I. J. Computer Network and Information Security, **2024, 4, 102-117**
Published Online on August 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.04.07

Chaotic Map based Random Binary Key Sequence Generation

Vishwas C. G. M.*
J.N.N. College of Engineering / Visvesvaraya Technological University, Belagavi-590018, 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 / Visvesvaraya Technological University, Belagavi-590018, Karnataka, India
E-mail: sanjeevkunte@jnnce.ac.in
ORCID iD: https://orcid.org/0000-0001-8424-582X

Received: 29 March 2023; Revised: 19 October 2023; Accepted: 07 November 2023; Published: 08 August 2024

Abstract: Image encryption is an efficient mechanism by which digital images can be secured during transmission over communication in which key sequence generation plays a vital role. The proposed system consists of stages such as the generation of four chaotic maps, conversion of generated maps to binary vectors, rotation of Linear Feedback Shift Register (LFSR), and selection of generated binary chaotic key sequences from the generated key pool. The novelty of this implementation is to generate binary sequences by selecting from all four chaotic maps viz., Tent, Logistic, Henon, and Arnold Cat map (ACM). LFSR selects chaotic maps to produce random key sequences. Five primitive polynomials of degrees 5, 6, 7, and 8 are considered for the generation of key sequences. Each primitive polynomial generates 61 binary key sequences stored in a binary key pool. All 61 binary key sequences generated are submitted for the NIST and FIPS tests. Performance analysis is carried out of the generated binary key sequences. From the obtained results, it can be concluded that the binary key sequences are random and unpredictable and have a large key space based on the individual and combination of key sequences. Also, the generated binary key sequences can be efficiently utilized for the encryption of digital images.

Index Terms: Linear Feedback Shift Register**,** Binary, Chaotic Map, Chaotic Random Binary Key Sequence, Binary Key Pool.

1. Introduction

Construction and use of pseudorandom binary sequences are vital in various areas such as watermarking, spread spectrum, error control coding, statistical sampling, and cryptography. Providing secured storage and protecting digital data in multimedia communication has gained much attention. Therefore, key sequence generation is a vital part of any encryption system. Initially, algorithms such as the linear congruential method, mid-square method, and linear and non-linear feedback shift registers were used for the generation of pseudorandom bit sequences. Such algorithms were not secure because of their fixed inner linear structure [1].

Chaotic encryption schemes exhibit certain properties which are appropriate for the encryption of images. Chaotic systems are extremely sensitive to initial conditions, pseudo-random, nonperiodic, and ergodicity. Such systems are able to generate a quick and accurate large number of key sequences [2] which are considered secure when compared to LFSR in their properties. Therefore, chaos-based systems are extensively used to generate pseudorandom sequences and even in the design of random bit generators. Oishi and Inoue (1982) proposed the first work on a pseudo-random number generator using a chaotic system. In recent years, major work on chaotic-based pseudorandom sequence generators is being proposed.

The proposed work focuses on generating random binary key sequences based on four chaotic maps. Using each primitive polynomial, uniquely, 61 random binary key sequences can be generated. And using five different primitive polynomials of degrees 5, 6, 7, and 8 with different initialization vectors (IV), 40 individual binary key sequences and 265 binary key sequence combinations from four chaotic maps are generated. Therefore, a total of 305 random binary

key sequences are generated. The desired characteristic of this implementation is in selecting keys randomly from all four maps in diffusion. All the key sequences pass both the NIST and FIPS tests. The key space obtained is greater than 2^{100}.

The paper is structured as follows. Section 2 reviews the existing work on the generation of key sequences. Next, we provide the basics of chaotic maps in section 3. The details of the proposed binary keystream generator are discussed in section 4. The result analysis is presented in section 5. Section 6 presents the conclusion of the evaluated results.

2. Literature Review

In literature, many papers are presented regarding the generation of pseudo-random sequence generators using chaotic sequences, and much attention is given to the generation of the key sequences using chaotic maps based on the Henon chaotic system, Chen chaotic system, Logistic map, and digital chaotic sequence generator. Recently, chaotic cryptosystems are considered a prime source to design pseudorandom binary sequence generators and a few of those works are reviewed in brief.

Liu Shu-Bo et.al., [3] have used a chaotic logistic system and proposed a modified digital chaotic sequence generator. A coupling scheme in which one chaotic subsystem generates perturbation signals to disturb the control parameter of the other one is used. The algorithm uses nonlinear coupled chaotic systems to extend the chaotic orbits which makes the frequency distribution more uniform. This makes the system more resilient to attacks.

Narendra K Pareek et.al., [4] proposed a new binary sequence generator that employs a random bit generator based on two chaotic systems which are cross-coupled. The algorithm uses piecewise linear chaotic maps, i.e., two 1D skew tent maps to generate sequences. This scheme uses cross-coupling in which the output of one map is fed as the initial condition (input) for another and vice versa. The system parameters for both the chaotic maps considered are the same and lie in the chaotic regime. The generated sequence is uniform and random.

Hamed Rahimov et.al., [5] have generated novel random sequence with lengthier period using a combination of logistic chaotic map and LFSR. The system iterates using initial conditions. The pseudo random bit sequence is obtained by combining the outputs of the logistic map and the LFSR. First, an iteration generates two random numbers using the logistic map and a function is used to express them in decimal. The results of NIST's statistical tests show that our proposed method for generating random numbers has a more efficient performance.

Han Ping Hu et.al [6] used Chen chaotic system and proposed a pseudorandom sequence generator. The algorithm addresses the non-uniform distribution problem existing in the Chen chaotic system when the sequence is generated. To generate the real-valued sequences with continuous time, a sampling of the chaotic signal is conducted. Then, an initial value is set to derive a 3D sequence by sampling the chaotic orbit. Results show that the generated binary sequences using the Chen system are pseudorandom and are resistant to attacks.

Huang F. and Zhao Y [7] use Henon chaotic system and proposed a new key-stream generation algorithm. A method is devised which divides the Henon attractor into several nonoverlapping sub regions in 2D space. Therefore, this scheme generates key streams with good statistical properties. This keystream generation process can be applied to three or four-dimensional chaotic systems. The generated key streams have large key spaces, and the cycle length is long enough.

Musheer Ahamed et.al., [8] proposed a method for a pseudo-random number generator that uses the discrete chaotic map. The generated binary key sequence has good cryptographic properties. The proposed algorithm is random and has a large key space.

Ahmed Yousif [9] proposed an algorithm for key generation using chaotic maps. This scheme is based on a chaotic Chebyshev 1D map which depends on the selected parameters. The generated key is transformed into a sequential element combining it with an ordinary image to generate the cipher. The key stream passes the NIST test for randomness, has good coding strength, and resists security analysis.

Wang L and Hai Cheng [10] have proposed a pseudo-random number generator (PRNG) based on a modified logistic chaotic system. The chaotic values of the previous state drive the system parameters of the next chaotic system. Confusing and rearrangement of the output sequence ensures the security of the system by having a large key space and minimizing the attacks. The result analysis proves that the generated sequence is random and passes the statistical tests.

Ansam Sabah Bader et.al., [11] have put forth a method using a 2D Henon map and the 3D Lorenz and generate keys that improve the security of the cryptosystem. The initial conditions of the 2D Henon chaotic map are (x_0,y_0) and for 3D Lorenz chaotic system, (x_0,y_0,z_0). Only four digits float numbers are considered in this process. A scatter procedure is used to achieve randomness. This scheme generates a sequence of random numbers between [0,1] and these numbers are processed to generate keys. This scheme is sensitive to initial conditions and has a large key space.

Rahman Z et.al., in [12] have proposed a key generation technique using the logistic map for AES-driven IoT Security. The algorithm modifies the key-origination matrix and the S-box. In the first key generation process, a 3-Dimensional Key Generation Matrix (3DKGM) is built. The algorithm runs till the bytes of the second and third keys are generated. The first byte of the final key is obtained after performing the XOR operation between the obtained first, second, and third keys. This scheme encrypts data in large volumes and is secure against linear and differential attacks.

From the literature survey, it can be found that most of the existing work concentrates on pseudo random bit generators based on a single chaotic system. The logistic map is widely used which has a smaller key space. This makes the pseudo random bit generators based on chaotic systems insecure. Also, the LFSR does not satisfy the unpredictability. Therefore, combining more than one chaotic map for the generation of key sequences increases key space, and combining keys from more than one chaotic map increases the randomness and unpredictability of the key sequences.

