algorithm and worked out its security analysis. The proposed algorithm handles bit-level permutation encryption in an efficient way. The generated chaotic sequence is not dependent on the input. During decryption, the diffusion and the permutation keys are retrieved by the input and the cipher. This implementation can be a framework for performing security analysis of the algorithm which uses bit-level operations.

Arwa Benlashram et.al., [14] used a 3D logistic map to design an encryption system. Initially, the pixels of the input image are shuffled, which is then xored with a key. Then, a 3D chaotic logistic map is applied to generate the cipher. The security of this scheme is enhanced by using additional (three) steps to generate the cipher. The algorithm attains large key space by using the 3D logistic map.

Moysis L et.al., [15] applied the shuffling method to bits of an image by placing these bits in a 3D matrix. Then, a three-step method of shuffling is carried out on each row, column, and at the bit level of the matrix. Combining both the shuffling operation and XOR results in a cipher that resists attacks like histogram and cropping attacks. The algorithm is also resistant to transmission noise.

Devipriya M and Brindha M [16] proposed a method for image diffusion using a Tent map and block encryption. Pixel permutation is applied at the block level which resists statistical attacks. Bit-level permutation using a PLWCM is performed to resist differential attacks. The algorithm resists statistical and differential attacks effectively.

From the survey carried out, it is observed that most work prevails on using a single chaotic map in the cryptosystem used for encryption. Combining more than one chaotic map enhances the key space and makes the cryptosystem strong and secure.

3. Concepts of Chaotic Maps

In the proposed work, four different chaotic maps are considered. The properties of these maps are presented in this section.

3.1. Henon Map

The Henon map introduced by Michel Henon is a commonly used example of a discrete-time dynamical system [18]. It is a system that behaves chaotically. It considers a point (x_n, y_n) in the plane and maps it to a new point which can be represented by equation (1).

$$\begin{gathered} x_n + 1 = y_n + 1 - a * x_n * x_n{}' \\ y_n + 1 = b * x_n \end{gathered} \tag{1}$$

The map is dependent on the parameters, *a* and *b*. For a classical Henon map, *a* and *b* are initialized with the values 1.4 and 0.3 respectively and is chaotic.

3.2. Arnold's Cat Map

ACM [18] was discovered by Vladimir Arnold is also a commonly used example of a discrete system that behaves chaotically. ACM can be represented by equation (2).

$$\begin{gathered} x_n + 1 = (2 * x_n + y_n)\, mod\ 1 \\ y_n + 1 = (x_n + y_n)\, mod\ 1 \end{gathered} \tag{2}$$

The chaotic cryptosystem also uses the Tent map and a Logistic map. More information can be found in [17,18].

4. Proposed Chaotic Cryptosystem

The proposed encryption system (Fig.1) employs stages namely, confusion, generation of chaotic random binary key sequences using Tent, Logistic, Henon, and Arnold chaotic maps, and diffusion. The pseudo random number, *RN* provides the selection of a random key from the selected chaotic map (determined by the LFSR) to encrypt each pixel to generate the cipher. A detailed explanation of the components is as follows.

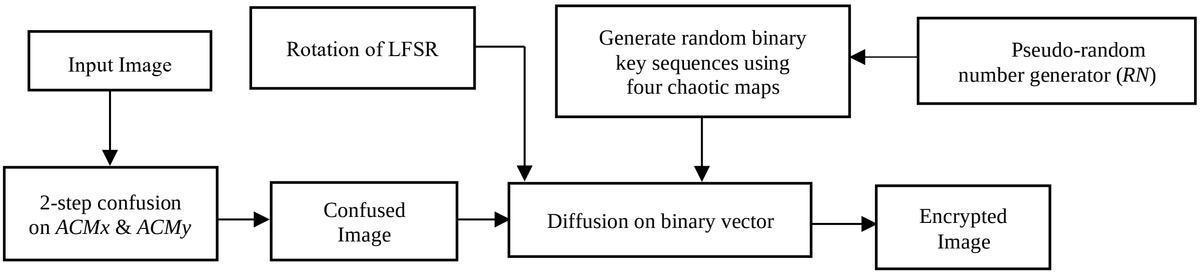

Input Image
Rotation of LFSR
Generate random binary key sequences using four chaotic maps
Pseudo-random number generator (*RN*)
2-step confusion on *ACMx* & *ACMy*
Confused Image
Diffusion on binary vector
Encrypted Image

Fig.1. Chaotic Cryptosystem based on random binary key sequences

4.1. Confusion

The RGB to the grey converted input image is given as input to the confusion process. Initially, using equation (2), chaotic ACM *x* and *y* sequences are generated. The pixel's positions of the image are confused in two steps. Initially, the pixels in the input image are altered based on the generated ACM *x* sequence and shuffled image in the first step, Shfld_Img1 is obtained. The first step of confusion can be represented as

$$Shfld\_Img1 = Confuse(input, ACMx) \tag{3}$$

The second step is to confuse the obtained image of the first step, Shfld_Img1 on the ACM *y* sequence to obtain *Shfld_Img2* by equation (4).

$$Shfld\_Img2 = Confuse(Shfld\_Img1\,, ACMy) \tag{4}$$

The two-step confusion method employed is represented in fig. 2. The obtained image after two steps of shuffling of pixel position is given as input to the diffusion process.

