A New Chaotic Jerk System with Cubic and Hyperbolic Sine Nonlinearities and Its Application to Random Number Generation and Biomedical Image Encryption

Rameshbabu Ramar[1]★, Sundarapandian Vaidyanathan[2], Akif Akgul[3], and Berkay Emin[4]

[1]Department of Electronics and Communication Engineering, V.S.B. Engineering College, Karur, Tamil Nadu, India – 639111. Email: rrameshbabu@vsbec.com, Mobile No. +91 8015166025

[2]Centre for Control Systems, Vel Tech University, 400 Feet Outer Ring Road, Vel Nagar, Avadi, Chennai-600062, Tamil Nadu, India. Email: sundar@veltech.edu.in, Mobile No. +91 9566113754

[3]Department of Computer Engineering, Faculty of Engineering, Hitit University, Corum, 19030, Turkiye. Email: akifakgul@hitit.edu.tr, Mobile No. +90 505 361 32 57

[4]Department of Electronics and Automation, Osmancık Omer Derindere Vocational School, Hitit University, Corum, 19500, Turkiye. Email: berkayemin@hitit.edu.tr, Mobile No. +90 536 512 17 71

Abstract

In this research paper, a new chaotic jerk system is proposed, which is constructed using cubic and hyperbolic sine nonlinearities. A detailed dynamical analysis of the chaotic jerk system is presented with the bifurcation diagrams and Lyapunov exponent spectrums. The novelties of the proposed system are that it can exhibit bistability, amplitude control, and offset boosting control. A random number generator (RNG) is designed using the proposed chaotic jerk system. The study was developed in the Python-based Google Colaboratory environment. The obtained random numbers have successfully passed the NIST 800-22, FIPS140-1, and ENT statistical tests, and it has been shown that they can be used successfully in encryption areas. Biomedical image encryption application was carried out using the generated random numbers. Finally, the reliability of the encryption process has been proven by performing histogram, correlation, NPCR-UACI, entropy analyses, key space analysis, key sensıtıvıty analysis, and robustness analyses.

Keywords: Chaotic systems, Jerk systems, Image Encryption, Random Number Generator, Security Analysis

1. Introduction

In recent years, many chaotic and jerk systems have been introduced with hidden attractors [1], memristor [2], and coexisting attractors [3]. The chaotic jerk systems have many engineering applications such as communication systems [4], wireless networks [5], and biomedical signals [6].

In [7], a jerk system with coexisting attractors was introduced and amplitude control was studied. In [8], a chaotic jerk system with multistability properties was described. In [9], the time-delay effect of a chaotic jerk system was analyzed. In [10], self-excited and hidden chaotic attractors in a jerk system were discussed. In [11], a chaotic jerk system with bistability properties was studied. In [12], a generalized Moore - Spiegel system was studied for multistability properties. In [13-16], memristor neural systems were analyzed for coexisting attractors.

Kengne et al. [17] proposed a chaotic jerk system with the dynamics

$$\dddot{x}+\sigma\gamma\dot{x}+\ddot{x}-\sigma x+\sigma\varepsilon\sinh(\rho x)=0 \tag{1}$$

with the parameter values $\sigma=9.3,\gamma=2$.

Joshi and Ranjan [18] introduced a chaotic jerk system with the dynamics

$$\dddot{x}+\beta\ddot{x}+(\alpha+1)\beta\dot{x}\pm\alpha\beta\gamma\sinh(x)=0 \tag{2}$$

with the parameter $\alpha=0.52,\beta=110,\gamma=300$.

Volos et al. [19] proposed a chaotic jerk system with the dynamics

$$\dddot{x}+x+b\ddot{x}+a\sinh(\dot{x})=0 \tag{3}$$

With the parameter, $a=3.846\times10^{-4},b=0.7$.