3. Basics of Chaotic Maps

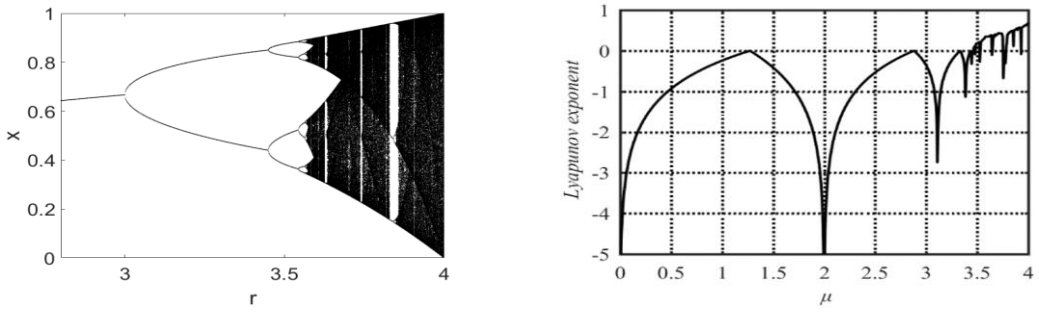
3.1. Logistic Map

The map was proposed by Robert May in 1976 who was a biologist. It is expressed as a polynomial mapping of degree 2. It is a classic description exhibiting both complex and chaotic behavior that evolved from simple nonlinear dynamical equations. Equation (1) shows the mathematical representation of the logistic map.

$$x_{n+1} = r * x_n(1 - x_n) \tag{1}$$

where x_n is between 0 and 1 and r represents the bifurcation parameter, and the system behaves chaotically when the value of r > 3.57 and r < 4.

The bifurcation diagram of a logistic map is depicted in fig.1(a). Initially, the attracting set starts with one point that divides itself into two at r = 3. Initial values exhibit oscillations of 2 and 4 values for r in the ranges (3, 3.44949) and (3.44949, 3.54409) respectively. When the value of r is in the range of 3.56995 to 4, considering all initial conditions, there exists no oscillation for a finite period. Making minute deviations in the initial condition changes the results drastically over a period. Fig.1(b) shows that for the value of r between 3.57 to 4, LE>0. The system behaves chaotically for the value of r between 3.57 and 4. If the value of r exceeds 4, all the initial values diverge and leave the interval [0,1].



(a) (b)

Fig.1. Logistic map (a) Logistic map bifurcation diagram (b) Logistic map lyapunov exponent (LE) diagram

3.2. Tent Map

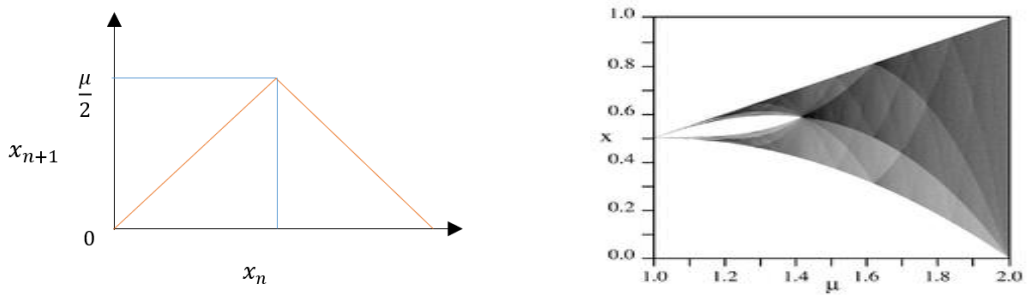
The Tent map considers a parameter μ, and is a real-valued function *fμ.* Tent map is the name given for its tent-like shape of the graph of *fμ*. Therefore, for the values μ in the range 0 and 2, *fμ* maps to [0, 1], which is a recurrence relation. *fμ* can be defined mathematically as follows [13].

$$f_\mu = \mu \, min\{ x, 1 - x \} \tag{2}$$

Iterating a point x_0 in [0, 1] generates a sequence as shown in equation (3).

$$x_{n+1} = f_\mu(x_n) \begin{cases} \mu x_n & for\ x_n < \frac{1}{2} \\ \mu(1 - x_n) & for\ \frac{1}{2} \le x_n \end{cases} \tag{3}$$

The graph for the Tent map is illustrated in fig.2(a).



(a) (b)

Fig.2. Tent map (a) Graph (b) Bifurcation diagram

If μ < 1, the point x=0 is an attractive fixed point, and for all initial values of x i.e. the system converges to x = 0 considering any initial value of x. If μ=1, all the values of $x \leq 1/2$ are the system's fixed points. If μ >1, the system obtains two fixed points, one 0, and the other is $\mu/(\mu+1)$.

The proposed system also uses the Henon map and ACM. More details can be found in [14].

4. Proposed Chaotic Binary Key Sequence Generator

The proposed block diagram of the novel binary key sequence generator based on chaotic maps is shown in fig.3. The system consists of stages such as the generation of chaotic sequences based on four maps viz., Logistic, Tent, Henon, and ACM. After the generation of key sequences of the four maps, they are converted to binary vectors. LFSR rotation is done to select a particular chaotic map to generate a binary key sequence. Here, a combination of key sequences is done by selecting the required key sequences from the generated pool of binary key sequences.

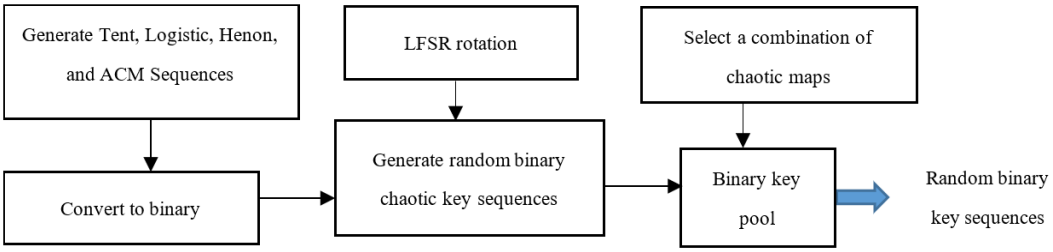


Fig.3. Proposed chaotic binary key sequence generator

4.1. Chaotic Sequences

Four chaotic sequences are generated viz., Logistic, Tent, Henon, and Arnold cat maps in a separate module. The sequences obtained are in decimal and are converted to binary form and stored as a binary vector. The generation of the chaotic logistic sequence is done as in equation 1. The generated values are in the range of [0,1]. These values are converted in the range [0-255] by using equation (4).

$$x = mod(unit8(round(x * power(10,8), 256)) \tag{4}$$

Similarly, the chaotic Tent sequence is generated using equation 3 and is converted to values of range [0-255]. The Henon x sequence is generated using equation (5) and converted to [0-255] as in equation (6).

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

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

Similarly, the Henon y sequence is generated using equation (7) and converted to [0-255] as shown in equation (8).

$$y(i) = b * x(i-1) \tag{7}$$

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

Similarly, the ACM *x* and *y* sequences are also generated.

4.2. Generation of Binary Key Sequences

Initially, four maps, Logistic, Tent, Henon, and ACM are used for the generation of key sequences. The generated keys are in decimal which is converted to binary and is expressed as bit vectors. The proposed implementation uses the conventions as shown in table 1 for the binary sequences generated for all four considered chaotic maps. The logistic sequence is generated and converted to binary (L) as given by algorithm 1. Similarly, the Tent binary sequence (*T*) is also generated. *Hx* and *Hy* are the generated Henon *x* and *y* binary sequences as given by algorithm 2. Similarly, *ACMx* and *ACMy* are the binary sequences of ACM. Whereas *Hxy* and *ACMxy* are the binary sequences combining both *x* and *y* sequences of Henon map and ACM respectively.

Table 1. Binary key sequence notation

Chaotic map	Generated binary key sequences (notation)
Logistic	*L*
Tent	*T*
Henon	*Hxy, Hx, Hy*
Arnold Cat Map	*ACMxy, ACMx, ACMy*

The generation of the binary Logistic sequence is explained in algorithm 1, (Logistic_Seq). The key sequence obtained is in decimal which is converted to a binary vector of size *m*n**8, where *m* and *n* are the dimensions of the input image considered.