4.2. Generation of Random Binary Key Sequences

Initially, the system generates four chaotic sequences viz., using Tent, Logistic, Henon, and ACM. The logistic map is defined as a polynomial mapping of degree 2 which exhibits chaotic behavior. This map arises from a non-linear dynamical equation [20]. The generation of a chaotic logistic sequence is done as in equation (5).

$$x(i+1) = r * x(i) * (1 - x(i)) \tag{5}$$

$$x = mod(unit8\big(round(x * power(10,8), 256)\big) \tag{6}$$

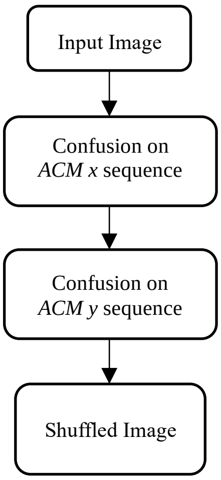


Fig.2. Two-step confusion on the position of ACM *x* and *y* sequence

The generated values are in the decimal value of range [0-255] as shown in equation (6). Covert the generated logistic sequence $x \in \{ 0, 1, \ldots, 255 \}$ into bit/binary vector $x \in \{ 0, 1, \ldots, 255 \}$ expressed as $Logistic\_Seq \in \{ 0, 1, \ldots, p*q*8 \}$, where p is the width and q is the height of the input. The chaotic Tent sequence also follows the same process of generation. Similarly, the ACM *x* sequence is generated using equation (7).

$$x(i) = 1 - a\ (x(i) * \big(x(i)\big) + y(i-1) \tag{7}$$

$$x = mod(unit8\big((x * power(7,6), 256)\big) \tag{8}$$

Similarly, the ACM *y* sequence is generated using equation (9).

$$y(i) = b * x(i-1) \tag{9}$$

$$y = mod(unit8\big(round(y * power(7,6), 256)\big) \tag{10}$$

A similar method is also used to generate the *x* and *y* sequences of the Henon map. An 8-bit LFSR [19] along with primitive polynomials of degrees 5, 6, 7, or 8 are considered for the generation of key sequences. Out of the four chaotic maps, the LFSR chooses one chaotic map for encryption. The selection of a particular chaotic map is done based on the 7th & 8th bit, the two most significant bits (MSB) of the LFSR. After selecting a chaotic map, a random key from this map is selected based on the pseudo random number *RN* for encryption of one pixel from the shuffled image. The same method is repeated till the shuffled image is encrypted. Table 1 shows the 8 individual binary key sequences which are generated along with their notation. Also, a combination of generated binary sequences from table 1 is done which generates 61 unique combinations of binary key sequences. ***Hxy_bin*** is the combined binary key sequence that is generated by combining keys from both the Henon *x*, ***Hx_bin***, and Henon *y*, ***Hy_bin*** binary key sequences. Similarly, the binary key sequence ***ACMxy_bin*** for ACM is generated.

Table 1. Generated random binary key sequences

Chaotic map	Random binary key sequences generated (notation)
Logistic	***L_bin***
Tent	***T_bin***
Henon	***Hx_bin, Hy_bin, Hxy_bin***
Arnold Cat Map	***ACMx_ bin, ACMy_ bin, ACMxy_bin***

A total of 8 individual and 53 random binary key sequences are formed by combining two or more chaotic binary sequences that can be generated to be used for encryption.

4.3. Diffusion

The diffusion process is illustrated in fig.3. The shuffled image is expressed as a binary vector, ***binary***. The binary is converted to a vector array of bits, ***RSh_binary*** of dimension *p***q**8 and arranged as ***RSh_binary_matrix*** of dimension *p**(*q**8). Inter-shuffling of rows and columns if adopted. Initially, all the even rows of ***RSh_binary_matrix*** are arranged first followed by odd rows. Then arrange all odd rows first followed by even rows. The last step is to exchange all the columns randomly based on the generated pseudo random number, *RN*. The diffusion process is described in Algorithm 1, ***Diffusion.***

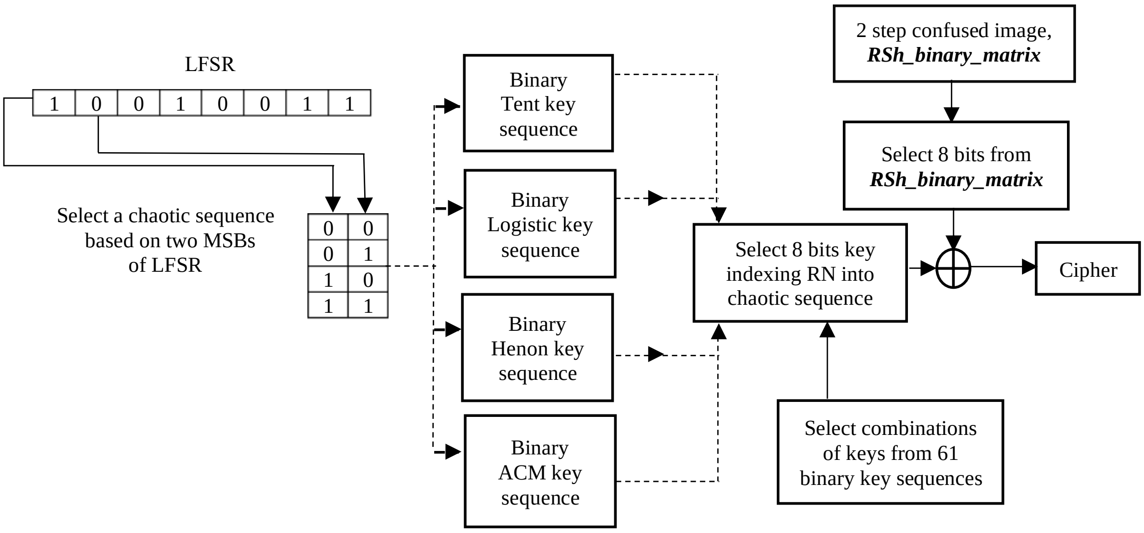

LFSR
1 0 0 1 0 0 1 1
Select a chaotic sequence based on two MSBs of LFSR
0 0
0 1
1 0
1 1
Binary Tent key sequence
Binary Logistic key sequence
Binary Henon key sequence
Binary ACM key sequence
2 step confused image, RSh_binary_matrix
Select 8 bits from RSh_binary_matrix
Select 8 bits key indexing RN into chaotic sequence
Cipher
Select combinations of keys from 61 binary key sequences