Algorithm 1: Logistic_Seq

```
Step 1: Initialize x(1)=0.2 and r=3.85566
Step 2: for i=1 to (m*n)-1 do
      x(i+1)=r * x(i)* (1-x(i))
                end for
Step 3: x=mod(round((x * power(10,8)),256))
Step 4: x=uint8(x)
Step 5: L=de2bi(x,8)
Step 6: L=reshape(L, m*n*8,1)
```

Similarly, the binary Tent sequence is generated. The generation of the Henon sequence is explained in algorithm 2 (Henon_Seq). The generated Henon *x* and *y* sequences are converted to a binary vector of size *m*n**8.

Algorithm 2: Henon_Seq

```
Step 1: Initialize a=1.4, b=0.3, x(1)=0, y(1)=0
Step 2: for i=2 to m*n do
                x(i)=1-a *(x(i-1)^2) + y(i-1)
                Henon_x_seq=mod(round(x * power(7,6)),256)
                y(i) = b * x(i-1)
                Henon_y_seq =mod((y * power(7,6)),256)
        end for
Step 3: Henon_x_seq=uint8(Henon_x_seq)
Step 4: Henon_y_seq=uint8(Henon_y_seq )
Step 5: Hx=de2bi( Henon_x_seq,8)
Step 6: Hx=reshape(Hx,m*n*8,1)
Step 7: Hy=de2bi( Henon_y_seq,8)
Step 8: Hy=reshape(Hy,m*n*8,1)
```

For *Hxy* and *ACMxy* key sequences, the keys are selected based on the division operator. During binary key sequence generation, the multiplication value of the indices (locations) of the pixels of the Lena image in a particular loop is evaluated. If the result of the division is even, then 8 bits from the generated Henon x sequence are selected as the random key based on a pseudo random number, *RN*. Or else an 8-bit random key from the Henon *y* sequence is selected. Here, *RN* is indexed to the binary Henon *x* or *y* sequence for the selection of a random key. Thus, the generated random binary *Hxy* key sequence comprises bits from both *Hx* and *Hy* binary sequences. The devised method for the generation of *Hxy* is presented in algorithm 3. Similarly, the *ACMxy* sequence is also generated.

Algorithm 3: Division

```
Step 1: if ((i*j)/2==0)
```

```
Step 2:        Hx(RN)
Step 3:        for j=1 : 8 do
Step 4:              Hxy(j)=Hx(RN+j)
               end for
Step 5:        randomkey=Hxy
Step 6: else
Step 7:        Hy(RN)
Step 8:              for j=1 : 8
Step 9:                    Hxy(j)=Hy(RN+j)
               end for
Step 10:       randomkey=Hxy
Step 11: end
```

4.3. Linear Feedback Shift Register (LFSR) (Rotation and Working Mechanism)

An 8-bit LFSR is used with initialization and primitive polynomials of degrees 5, 6, 7, and 8 are considered for the generation of key sequences. Out of four chaotic maps, the LFSR chooses one chaotic map for the generation of key sequence. The selection of a particular chaotic map out of four chaotic maps generated is done based on the two most significant bits (MSB) of the LFSR as shown in fig.4. Also, a combination of generated binary sequences from *L, T, Hxy, Hx, Hy, ACMxy, ACMx, and ACMy* can be made which gives individual 8 binary key sequences and 53 combinations of binary key sequences from four chaotic maps generating a total of 61 binary key sequences.

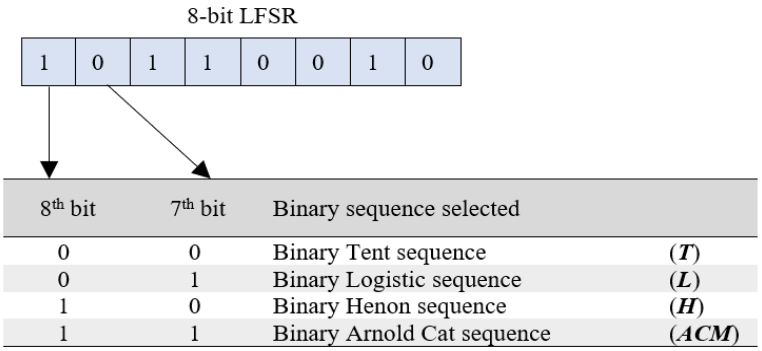


Fig.4. Selection of binary key sequence on MSB bits

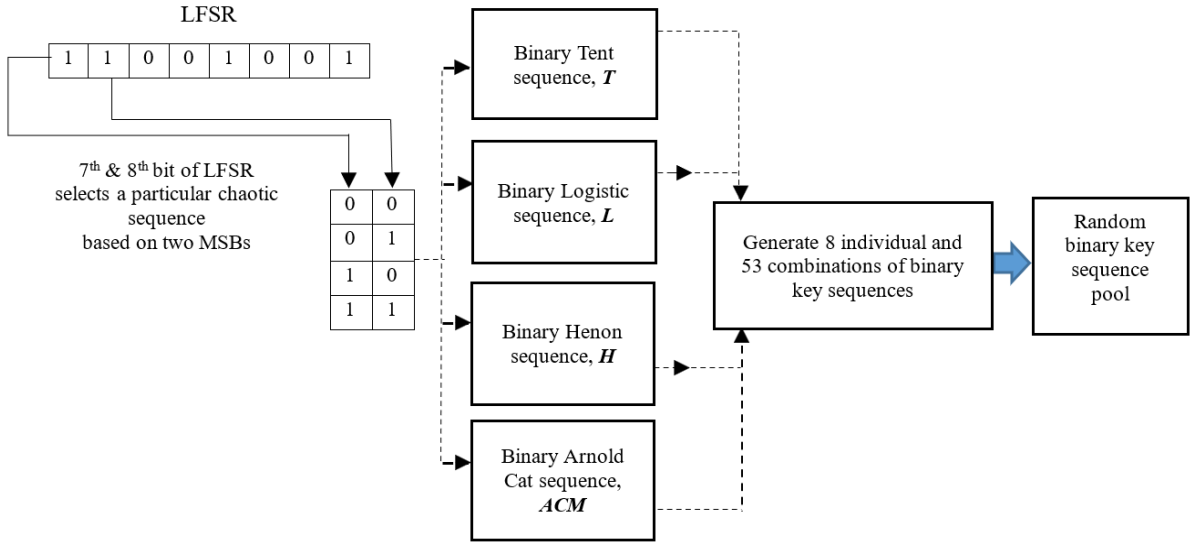


Fig.5. Generation of random binary chaotic key sequences

4.4. Generation of Random Binary Chaotic Key Sequences

One method to generate the binary key sequence is to multiply the real number with a large integer and discard the fractional part. Then take the modulus of the integer number by a small integer. Finally represent the integer in binary. This method is used in the proposed work. Also, many binarization methods are proposed in the literature which can be found in [15-18]. In the proposed work, a random binary key sequence pool is constructed as shown in fig.5 which consists of 61 binary key sequences. Any combination of key sequences can be generated using binary Logistic (*L*),

Tent (*T*), and both *x* and *y* sequences, (*Hxy*) of Henon map and ACM, (*ACMxy*). Similarly, binary key sequences are generated using all four chaotic maps. A total of 8 individual binary key sequences, one from each, Logistic (*L*) and Tent map (*T*) respectively, three from Henon map, that is, *Hx*, *Hy*, and *Hxy*, similarly, three from Arnold Cat Map, that is, *ACMx, ACMy,* and *ACMxy,* can be generated. And 53 combinations of binary key sequences can be generated by combining two, three, and four combinations of chaotic binary sequences from *L, T, Hx, Hy, Hxy, ACMx, ACMy,* and *ACMxy*. Therefore, the binary key pool consists of the generated random binary key sequences.

5. Experimental Results

5.1. Possible Combination of Key Sequences

In this implementation, the binary key sequences are generated using the primitive polynomials of degrees 5, 6, 7, and 8 as shown in table 2. From all four chaotic maps considered, 8 individual binary key sequences are generated. A combination of any two key sequences generates 22 random binary key sequence combinations. A combination of any three key sequences gives binary key sequences of 22 combinations and combinations of four key sequences give 9 random binary key sequence combinations. Hence, a total of 61 binary key sequences can be generated by using each polynomial (table 6).