Fig.3. Diffusion: Two MSBs select a particular binary chaotic map

Algorithm 1: ***Diffusion***

```
Step 1: Read the shuffled image of dimension p*q
Step 2: Convert the shuffled image of the confusion process to binary vector, binary
Step 3: Reshape binary and convert to vector array of bits, RSh_binary of dimension p*q*8
Step 4: Arrange RSh_binary as a binary matrix, RSh_binary_matrix of dimension p*(q*8)
Step 5: for i = 1 to p do
Step 6:         arrange even rows first followed by odd rows
        end for
Step 7: for j = 1 to q do
Step 8:         arrange odd rows first followed by even rows
        end for
Step 9: Exchange q columns of RSh_binary_matrix based on the sequence of pseudo random number generator
Step 10: for j = 1 to p*q do
                fetch 8-bits from RSh_binary to encrypt with an 8-bit key
                rotate LFSR
                generate a pseudo random number, RN
               call Chaotic_Choice_Enc
          end for
```

The steps to exchange all the columns randomly (Step 9) of ***RSh_binary_matrix*** based on a pseudo random number generator is presented in Algorithm 2, ***Exchange_q_col.***

Algorithm 2: ***Exchange_q_col***

```
Step 1: Generate 2 pseudo random numbers RN1 and RN2
Step 2: for r = 1 : q*8
            RSh_binary_matrix(:,RN1) =    RSh_binary_matrix(:,RN2)
        end for
```

The encryption and the generation of the cipher are done in the ***Chaotic_Choice_Enc*** (Step 10) as shown in Algorithm 3.

Algorithm 3: ***Chaotic_Choice_Enc***

```
Step 1: Fetch the 7th & 8th bits, the two MSB from LFSR
Step 2: Select binary Logistic, Tent, Henon, or ACM if MSB = (0,0), (0,1), (1,0), or (1,1) correspondingly
Step 3: Index to the location RN to fetch 8 bit key from the binary key sequence selected
Step 4: xor 8-bits fetched from RSh_binary and 8 bit key to generate 8 bit cipher
```

Algorithm 3 is repeated to encrypt all bits in the shuffled image. Intra-shuffling carried out on the bits reduces the correlation value between the adjacent pixels of the shuffled image. The resultant binary matrix after a random exchange of columns is again expressed as binary vector ***RSh_binary_ip***.

A. Encryption of Binary Vector, RSh_binary_ip

In the diffusion process, 8 bit keys are selected based on the pseudo random number, *RN*. The generated *RN* is indexed as the starting location from which 8 bits of a key are to be fetched from the selected chaotic sequence based on MSBs of the LFSR. These 8 bits fetched from the selected binary vector of chaotic maps act as a random key to encrypt 8 bits of the shuffled image selected from the binary vector ***RSh_binary_ip***. Algorithm 4 for fetching a random 8-bit key from the logistic sequence is described as follows.

Algorithm 4: ***Logistic_Seq_fetch8bits***

```
Step 1: L_bin(RN)
Step 2: for i = 1 : 8
            L_bin_8bits(i) = L_bin(RN+i)
        end for
Step 3: randomkey = L_bin_8bits
Step 4: Cipher8bits = bitxor(Bits8,randomkey)
```

The 8 bits of the random key are then bitxored with 8 bits from ***RSh_binary_ip*** and 8 bits cipher is obtained. The generated 8 bits of cipher shown in step 4 of algorithm 4 are represented in the range [0-255]. Once a single pixel from the ***RSh_binary_ip*** is encrypted, the LFSR is rotated and the two new MSBs of the LFSR are considered to decide which new chaotic sequence to select next for encryption. The above steps are repeated to process all the bits in the ***RSh_binary_ip*** to generate the cipher image, Cipher. Fig.4 displays the selection of a particular binary key sequence for encrypting bits from the ***RSh_binary_matrix*** to generate the cipher image.

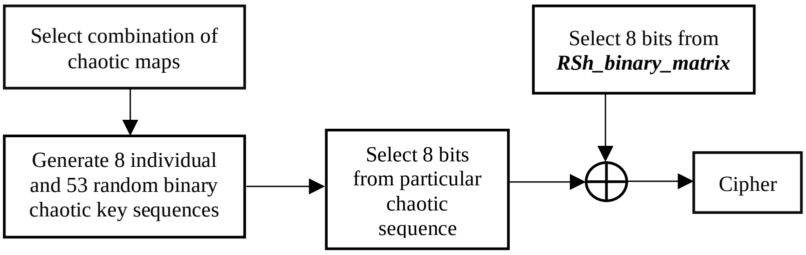

Select combination of chaotic maps
Generate 8 individual and 53 random binary chaotic key sequences
Select 8 bits from particular chaotic sequence
Select 8 bits from *RSh_binary_matrix*
Cipher

Fig.4. Selecting a particular chaotic key sequence in diffusion

B. Estimations of the Cryptographic Strength of the Proposed Algorithm

The binary key sequences are generated based on the chaotic maps. Any minor changes in the parameters result in changes in the random binary key sequence. As the encryption is carried out using the generated binary key sequences, any minor changes in the key will not retrieve the original image.

5. Experimental Results

5.1. Key Sequences

In this implementation, five primitive polynomials of degrees 5, 6, 7, and 8 are considered for the generation of random binary key sequences. Each polynomial is used to generate 61 possible key sequences from 8 binary key sequences presented in table 1. Table 2 illustrates the five primitive polynomials considered for implementation. Therefore, considering all five primitive polynomials, a total of 305 unique binary key sequences can be generated.

Parameters of Binary Key Sequence Generator

For the generation of the binary Tent sequence, the initial value of x, $x(1)$=0.2, and μ=1.9. For Logistic binary sequence generation, $x(1)$=0.2 and r=3.85566. For the binary Henon sequence, the values of a and b are 1.4 and 0.3 respectively and the initial values of x and y are $x(1)$=20 and $y(1)$=200. Similarly, for ACM sequence, $x(1)$=0.2 and *y(1)*=0.3. For the pseudo random number generator, the seed value is initialized to 2.