Table 2. Primitive Polynomials of degrees 8, 7, 6, and 5

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

5.2. NIST SP 800-22 Revision 1a Testing

NIST SP 800-22 REV.1a Test suite [19] is performed to check the randomness of the generated sequences. The result reveals the randomness of the binary key sequence with 99% confidence if the p-value ≥ 0.01. All 61 binary key sequences generated are submitted to the tests in NIST. A sample of a NIST test performed on a particular key sequence combination of binary Tent (*T*), Logistic (*L*), Henon y sequence (*Hy*), and ACM y sequence (*ACMy*) with p-values are presented in table 3. The binary sequence passed all the tests with a p-value ≥ 0.01.

The result of the NIST test conducted on individual and all the possible combinations of the binary key sequences which are generated for the primitive polynomials of degrees 5, 6, 7, and 8 is considered. All the 305 binary key sequences generated pass the NIST test with a p-value ≥ 0.01. The average pass percentage of a total of 305 binary key sequences is 100%. It is observed that the binary chaotic key sequences exhibit randomness.

5.3. FIPS 140-2 Test

The Federal Information Processing Standard (FIPS) test [20] is executed on 20,000 consecutive bits which are randomly chosen from generated binary key sequences. The FIPS test is carried out for all the binary key sequences from primitive polynomials of degrees 5, 6, 7, and 8. From each primitive polynomial, 61 binary key sequences are generated. In total, all 305 binary key sequences generated from five primitive polynomials listed in table 2 are submitted to the FIPS test. It is observed that all 305 binary key sequences pass the monobit, run, long run, and the Poker test. Therefore, the binary key sequences pass the FIPS test and qualify for the randomness test successfully.

5.4. Key Generation Time

A Lena image of dimension 100x100 is considered to generate binary key sequences for which a total number of 10,000 keys must be initially generated. Fig.6 shows the generation time for a few of the binary key sequences considering primitive polynomials of degrees 5, 6, 7, and 8. Also, the count of keys involved in the generation of key sequence when a combination of chaotic sequences is plotted in fig.7. Count 1 and count 2 represent the number of keys chosen from the Tent map and Logistic map. Count 3 and count 4 are the number of keys chosen from Henon and ACM respectively. In total, the number of keys selected from all four counts is 10000, which justifies that the selection of keys from an individual chaotic map is correct. The implementation is carried out on MATLAB R2019b, on Windows 10 O.S, Intel Core i5 with 8.00 GB RAM. The key generation time of all the 305 binary key sequences consumes less than 1 second.

5.5. Pass Percentage

For the Lena image of dimension 256X256, for each primitive polynomial, each binary sequence of length 5,24,288 bits is generated. For 61 binary sequences generated from each primitive polynomial, a total of 83,88,608 bits are generated. From all five primitive polynomials, a total of 4,19,43,040 bits are generated. All the 61 binary key sequences generated from a polynomial are submitted to NIST and the FIPS test. The passing percentage of the binary

key sequences is shown in fig.8. Each binary key sequence is random with a confidence of 1, which implies that the binary key sequences generated have passed all the tests and have good statistical properties.

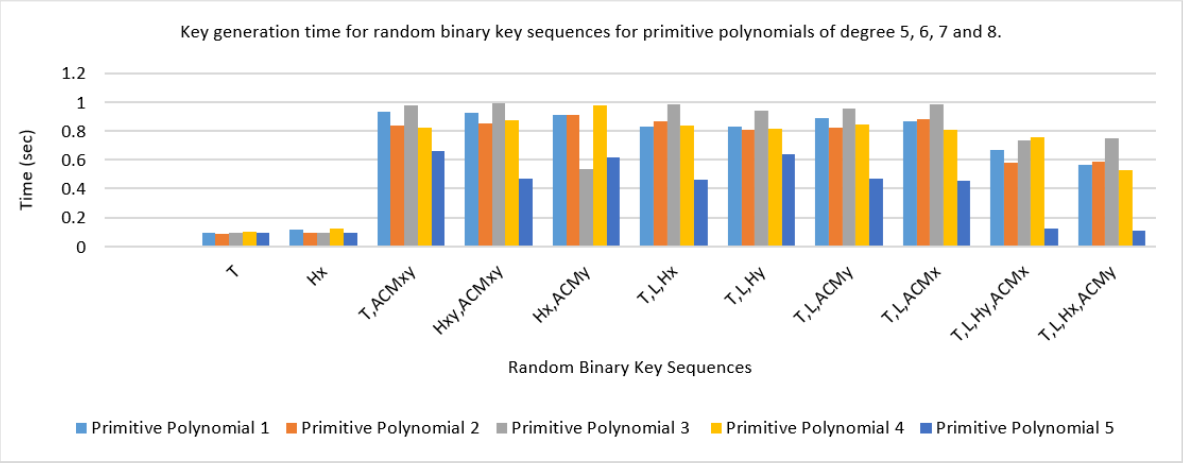
Table 3. NIST Test for binary key sequence *T,L,Hy,ACMy* for a primitive polynomial of degree 6, $x^6+x^5+ x^3+x^2 +1$

Test No.	Type of Test	P-Value			Conclusion
1	Frequency Test (Monobit)	0.6023853065166924			Random
2	Frequency Test within a Block	0.13291798532487312			Random
3	Run Test	0.33705730459729			Random
4	Longest Run of Ones in a Block	0.16003509627314055			Random
5	Binary Matrix Rank Test	0.26071546898407283			Random
6	Discrete Fourier Transform (Spectral)	0.8830523678176525			Random
7	Non-Overlapping Template Matching	0.4772898953699347			Random
8	Overlapping Template Matching Test	0.3953535391911965			Random
9	Linear Complexity Test	0.3740001983761567			Random
10	Serial test	0.2570501697260661			Random
11	Approximate Entropy Test	0.036192450181021504			Random
12	Cumulative Sums (Forward) Test	0.8252760788758613			Random
13	Cumulative Sums (Reverse) Test	0.910767715714798			Random
14	Random Excursions Test:	State	Chi-squared	P-Value	Conclusion
		-4	2.234561935071583	0.8158283005170832	Random
		-3	2.5404702290076346	0.7703882214710572	Random
		-2	1.3532183583074167	0.9293614640298524	Random
		-1	7.183206106870228	0.20736771507584842	Random
		+1	9.335877862595419	0.09639407010207023	Random
		+2	7.034869475073037	0.21805991208549574	Random
		+3	1.889575572519084	0.8642036750862642	Random
		+4	9.851846081944228	0.0795436241651087	Random
15	Random Excursions Variant Test:	State	Counts	P-Value	Conclusion
		-9.0	84	0.4812812419290967	Random
		-8.0	86	0.47286758270383333	Random
		-7.0	90	0.4823520673174424	Random
		-6.0	91	0.45621288730484266	Random
		-5.0	94	0.4460863382097743	Random
		-4.0	128	0.9441521364333806	Random
		-3.0	155	0.5072698684863843	Random
		-2.0	164	0.239167161372414	Random
		-1.0	150	0.24046556082213122	Random
		+1.0	128	0.8529618805590452	Random
		+2.0	115	0.5682022730188173	Random
		+3.0	118	0.7194631852678391	Random
		+4.0	119	0.7793174095653339	Random
		+5.0	95	0.45847369005900795	Random
		+6.0	93	0.4790430972840396	Random
		+7.0	102	0.6192541820681239	Random
		+8.0	108	0.7137037894539473	Random
		+9.0	105	0.6968460898391036	Random

A comparison of the implementation of binary key sequences with the existing work for the NIST test and the FIPS [21] test is tabulated in table 4 and table 5 respectively. It is observed that the pass percentage for the current implementation is better for both the NIST and the FIPS tests.

5.6. Key Space

The number of possible keys that an attacker should use and break any cryptosystem is termed key space. Table 6 shows the 61 binary key sequences generated for the considered primitive polynomial of degree 8, $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$, in which key space of individual and combination of binary key sequences is listed. During the generation of the key sequences, in the Tent map, the initial value *x(1)* is taken as 0.2, and for the Logistic map *x(1)*=0.2, *r=3*.85566. For Henon map, *x(1)*=20, *y(1)*=200 and for ACM, *x(1)*=0.2 and *y(1)*=0.3 are initialized. The secrecy of all these parameters must be maintained and can be used as secret keys. The total key space of 2^{108} is enough to resist attack.



Primitive Polynomial 1: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$, Primitive Polynomial 2: $x^8 + x^6 + x^5 + x^4 + 1$,
Primitive Polynomial 3: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$, Primitive Polynomial 4: $x^6 + x^5 + x^3 + x^2 + 1$, Primitive Polynomial 5: $x^5 + x^4 + x^3 + x^2 + 1$.