5.2. Input and Cipher Image

To evaluate the chaotic cryptosystem, Lena, Peppers, Flowers, and Barbara are considered. The results are presented in fig.5 for a primitive polynomial of degree 6, $x^6+x^5+x^3+x^2+1$. The input to the encryption system is the Lena image. The cipher image depicts that the original image cannot be retrieved. Using the appropriate secret keys, the decrypted image (fig.5(c)) is retrieved which is the same as the input Lena. The Lena image is retrieved without any loss.

Table 2. Considered primitive polynomials

Primitive Polynomials
$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$
$x^8 + x^6 + x^5 + x^4 + 1$
$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$
$x^6 + x^5 + x^3 + x^2 + 1$
$x^5 + x^4 + x^3 + x^2 + 1$

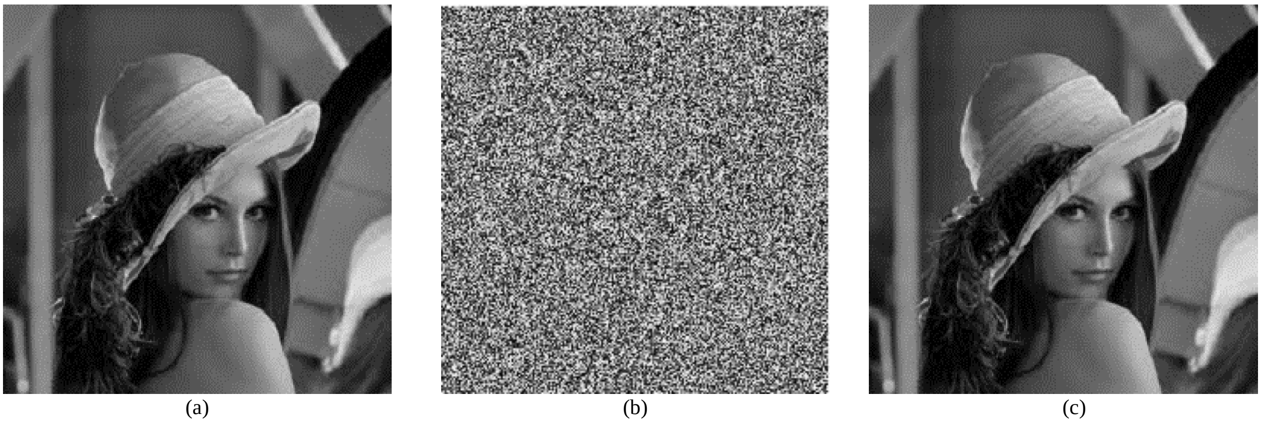

(a) (b) (c)

Fig.5. (a) Input Lena (b) Cipher (c) Decrypted

5.3. Mean Squared Error and Peak-signal-to-noise Ratio (MSE & PSNR)

MSE is the mean squared error. PSNR can be easily defined using MSE. MSE and PSNR are given by equations (11) and (12).

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{11}$$

PSNR is given by equation (12).

$$PSNR = 10\ log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{12}$$

Here, MAX_I is taken as the maximum possible pixel value of the image. From table 3, it is observed that all the cipher obtained for the four images considered with the listed primitive polynomial and specific map combinations have high MSE and low PSNR values.

Table 3. MSE & PSNR

	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ ***L_bin,Hx_bin,ACMx_bin***		$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ ***T_bin,L_bin,Hxy_bin***		$x^6+x^5+x^3+x^2+1$ ***T_bin,L_bin,Hxy_bin,ACMy_bin***	
Images	MSE	PSNR(db)	MSE	PSNR(db)	MSE	PSNR(db)
Lena	74.2142	29.4599	75.1593	29.4049	74.9137	29.4191
Peppers	108.6285	27.8053	110.007	27.7505	109.1643	27.7839
Flowers	87.5138	28.7440	88.2537	28.7074	87.7060	28.7345
Barbara	99.8490	28.1713	102.422	28.0606	99.7132	28.1772

5.4. *Differential Analysis*

The number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) [24] are two significant measures for displaying the strength of any cryptosystem or ciphers with regard to differential attacks. Normally, obtaining high values of NPCR and UACI (equations (13) and (14)) by any encryption algorithm shows effective resistance to differential attacks. Also, the system must exhibit sensitivity to minimal alterations in the input thereby resisting differential attacks. Even a change of one bit in the input must show drastic differences in the cipher.

$$NPCR = \frac{1}{W\,X\,H}\sum_{ij} D(i,j)\,X\,100\% \tag{13}$$

$$UACI = \frac{1}{W\,X\,H}\left[\sum_{ij}\frac{C_1(i,j) - C_2(i,j)}{255}\right]X\,100\% \tag{14}$$

where W and H are the sizes of the input. C_1 and C_2 2 are the ciphers in which the input image differs by one pixel. *D(i,j)* is given by the equation (15).

$$D(i,j) = \begin{cases} 0 & C_1(i,j) - C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{15}$$

The NPCR and UACI are computed for the considered standard images with three chaotic map combinations of binary key sequences generated by the primitive polynomials. The obtained values for three primitive polynomials considering specific random binary key sequences are listed in table 4. The obtained values are 99% and 33% respectively. The obtained values are ideal, and the cryptosystem resists differential attacks. A comparison of the implementation (with primitive polynomial $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and random binary key sequence, ***T_bin,L_bin,Hxy_bin*** with the reference papers 23, 24, 25, and 26 is carried out and it reveals that values for the current implementation are better.