Fig.6. Key generation time for particular binary key sequences for the primitive polynomials generated for lena image

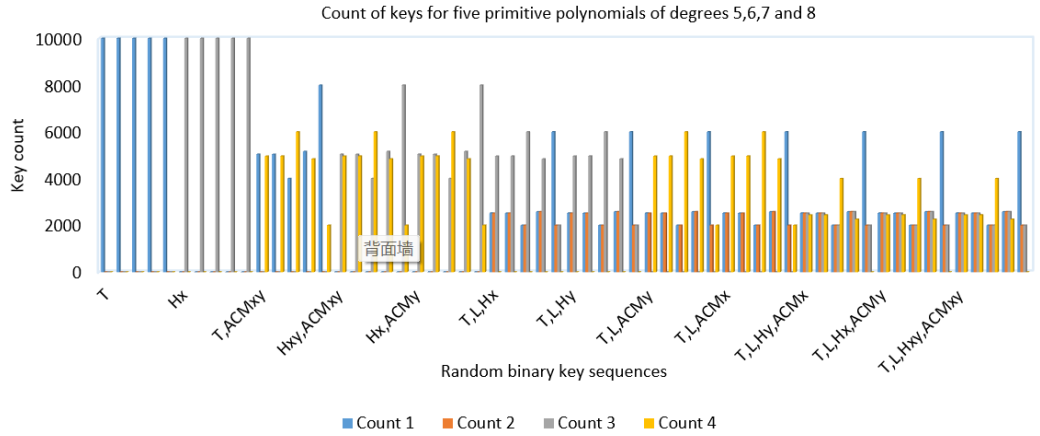


Fig.7. Count of individual keys for particular binary key sequences for the primitive polynomial generated for Lena image

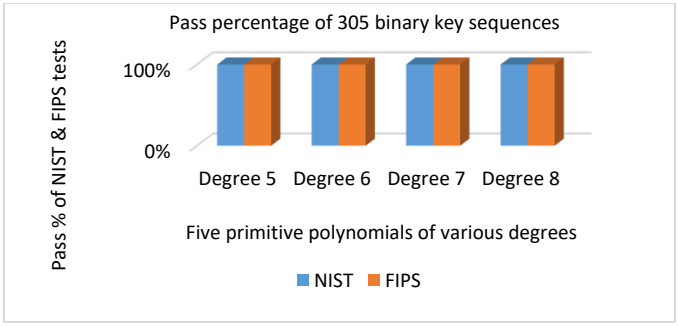


Fig.8. Pass percentage of NIST and FIPS tests

5.7. Performance Analysis

The comparison results of the NIST test with existing work [22-26] and the proposed method with p-values is presented in fig.9. The binary key sequences generated with primitive polynomials of degrees 6, 7, and 8 are compared with existing methods. The binary key sequences pass the test with a p-value ≥ 0.01. It is observed that the binary key sequences generated pass the NIST test with p-values as similar to the considered existing methods.

Table 4. Comparison of proposed system with the existing work on NIST test

NIST Tests	Liu et.al.,[1]	Hamed et.al., [5]	Proposed with 305 binary key sequences
Frequency Test (Monobit)	99.20%	99.00%	100%
Frequency Test within a Block	99.40%	99.15%	100%
Run Test	99.10%	99.50%	100%
Longest Run of Ones in a Block	99.50%	98.60%	100%
Binary Matrix Rank Test	-	-	100%
Discrete Fourier Transform (Spectral)	99.40%	99.50%	100%
Non-Overlapping Template Matching	-	-	100%
Overlapping Template Matching Test	99.50%	98.85%	100%
Linear Complexity Test	99.10%	99.20%	100%
*Serial test	99.20%	99.27%	100%
Approximate Entropy Test	99.10%	99.15%	100%
Cumulative Sums (Forward) Test	99.30%	98.95%	100%
Cumulative Sums (Backward) Test	-	98.85%	
Random Excursions Test:	99.00%	-	100%
Random Excursions Variant Test	99.30%	-	100%

* indicates average value, - indicates value not specified.

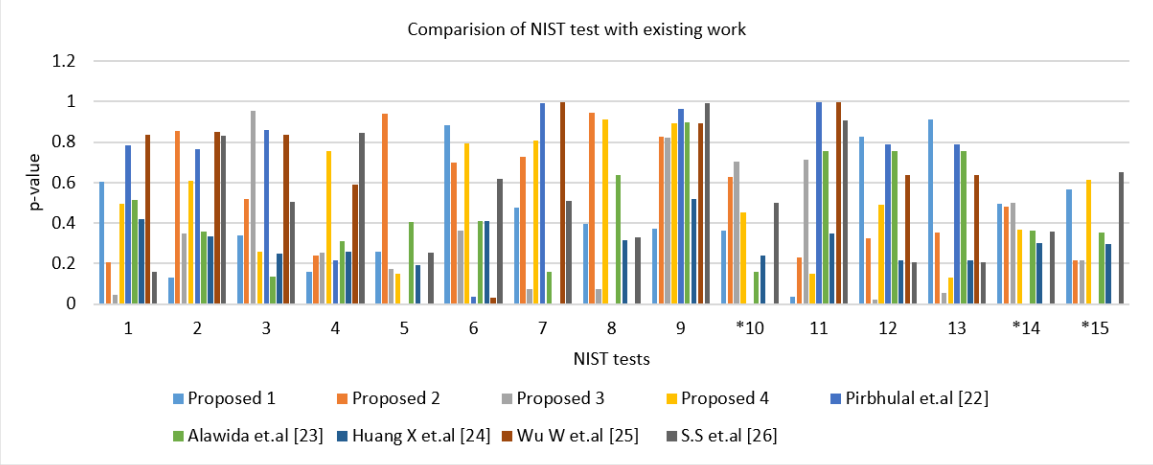
Table 5. Comparison of the proposed system with the existing work on FIPS test

Huang F. et.al., [21]				
Monobit test	Poker test	*Runs test	Long runs test	Proposed with 305 binary key sequences
99.27%	99.89%	99.93%	99.99%	100%

* indicates average value

Table 6. Key space for 61 binary key sequences generated using the primitive polynomial of degree 8, $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$

Individual and Combination of Key Sequences	Key Space
T, L, Hxy, Hx, Hy, ACMxy, ACMx, ACMy	2^{14}
(L,T), (L,Hxy), (L,Hx), (L,Hy), (L,ACMxy), (L,ACMx), (L,ACMy), (T,Hxy), (T,Hx), (T,Hy), (T,ACMxy), (T,ACMx), (T,ACMy), (Hxy,ACMxy), (Hxy,ACMx), (Hxy,ACMy), (Hx,ACMx), (Hx,ACMy), (Hy,ACMx), (Hy,ACMy), (Hx,ACMxy), (Hy,ACMxy)	2^{24}
(T,L,Hxy), (T,L,Hx), (T,L,Hy), (T,L,ACMxy), (T,L,ACMx), (T,L,ACMy), (L,Hxy,ACMxy), (L,Hx,ACMxy), (L,Hy,ACMxy), (L,Hxy,ACMy), (L,Hxy,ACMx), (L,Hx,ACMx), (L,Hy,ACMx), (L,Hy,ACMy), (T,Hxy,ACMxy), (T,Hx,ACMxy), (T,Hy,ACMxy), (T,Hx,ACMx), (T,Hy,ACMx), (T,Hy,ACMy), (T,Hxy,ACMx),(T,Hxy,ACMy)	2^{32}
(T,L,Hxy,ACMxy), (T,L,Hx,ACMxy), (T,L,Hy,ACMxy), (T,L,Hxy,ACMx), (T,L,Hxy,ACMy), (T,L,Hx,ACMx), (T,L,Hx,ACMy), (T,L,Hy,ACMy), (T,L,Hy,ACMx)	2^{38}
Total key space	2^{108}



*Average of multiple tests, -Values not specified
Proposed 1 with Key Sequence: ***T,L,Hy,ACMy*** & Primitive Polynomial: $x^6+x^5+ x^3+x^2 +1$
Proposed 2 with Key Sequence: ***T,L,ACMxy*** & Primitive Polynomial: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$
Proposed 3 with Key Sequence: ***L,Hx,ACMy*** & Primitive Polynomial: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$
Proposed 4 with Key Sequence: ***T,Hxy*** & Primitive Polynomial: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$

Fig.9. Comparison of the NIST test of the proposed algorithm with existing methods

5.8. Time Complexity

In [26], the time to generate 16-bit sequence in one iteration is evaluated as 15 ms. Whereas in our implementation, 80,000 bits of a binary key sequence must be generated to encrypt the input image of dimension 100x100. It takes an average of 359.253 ms to generate any binary key sequence with a primitive polynomial of degree 5 and 670 ms for degrees 6, 7, and 8 (fig.6).