Table 4. NPCR & UACI

	Primitive polynomial & Key Sequences								
	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ ***L_bin,Hx_bin,ACMx_bin***		$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ ***T_bin,L_bin,Hxy_bin***		$x^6+x^5+ x^3+x^2 +1$ ***T_bin,L_bin,Hxy_bin, ACMy_bin***		Comparison of Primitive Polynomial: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and Key Sequence: ***T_bin,L_bin,Hxy_bin*** with reference papers		
Images	NPCR	UACI	NPCR	UACI	NPCR	UACI		NPCR	UACI
Lena	99.53%	32.01%	99.72%	32.40%	99.60%	32.10%	Chai et.al [20]	99.59%	33.42%
Peppers	99.60%	33.46%	99.64%	33.46%	99.55%	33.35%	Chai et.al [21]	99.60%	33.48%
Flowers	99.62%	35.28%	99.66%	35.26%	99.57%	35.31%	Yue et.al [22]	99.60%	33.45%
Barbara	99.61%	33.46%	99.64%	33.46%	99.66%	33.46%	Sun et.al [23]	99.61%	33.46%

5.5. *Information Entropy*

Information entropy measures randomness. If m is the source of information, the entropy is evaluated using equation (16).

$$H(m) = \sum_{i=0}^{M-1} p(m_i)\,log\,\frac{1}{p(m_i)} \tag{16}$$

where M is the total number of symbols and $m_i \in m$; $p(m_i)$ is the probability of symbols. The source of the information sends 256 symbols, and the theoretical value H(m) is equal to 8. The value obtained should be closer to 8 indicating that the possibility of decoding the cipher by attackers is minimized. From table 5, the computed entropy values are closer to the ideal value of 8.

5.6. *Security Analysis*

A. *Key Space Analysis*

In total, 61 random binary key sequences have a key space of 2^{108}. This key space is sufficient to resist attack.

B. *Key Sensitivity*

A change made to the key in terms of 1 bit should generate a different output in either the cipher or in the decrypted image. To ensure the security aspect, any efficient algorithm must exhibit sensitivity to the keys. Wrong decryption of the output images occurs when the keys when decryption is performed using different keys as compared to the correct keys used during encryption. The key used to encrypt the original Lena is the binary key generated using binary Tent, ***T_bin***, binary Logistic ***L_bin,*** and binary Henon *y* sequence, ***Hy_bin*** for the considered primitive polynomial $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$. The cipher of the input Lena is in fig.6 (b). In decryption, for the generation of binary Logistic sequence, ***L_bin,*** the original value of x(1)= 0.2 is changed to x(1)= 0.2000000001 and for the Tent sequence generation, ***T_bin***, if the initial value is taken as x(1)=0.2000000001 and in Henon *y* sequence generation, ***Hy_bin***, the value of *a* is changed from *a*=1.4 to *a*=1.4000000001 and initial values are *x(1)*=20 and *y(1)*=255 are taken, decrypts as Fig.6 (c) which is the wrong decryption. In this case, the original Lena image is not retrieved due to minimal alterations in the key. Fig.6 (d) is the obtained image after decryption using the correct keys. Therefore, the keys are sensitive and pass the key sensitivity test.

Table 5. Entropy for standard images for three polynomials and binary key sequences

	Primitive polynomial and Key Sequences					
	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ ***L_bin,Hx_bin,ACMx_bin***		$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ ***T_bin,L_bin,Hxy_bin***		$x^6+x^5+ x^3+x^2 +1$ ***T_bin,L_bin,Hxy_bin,ACMy_bin***	
Images	Input	Cipher	Input	Cipher	Input	Cipher
Lena	7.4948	7.9827	7.4948	7.9827	7.4948	7.9806
Peppers	7.5924	7.9774	7.5924	7.9811	7.5924	7.9820
Flowers	7.1945	7.9796	7.1945	7.9808	7.1945	7.9815
Barbara	7.4291	7.9796	7.4291	7.9794	7.4291	7.9786

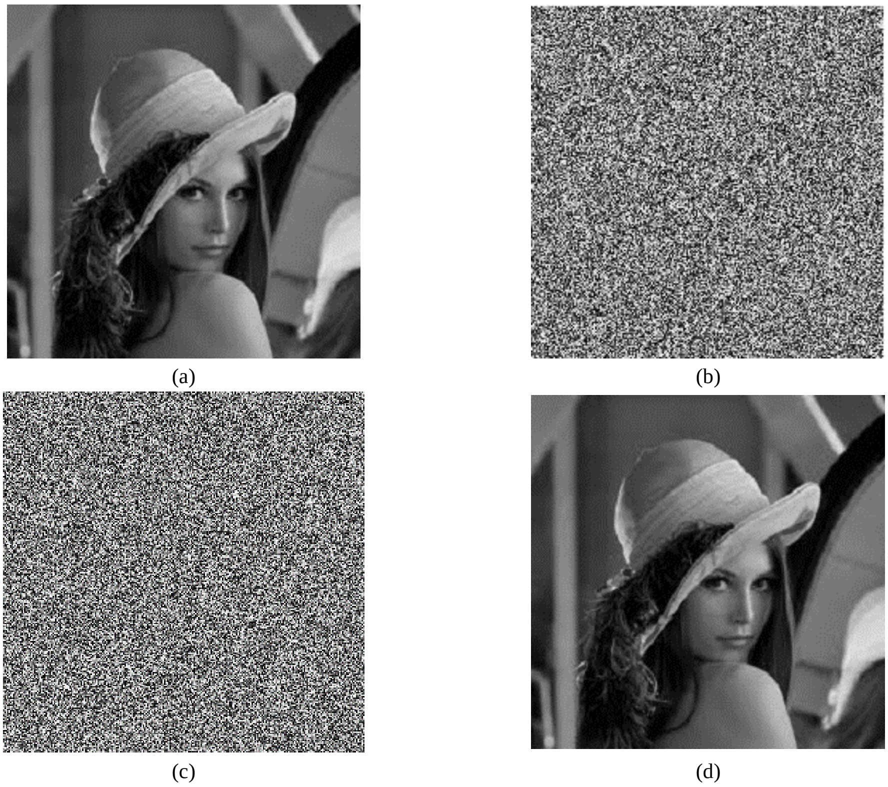

(a) (b) (c) (d)

Fig.6. (a) Input Lena image, (b) Cipher Lena image, (c) Decrypted Lena image with wrong key, (d) Decrypted Lena image with the correct key

5.7. Correlation Coefficients

The adjacent pixels in the input image are highly correlated. Any efficient encryption system should minimize the pixel correlation coefficients in the cipher image and resist statistical attacks. The inter-shuffling method adopted in the diffusion process results in a reduced correlation significantly in the binary matrix, ***RSh_binary_matrix***. Table 6 lists the evaluated correlation values for Lena, Peppers, Flowers, and Barbara images in a horizontal, vertical, and diagonal direction. To evaluate the correlation between the input and cipher images, 5000 pixels were selected randomly in each direction. The correlation values of four images of input and cipher with different combinations of the key sequence under considered primitive polynomials are listed. It can be observed that for the input images in all three directions, the pixels are highly correlated with their neighbor pixels, and the value is closer to 1. Whereas, in the cipher image, the correlation is less and is closer to 0.