6. Cryptographic Security of the Proposed System

6.1. Proposed Cryptosystem Based on the Generated Binary Key Sequences

The proposed chaotic cryptosystem (fig. 10) which encrypts the input image is based on a binary key sequence generated using four chaotic maps namely, Tent map, Logistic map, Henon map, and Arnold Cat map. The cryptosystem consists of stages namely, confusion, linear feedback shift register (LFSR) operation, generation of random binary chaotic key sequences, and diffusion. Initially, the input image is confused in two steps. Rotation of LFSR is performed to generate a chaotic binary key sequence from four maps considered. The pseudo random number generator generates *RN*, which is indexed to the generated binary chaotic map. Here, an 8-bit random key is chosen based on *RN* from the selected binary chaotic map. In diffusion, these 8 bits of the random key is used to generate the cipher by encrypting each of the 8 bits from the confused image. A combination of key sequences is done by selecting the required key sequences from the generated pool of binary key sequences. 8 individual and 53 combinations of binary key sequences from four chaotic maps are generated giving 61 binary key sequences from each primitive polynomial. In total, from five primitive polynomials (table 2), a total of 305 binary key sequences are generated.

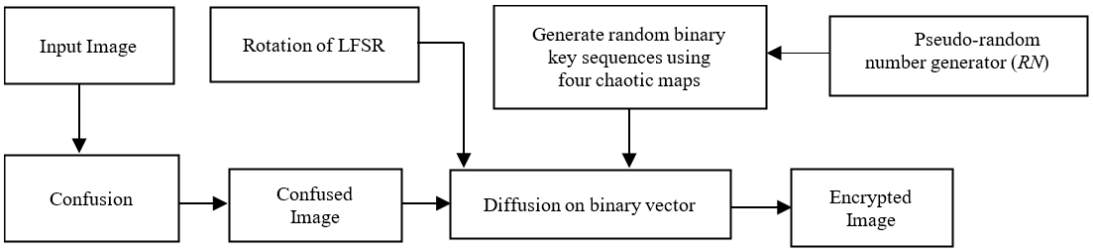
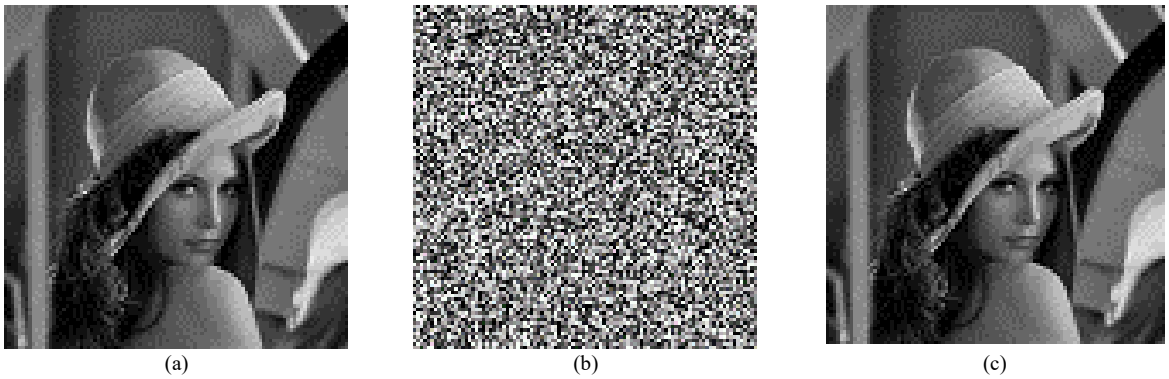


Fig.10. Chaotic Cryptosystem based on random binary key sequences

6.2. Encryption and Decryption

The input Lena image is considered as input to the cryptosystem. The binary ***T,L,Hx,ACMy*** sequence takes 0.106074 sec to generate which is considered as the key to encrypt the input image. Fig 11.(a) is the input to the cryptosystem. From the cipher in fig.11 (b), no clue about the input can be deduced. The cipher is completely different from the input image. After decryption, the original image is successfully retrieved as shown in fig.11 (c) without loss of information.



(a) (b) (c)

Fig.11. (a) Input image (b) Cipher (c) Original image

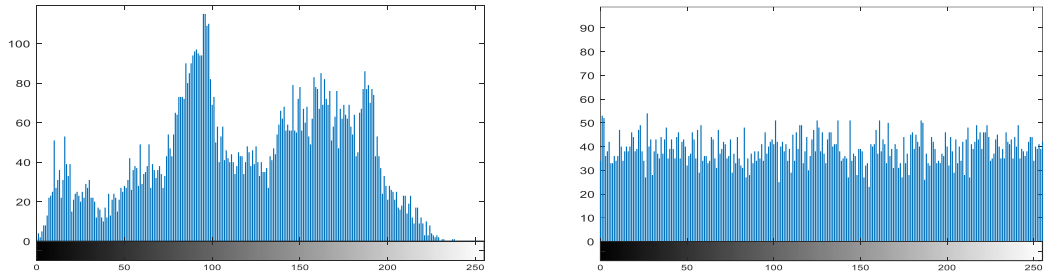
6.3. Histograms Analysis and Chi-square Test

Histogram is the method for measuring the uniformity of the encrypted images. The uniformity of a histogram caused by the proposed encryption scheme is justified by the chi-square test [27] given by the equation (9).

$$X^2 = \sum_{k=1}^{256} \frac{(V_k - e_k)^2}{e_k} \tag{9}$$

Here, k is the number of gray levels which is 256, V_k is the frequency of occurrence of each gray level which is in the range (0–255), and $e_k=M\times N/256$ is the expected occurrence frequency of each gray level. M and N is the size of the plain/cipher images, respectively. If the significance level is 0.05 and the test result on the encrypted images is lower than $X^2_{0.05}(255) = 293.25$, then the histogram indicates better uniformity.

The histogram of the Peppers image is shown in fig.12 (a) The histogram for the corresponding cipher which is obtained by encrypting using the binary key sequence *L,Hx,ACMx* of degree 8 is shown in fig.12 (b). It is evident that the pixels in the cipher are almost flat indicating the appearance of all the pixels in the cipher in almost equal probability. Similar histograms are obtained for other standard images.



(a) (b)

Fig.12. (a) Histogram of the input (peppers). (b) Histogram of the cipher (peppers)

In table 7, the chi-square values for the input and cipher for the standard images are tabulated. The chi-square values evaluated for the cipher are lesser than 293.25 indicating better uniformity of the cipher. The comparison results with the reference papers [27,28] indicate that the proposed system's evaluated chi-square values for standard images is efficient as listed for the existing work.

6.4. Entropy

Entropy measures uncertainty in the pixels of an image. If d is the source of information, the entropy is evaluated given by equation (10).

$$H(d) = \sum_{i=0}^{M-1} p(d_i)\, log \frac{1}{p(d_i)} \tag{10}$$

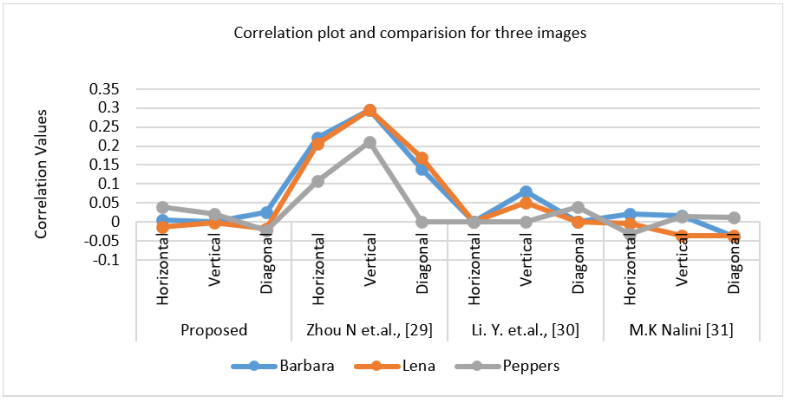
where M is the total number of symbols and $d_i \in d$; $p(d_i)$ is the probability of symbols. Higher entropy value increases the security of the cipher image. If $H(d)$=8 for an 8-bit image, then the cipher is able to resist statistical attacks. From table 8, the computed entropy values are $\cong$ 8 indicating that the proposed system resists statistical attacks. Comparison with existing work indicates that the proposed system evaluates better entropy values.