Also, the correlation coefficient result of the implementation from table 7 for the primitive polynomial $x^6 + x^5 + x^3$

+ x^2 + 1 with key sequence ***T_bin,L_bin,Hxy_bin,ACMy_bin*** is compared with reference papers 27, 28, 29, and 30. The obtained values in our implementation are better. Table 7 compares the obtained values of correlation for the primitive polynomial of degree 6 (III, table 6) with key sequence, ***T_bin,L_bin,Hxy_bin,ACMy_bin,*** and the correlation values of other references. The obtained correlation is better as compared to the references.

Table 6. Correlation of four images and corresponding cipher in three directions

1. Primitive Polynomial: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ and Key Sequence: ***L_bin,Hxy_bin,ACMy_bin***								
	Lena		Peppers		Flowers		Barbara	
	Input	Cipher	Input	Cipher	Input	Cipher	Input	Cipher
Horizontal	0.8910	-0.0025	0.8602	0.0086	0.9929	0.0048	0.8265	0.0081
Vertical	0.8562	0.0100	0.7877	-0.0063	0.9884	-0.0155	0.7466	0.0049
Diagonal	0.8894	-0.0070	0.8230	0.0072	0.9875	-0.0016	0.7891	0.0037
II. Primitive Polynomial: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and Key Sequence: ***T_bin,L_bin,Hxy_bin***								
Horizontal	0.8910	-0.0111	0.8602	0.0065	0.9929	-0.0053	0.8265	-0.0317
Vertical	0.8562	0.0016	0.7877	0.0184	0.9884	-0.0028	0.7466	0.0047
Diagonal	0.8894	-0.0265	0.8230	-0.0060	0.9875	0.0158	0.7891	0.0299
III. Primitive Polynomial: $x^6 + x^5 + x^3 + x^2 + 1$ and Key Sequence: ***T_bin,L_bin,Hxy_bin,ACMy_bin***								
Horizontal	0.8910	-0.0208	0.8602	0.0108	0.9929	0.0008	0.8265	0.0229
Vertical	0.8562	0.0002	0.7877	0.0065	0.9887	-0.0029	0.7466	-0.0068
Diagonal	0.8894	-0.0137	0.8230	-0.0203	0.9876	-0.0024	0.7891	-0.0021
IV. Primitive Polynomial: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ and Key Sequence: ***L_bin,Hxy_bin,ACMy_bin***								
Horizontal	0.8910	-0.0025	0.8602	0.0086	0.9929	0.0048	0.8265	0.0081
Vertical	0.8562	0.0100	0.7877	-0.0063	0.9884	-0.0155	0.7466	0.0049
Diagonal	0.8894	-0.0070	0.8230	0.0072	0.9875	-0.0016	0.7891	0.0037
V. Primitive Polynomial: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and Key Sequence: ***T_bin,L_bin,Hxy_bin***								
Horizontal	0.8910	-0.0111	0.8602	0.0065	0.9929	-0.0053	0.8265	-0.0317
Vertical	0.8562	0.0016	0.7877	0.0184	0.9884	-0.0028	0.7466	0.0047
Diagonal	0.8894	-0.0265	0.8230	-0.0060	0.9875	0.0158	0.7891	0.0299
VI. Primitive Polynomial: $x^6 + x^5 + x^3 + x^2 + 1$ and Key Sequence: ***T_bin,L_bin,Hxy_bin,ACMy_bin***								
Horizontal	0.8910	-0.0208	0.8602	0.0108	0.9929	0.0008	0.8265	0.0229
Vertical	0.8562	0.0002	0.7877	0.0065	0.9887	-0.0029	0.7466	-0.0068
Diagonal	0.8894	-0.0137	0.8230	-0.0203	0.9876	-0.0024	0.7891	-0.0021

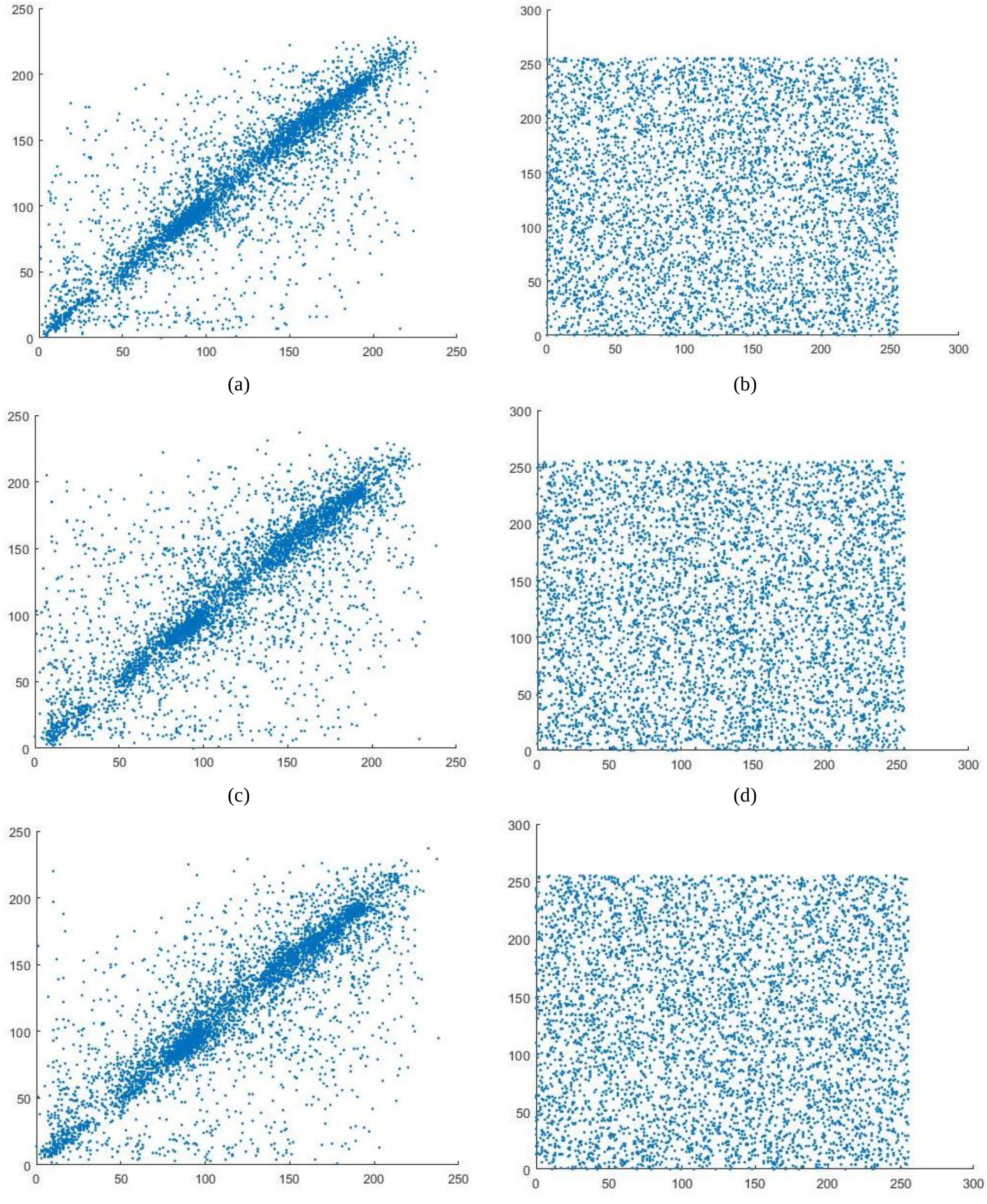
Table 7. Comparison of correlation coefficients of Lena image with Primitive Polynomial: $x^6 + x^5 + x^3 + x^2 + 1$ and Key Sequence: ***T_bin, L_bin, Hxy_bin, ACMy_bin***

Correlation Coefficients of Lena Cipher image			
	Horizontal	Vertical	Diagonal
Proposed III	-0.0208	0.0002	-0.0137
Gao et.al [24]	0.0142	0.0007	0.0183
Teng et.al [25]	0.0242	0.0194	0.0024
Chen et.al [26]	0.0021	0.0046	0.0033
Wang et.al [27]	0.0056	0.0065	0.0073

Fig.7 displays the correlation plot for the input Peppers considered in all three directions. It is observed that the pixels in the input in fig.7 (a), (c), and (e) are highly correlated whereas the pixels in cipher in fig.7 (b), (d), and (f) are scattered, indicating a reduced correlation considerably.

6. Conclusions

This paper focuses on applying chaotic maps to generate random binary key sequences which are used to encrypt four standard images. The rearrangement of bits of rows and random shuffling of columns in the binary matrix disturbs the correlation among neighboring pixels. This method uses an intra-shuffling scheme at the bit level to minimize the correlation values among adjacent pixels. Differential analysis reveals that the obtained NPCR values are nearer to 99% and the UACI values are at 33% which is the ideal score resisting differential attack. The obtained entropy values using the random binary key sequences generated using the primitive polynomial of three different degrees are evaluated and found to be nearer to the ideal value of 8. The algorithm also exhibits sensitivity to the secret key. The algorithm has a large key space of 2^{108}. The correlation values are evaluated using six binary key sequences for polynomials of all the degrees considered. From the evaluated values, adjacent pixels in the input are nearer to 1 and the cipher values are nearer to 0, which shows that the chaotic encryption algorithm effectively resists statistical attack. Here, the binary key sequences generated are chaotic. Any minor changes in the parameters result in changes in the key sequence. As the cryptosystem is chaotic based, these minor changes are reflected during decryption in the cryptosystem. Also, the series of operations applied in the diffusion process makes cryptanalysis difficult for any attacker. The obtained results accomplish that the cryptosystem is secure. Therefore, the cryptosystem can be effectively used to encrypt digital images, securely transfer information on the Internet, and can be used in social media storage.



(a)

(b)

(c)

(d)

(e)

(f)

Fig.7. Correlation Coefficient in three directions for input image (Peppers) and the cipher. (a) Input-Horizontal (b) Cipher-Horizontal (c) Input-Vertical (d) Cipher-Vertical (e) Input-Diagonal (f) Cipher-Diagonal

References

[1] Fu C, Meng W-h, Zhan Y-f et al., “An efficient and secure medical image protection scheme based on chaotic maps,” Computers in Biology and Medicine, Vol 43(8), pp.1000–1010, 2013.

[2] Khanzadi H, Eshghi M, Borujeni SE, “Image encryption using random bit sequence based on chaotic Maps,” Arabian Journal for Science and Engineering”, Vol 39, pp. 1039–1047, 2014.

[3] W. Stallings, “Cryptography and Network Security: Principles and Practice,” Prentice-Hall, New Jersey, 1999.

[4] K. Gupta, R. Gupta, R. Agrawal, and S. Khan, "An Ethical Approach of Block Based Image Encryption Using Chaotic Map," International Journal of Security and Its Applications, vol.9, no.9, pp.105-122, 2015.

[5] W. Wang, H. Tan, Y. Pang, Z. Li, P. Ran, and J. Wu," A Novel Encryption Algorithm Based on DWT and Multichaos Mapping," Hindawi Publishing Corporation, Journal of Sensors Volume, pp. 105-121, 2016.

[6] Yu F, Li L, Tang Q, Cai S , Song Y , Xu Q, " A Survey on True Random Number Generators Based on Chaos, Discrete Dynamics in Nature and Society," pp. 1-10, 2019.