Table 7. Chi-square test for input and cipher with different binary key sequences and comparison

	Proposed				Norouzi, B. et.al.,[27]	Borujeni S. et.al.,[28]
Image	Primitive Polynomial	Binary Key Sequence	Input	Cipher	Cipher	Cipher
Lena	$x^6 + x^5 + x^3 + x^2 + 1$	*Hxy,ACMy*	18410	233.3440	184	263
Lena	$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$	*T,L,Hxy*	18410	241.0752	184	263
Barbara	$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$	*T,L,Hxy*	11037	277.5296	-	-
Flowers	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$	*L,Hxy,ACMy*	50205	281.9328	-	-
Peppers	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$	*L,Hx,ACMx*	13946	235.2384	270	274

Table 8. Entropy of standard images with primitive polynomials and different binary key sequences

	Proposed				Existing methods		
Image	Primitive Polynomial	Binary Key Sequence	Input	Cipher	Zhou N et.al., [29]	Li Y et.al., [30]	M.K Nalini et.al [31]
Lena	$x^5 + x^4 + x^3 + x^2 + 1$	*T,L,Hx,ACMy*	7.4948	7.9823	7.5983	7.3866	7.7795
Barbara	$x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$	*T,L,Hxy*	7.4291	7.9794	-	-	-
Peppers	$x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$	*L,Hxy,ACMy*	7.5924	7.9774	7.5715	7.3597	7.7408



Barbara with Primitive Polynomial: $x^6 + x^5 + x^3 + x^2 + 1$ and Key Sequence: ***T,L,Hxy,ACMy***
Lena Primitive Polynomial: $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and Key Sequence: ***T,L,Hxy***
Peppers Primitive Polynomial: $x^8 + x^7 + x^6 + x^5 + x^2 + x^1 + 1$ and Key Sequence: ***Hx,ACMx***

Fig.13. Correlation plot and comparison of cipher in three directions

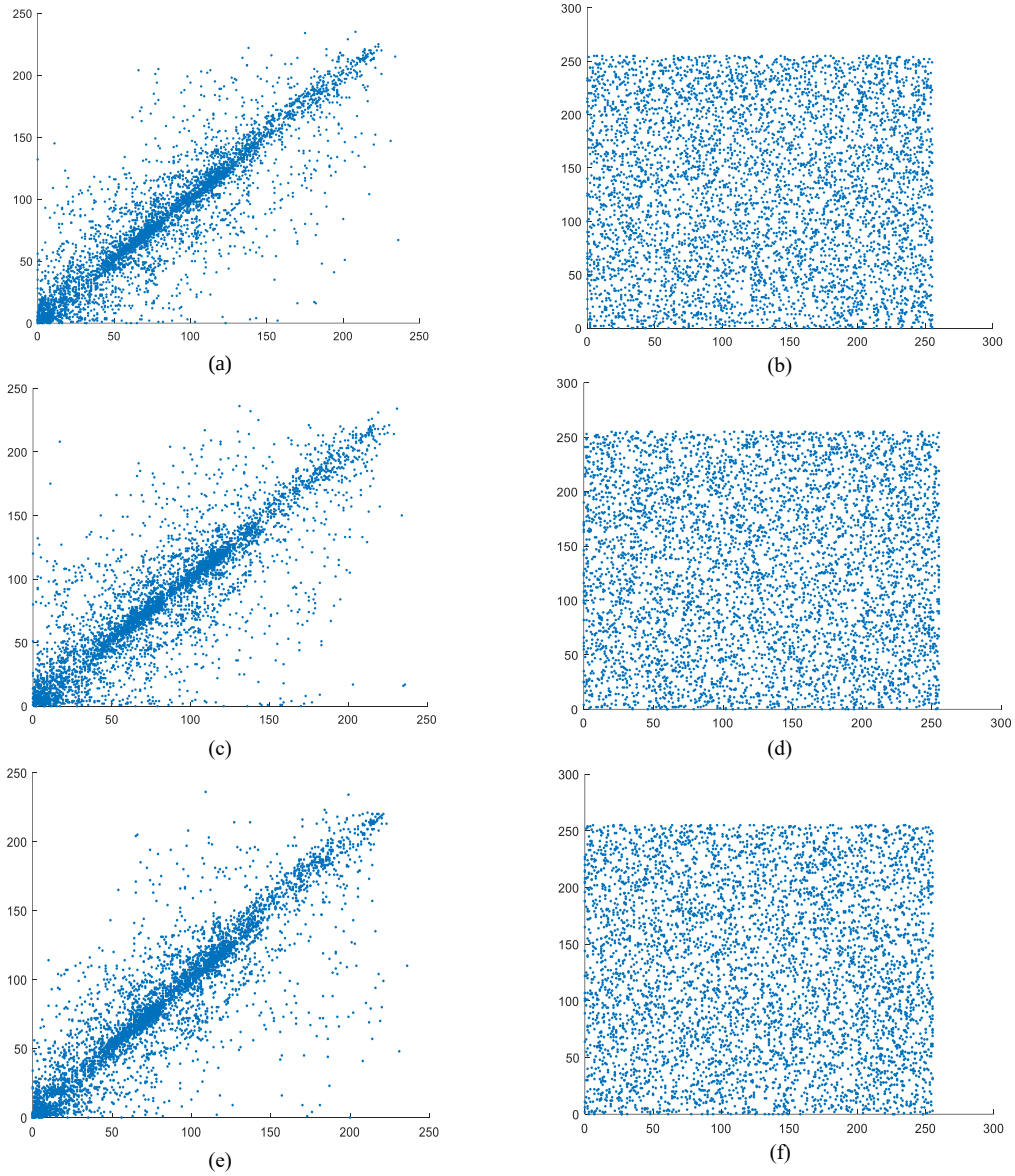


Fig.14. Correlation Coefficient for Lena. (a) Input-Horizontal. (b) Cipher-Horizontal. (c) Input-Vertical. (d) Cipher-Vertical. (e) Input-Diagonal. (f) Cipher-Diagonal

6.5. Resistance Against Correlation Attack

The correlation between neighboring pixels in three directions of considered cipher images is evaluated. Fig.13 shows that in the cipher, there is less correlation among the neighboring pixels which indicates that the system resists correlation attacks. The evaluated correlation values of cipher are also similar to that of the existing work [29-31]. Fig.13 shows the correlation plot for input and cipher generated fort the Lena image with primitive polynomial $x^7 + x^6 + x^5 + x^4 + x^2 + x^1 + 1$ and key sequence: *T,L,Hxy*. From fig.14 (a), (c), and (e), it can be observed that the pixels in all three directions are strongly correlated in the input whereas in (b), (d), and (f), the correlations of the pixels of the cipher in all three directions are scattered.

6.6. Number of Changing Pixel Rate (NPCR) and the Unified Averaged Changed Intensity (UACI)

The chaotic maps are considered highly sensitive to initial conditions. Therefore, any minor changes in the inputs result in large changes in the chaotic trajectories. The NPCR and UACI [32] are two significant methods to exhibit the strength of the cryptosystem in resisting differential attacks with ideal values of 100% and 33.33% respectively. The number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) 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 (11) and (12)) 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.

$$NPCR = \frac{1}{W\,X\,H}\sum_{ij} D(i,j)\,X\,100\% \tag{11}$$

$$UACI = \frac{1}{W\,X\,H}\left[\sum_{ij}\frac{C_1(i,j) - C_2(i,j)}{255}\right]X\,100\% \tag{12}$$

where *W* and *H* are the sizes of the input. C_1 and C_2 are the ciphers in which the input image differs by one pixel. *D(i,j)* is given by the equation (13).

$$D(i,j) = \begin{cases} 0 & C_1(i,j) - C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{13}$$

Three binary key sequences are considered to evaluate the NPCR and UACI values of the Lena image with primitive polynomial of degree 5. For the proposed system, from table 9, the average NPCR is 99.69% and UACI is 32.14%. These values are nearer to the ideal value which indicates that the proposed system resists differential attacks.

Table 9. NPCR and UACI analysis

Primitive Polynomial of degree 5: $x^5 + x^4 + x^3 + x^2 + 1$			
Image	Binary Key Sequence	NPCR	UACI
	T,L,Hx,ACMy	99.70%	32.27%
Lena	*T,L,Hx*	99.70 %	32.27%
	Hx,ACMy	99.67%	31.89%

7. Conclusions

This work focuses on the application of chaotic maps to generate chaotic random binary key sequences. A set of novel random binary key sequences are generated using four chaotic maps considering five different primitive polynomials of degrees 5, 6, 7, and 8. Considering each polynomial, 61 binary key sequences are generated. The randomness of the generated binary key sequences is verified using NIST and FIPS tests. The binary key sequences *T,L,Hy,ACMx*, *T,L,Hx,ACMy* and *T,L,Hxy,ACMxy* with a primitive polynomial of degree 5 which consumes an average of 0.1176 sec, which is very quick can be used to encrypt the images. All 305 binary key sequences are generated by consuming less than 1 second which can be used in efficient encryption. Performance analysis of the proposed generator is carried out which shows that the proposed system is efficient and secure in generating random binary key sequences. Also, the key space is evaluated which is greater than 2^{100}, sufficient to resist attacks. NPCR and UACI are the two criteria used to demonstrate that the proposed cryptosystem resists differential attacks. The values of the correlation analysis proved that the correlation between the input and cipher has considerably reduced after encryption. The entropy values which are nearer to the ideal value disclose that the proposed system resists statistical attacks. Histogram analysis is carried out using the chi-square test. Based on the obtained results and comparison conducted with existing work in literature, it is determined that the proposed key generation method is cryptographically secure and can be effectively used to encrypt digital images using the proposed chaotic encryption algorithm to transmit digital images on the Internet securely.

References

[1] Liu, L., Hu, H., Deng, Y., & Miao, S., "Pseudorandom bit generator based on non-stationary logistic maps," IET Information Security, Vol. 10(2), pp. 87–94, 2016.

[2] 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.

[3] Shu-Bo, L., Jing, S., Zheng-Quan, X., Jin-Shuo L., "Digital chaotic sequence generator based on coupled chaotic systems," Chinese Physics B, Vol. 18(12), pp. 5219–5227, 2009.

[4] Narendra K Pareek, Vinod Patidar, and Krishan K Sud, "A Random Bit Generator Using Chaotic Maps," International Journal of Network Security, Vol.10, No.1, pp. 32–38, 2010.

[5] Hamed Rahimov, Majid Babaei, Mohsen Farhadi, "Cryptographic PRNG Based on Combination of LFSR and Chaotic Logistic Map," Applied Mathametics, Vol. 2, pp. 1531-1534, 2011.

[6] Hu, H., Liu, L., Ding, N., "Pseudorandom sequence generator based on the Chen chaotic system," Computer Physics Communications, Vol. 184(3), pp. 765–768, 2013.

[7] Huang F. & Zhao Y. "New key-stream generation scheme based on Henon chaotic system," Journal of Central South University, Vol. 20(7), pp. 1904–1908, 2013.

[8] Ahmad, M., Doja, M.N., Beg, M.M.S., "A New Chaotic Map Based Secure and Efficient Pseudo-Random Bit Sequence Generation," Security in Computing and Communications Vol. 969, pp 543-553, 2019.

[9] Yousif, A, and Kashmar A. H, "Key Generator to Encryption Images Based on Chaotic Maps," Iraqi Journal of Science, Vol. 60(2), pp. 362–370, 2019.

[10] Wang, L., & Cheng, H., "Pseudo-Random Number Generator Based on Logistic Chaotic System," Entropy, Vol. 21(10), 960, pp. 1-12. 2019.

[11] Ansam Sabah Bader, Shaymaa Hameed and Maisa'a Abid Ali K, "Key Generation based on Henon map and Lorenz System," Al-Mustansiriyah Journal of Sciences, Vol. 31, pp. 41-46, 2020.

[12] Rahman, Z., Yi, X., Khalil, I., Sumi, M., "Chaos and Logistic Map Based Key Generation Technique for AES-Driven IoT Security," Quality, Reliability, Security, and Robustness in Heterogeneous Systems. QShine, 2021.

[13] Li C., Luo, G., Qin., K, and Li C., "An image encryption scheme based on chaotic tent map," Nonlinear Dynamics, Vol. 87, pp. 127–133, 2019.

[14] 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, Vol. 15, No.4, pp. 48-60, 2023.

[15] M. V. Mandi, K. Haribhat, and R. Murali, "Generation of a large set of binary sequences derived from chaotic functions with large linear complexity and good cross correlation properties," International Journal of Advanced Engineering Applications, Vol. 3, pp. 313–322, 2010.

[16] D. Malchev and I. Ibryam, "Construction of pseudorandom binary sequences using chaotic maps," Applied Mathematical Sciences, Vol. 9, no. 78, pp. 3847–3853, 2015.

[17] C. Fatima and D. Ali, "New chaotic binary sequences with good correlation property using logistic maps?" IOSR Journal of Electronics and Communication Engineering, Vol. 5, no. 3, pp. 59–64, 2013.

[18] L. Merah, A.-P. Adda, and H.-s. Naima, "Enhanced chaos-based pseudo random numbers generator," International Conference on Applied Smart Systems, pp. 1–7, 2018,

[19] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks et.al., "SP 800-22 rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.

[20] D. Hurley-Smith, C. Patsakis, and J. Hernandez-Castro, "On the unbearable lightness of fips 140-2 randomness tests," IEEE Transactions on Information Forensics and Security, 2020.

[21] Huang, F., & Zhao, Y.), "New key-stream generation scheme based on Hénon chaotic system," Journal of Central South University, Vol. 20(7), pp. 1904–1908, 2013.

[22] Pirbhulal S, Zhang H, Wu W, Mukhopadhyay SC, Zhang Y-T., "Heartbeats based biometric random binary sequences generation to secure wireless body sensor networks," IEEE Transactions on Biomedical Engineering", Vol. 65(12), pp. 2751-2759, 2018.

[23] Alawida M, Samsudin A, Teh J S., and Alshoura W. H, "Digital cosine chaotic map for cryptographic applications," IEEE Access. Vol. 7, pp. 150609-150622, 2019.

[24] Huang X, Liu L, Li X, Yu M,Wu Z, "A new two-dimensional mutual coupled logistic map and its application for pseudorandom number generator", Mathematical Problems in Engineering, pp. 1-10, 2019.

[25] Wu W, Pirbhulal S, Li G. Adaptive computing-based biometric security for intelligent medical applications. Neural Computing and Applications. Vol. 32, pp. 11055-11064, 2020.

[26] Siddaramanna S, Sarapady Venkatramanayya S., "Generation of chaotic random binary sequences for cryptographic applications," Concurrency Computation: Practice and Experience, e6497, 2021.

[27] Norouzi, B., Seyedzadeh, S.M., Mirzakuchaki, S., Mosavi M. R, "A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos," Multimedia Tools and Applications, Vol. 74, pp. 781–811, 2015.

[28] Borujeni S. E, Eshghi M, "Chaotic image encryption system using phase-magnitude transformation and pixel substitution," Telecommunication Systems, Vol. 52, pp. 525–537, 2013.

[29] Zhou N, Zhang A, Zheng F, Gong L, "Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing," Optics and Laser Technologies, Vol. 62(2), pp. 152–160, 2014.

[30] 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.

[31] M. K, Nalini and K.R, Radhika, "Secured Key Generation for Biometric Encryption using Hyper-chaotic Map and DNA Sequences," ICICNIS 2020, http://dx.doi.org/10.2139/ssrn.3769813, 2021.

[32] Siddaramanna S, Sarapady Venkatramanayya S, “Key Sequences based on Cyclic Elliptic Curves over GF(2^8) with Logistic Map for Cryptographic Applications,” Concurrency Computation: Practice and Experience, 2022.

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 J.N.N.C.E 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.

How to cite this paper: Vishwas C. G. M., R. Sanjeev Kunte, "Chaotic Map based Random Binary Key Sequence Generation", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.4, pp.102-117, 2024. DOI:10.5815/ijcnis.2024.04.07