[7] Soleymani, A., Nordin, M. J., & Sundararajan, E., "A Chaotic Cryptosystem for Images Based on Henon and Arnold Cat Map," The Scientific World Journal, Vol 2014, pp. 1–21, 2014.

[8] Asia Mahdi Naser Alzubaidi, "Color Image Encryption and Decryption using Pixel Shuffling with Henon Chaotic System", International Journal of Engineering Research & Technology, Vol. 3, pp. 1414-1418, 2014.

[9] Ye, G., & Huang, X., "A feedback chaotic image encryption scheme based on both bit-level and pixel-level," Journal of Vibration and Control, Vol 22(5), pp. 1171–1180, 2015.

[10] Li, Y., Wang, C., & Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Optics and Lasers in Engineering, Vol 90, pp. 238–246, 2017.

[11] Sravanthi, D., Abhimanyu Kumar Patro, K., Acharya, B., & Majumder, S., "A Secure Chaotic Image Encryption Based on Bit-Plane Operation. Advances in Intelligent Systems and Computing," pp. 717–726, 2018.

[12] Askar, S. S., Karawia, A. A., & Alammar, F. S, "Cryptographic algorithm based on pixel shuffling and dynamical chaotic economic map," IET Image Processing, 12(1), 158–167, 2018.

[13] Hu, Yu, S., & Zhang, Z., "On the Cryptanalysis of a Bit-Level Image Chaotic Encryption Algorithm. Mathematical Problems in Engineering, Vol 2020, pp. 1–15, 2020.

[14] Benlashram, A., Al-Ghamdi, M., Al Talhi, R., & Kaouther Laabidi, P, "A novel approach of image encryption using pixel shuffling and 3D chaotic map," Journal of Physics: Conference Series, 1447, 012009, pp. 1-10, 2020.

[15] Moysis, L., Kafetzis, I., Tutueva, A., Butusov, D., Volos, C, "Chaos-Based Image Encryption Based on Bit Level Cubic Shuffling," Abd El-Latif, A.A., Volos, C. (eds) Cybersecurity. Studies in Big Data, Vol 102, 2022.

[16] Devipriya M., Brindha M, "Image encryption using modified perfect shuffle-based bit level permutation and learning with errors based diffusion for IoT devices," Computers and Electrical Engineering, Vol 100, May 2022.

[17] C.G.M. Vishwas and R. S. Kunte, "An Image Cryptosystem based on Tent Map," Third International Conference on Smart Systems and Inventive Technology, pp. 1069-1073, 2020.

[18] Vishwas C.G.M and R Sanjeev Kunte, "Image Encryption using Hybrid Chaotic Cryptosystem," Grenze International Journal of Engineering and Technology, Vol 9, Issue 1, 2023.

[19] Deb S, Bhuyan B, "Chaos-based medical image encryption scheme using special nonlinear filtering function based LFSR," Multimedia Tools and Applications, 80, pp. 19803–19826, 2021.

[20] Chai, X., Yang, K. & Gan, Z., "A new chaos-based image encryption algorithm with dynamic key selection mechanisms," Multimedia Tools & Applications 76, pp. 9907–9927, 2017.

[21] Chai XL, Han DJ, Lu Y, Chen YR, Gan ZH., "A novel image encryption algorithm based on the chaotic system and DNA computing," International Journal of Modern Physics C, Vol 28, pages 1750069, 2017.

[22] Yue Wu, Joseph P. Noonan, and Sos Again, "NPCR and UACI Randomness Tests for Image Encryption," Cyber Journals: Journal of Selected Areas in Telecommunications, pp. 31-38, 2011.

[23] Sun. SL, "A Novel Hyperchaotic Image Encryption Scheme Based on DNA Encoding, Pixel-Level Scrambling and Bit-Level Scrambling," in IEEE Photonics Journal, vol. 10, no. 2, pp. 1-14, 2018.

[24] Gao TG, Chen ZQ., "A new image encryption algorithm based on hyper-chaos," Physics Letters A, Vol 372, pp. 394–400, 2008.

[25] Teng L, Wang XY., "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," Optics Communication, Vol 285, pp. 4048–4054, 2012.

[26] Chen JX, Zhu ZL, Fu C. "An efficient image encryption scheme using lookup table based confusion and diffusion," Nonlinear Dyn 81, pp. 1151–1166, 2015.

[27] Wang XY, Zhang HL., "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," Nonlinear Dyn 83, pp. 333–346, 2016.

Authors' Profiles

Vishwas C. G. M. is currently working as Assistant Professor in the Department of Information Science & Engineering at Jawaharlal Nehru New College of Engineering (J.N.N.C.E), Shivamogga, Karnataka, India. He completed his Bachelor's in Computer Science & Engineering and Master's in Computer Science & Engineering from JNNCE in 2000 and 2003 respectively. He is interested in research areas including Information Security and Cryptography.

Dr. R. Sanjeev Kunte is currently working as Professor and Head of the Department of Information Science & Engineering at Jawaharlal Nehru New College of Engineering (J.N.N.C.E), Shivamogga, Karnataka, India. His research interests include Cryptography, Information Security, Cloud Computing, and Image Processing.

Varun Yarehalli Chandrappa is a research PhD student at Central Queensland University. He has 19 years of industry experience in embedded systems, IoT (Internet of Things), and database server development. Recently, he has been actively involved in a smart irrigation project with the Cairns Regional Council, focusing on the implementation of an efficient irrigation system. Varun's expertise lies in providing technology solutions that are accessible to everyone. He possesses a strong interest in Cyber-Physical Systems, Artificial Intelligence (AI), and project management solutions. Notably, he holds five patents under his name and has published numerous journals and conference papers. Furthermore, Varun is actively engaged in the IEEE Victoria IoT community, where he serves as an event and publication manager.

How to cite this paper: Vishwas C. G. M., R. Sanjeev Kunte, Varun Yarehalli Chandrappa, "Encryption Using Binary Key Sequences in Chaotic Cryptosystem", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.4, pp.48-60, 2023. DOI:10.5815/ijcnis.2023.04.05