Hu et al. [20] presented a new chaotic jerk system with the dynamics

$$\dddot{x}+ax+b^2\dot{x}+c\ddot{x}-d\sinh(x)=\varepsilon \tag{4}$$

Liu et al. [21] described a chaotic system as given in Equation (5).

$$\dddot{x}+0.75\ddot{x}+x+1.2\times10^{-6}\sinh(\frac{x}{0.026})=0 \tag{5}$$

Sundarapandian et al. [22] proposed a chaotic jerk system as represented in Equation (6).

$$\dddot{x}=x-0.4\left[\sinh(x)-\sinh(\dot{x})\right]-0.8\ddot{x} \tag{6}$$

The proposed chaotic jerk system has one cubic nonlinear term, and one hyperbolic sine term and exhibits coexisting attractors when initial conditions are changed. The proposed system has a high positive Lyapunov exponent and exhibits highly complex dynamics compared to existing systems which have hyperbolic sine nonlinearity. The amplitude control and offset boosting control are also observed in the new system. The amplitude control of the proposed system can be achieved by multiplying the control parameter with any one of its signals. In offset boosting control, the location of the attractor can be varied by varying the booster parameter that is added to the particular signal. The comparison of the proposed system with the existing systems is given in Table 1.

Sun et al. [23] designed a random number generator by creating a new chaotic system named SSCS based on CML (Cellular Neural Network) and a logistic map. They performed the safety and speed analyses of the designed generator. Gong et al. [24] proposed a new 4D chaotic system. Based on the proposed chaotic system, a random number generator and image encryption application were performed. Proving the randomness of the generator with the NIST test, they encrypted the gray images. Adhikari and Karforma [25] proposed a chaotic-based image encryption algorithm. The images were encrypted by using the random number sequence as the secret key and XORing the image pixels. Handwritten signature images have been successfully encrypted. Mondal et al. [26] and their chaotic skew tent map and cellular automata-based image encryption application have been implemented. The implemented encryption method is resistant to various known attacks. Cavusoglu et al. designed a PRNG using a new chaotic system. The generated random numbers have successfully passed the NIST tests. In the study, chaos-based image encryption and decryption applications were performed using simple scrambling and XOR operations [27]. Ismail et al. performed a biomedical image encryption application based on double humped logistic map and fractional order logistic map. The system has been tested on medical images such as MRI and lung X-ray [28].

2. New Chaotic Jerk System

In this section, a new chaotic jerk system is introduced and analyzed their dynamical behaviors. The new chaotic jerk system is in the form of Equation (7).

$$\begin{cases} \dot{x} = y \\ \dot{y} = z \\ \dot{z} = ax - x^3 - b\sinh y - cz \end{cases} \tag{7}$$

where $(a,b,c) = (0.3, 0.1, 2.3)$ are the bifurcation parameters.

2.1. Dissipative Nature

The divergence of the system (7) can be calculated using Equation (8).

$$\nabla f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z} = -c \tag{8}$$

Where $f_x = \dot{x}, f_y = \dot{y}, f_z = \dot{z}$. Since the divergence of the system (7) is negative for all positive values of *c*, the proposed system has dissipative nature.

2.2. Equilibrium Points

The equilibrium points of the system (7) can be calculated by letting $\dot{x} = 0, \dot{y} = 0$ and $\dot{z} = 0$ in Equation (7) as given in Equation (9).

$$\begin{cases} y = 0 \\ z = 0 \\ ax - x^3 - b\sinh y - cz = 0 \end{cases} \tag{9}$$

The solution of Equation (9) can be obtained such that $x = \pm\sqrt{a}$ and thus the equilibrium points of the system (7) are, $E_1 = [0,0,0]$, $E_{2,3} = [\pm\sqrt{a},0,0]$.The Jacobian matrix of the system (7) can be written as in Equation (10).

$$J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a - 3x^2 & -b\cosh y & -c \end{bmatrix} \tag{10}$$

The Jacobian Matrix at equilibrium point E_1 can be written as in Equation (11).

$$J(E_1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & -b & -c \end{bmatrix} \tag{11}$$

The polynomial characteristic equation of Equation (10) is given in Equation (12).

$$\lambda^3 + c\lambda^2 + b\lambda - a = 0 \tag{12}$$

Jacobian matrix at equilibrium points $E_{2,3}$ can be written as given in Equation (13).

$$J(E_{2,3}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2a & -b & -c \end{bmatrix} \tag{13}$$

The polynomial characteristic equation of Equation (13) is given in Equation (14).

$$\lambda^3 + c\lambda^2 + b\lambda - 2a = 0 \tag{14}$$

According to Routh-Hurwitz criterion, the Equation (12) and Equation (14) has a positive real root and a negative real root. This indicates that the equilibrium points $E_{2,3}$ are saddle and unstable.

Table 2 summarizes the equilibrium points (E) and eigen values of the new system (7). It can be concluded from Table 2 is that the equilibrium points of the system (7) are unstable.

3. Lyapunov Exponents and Lyapunov Dimension

The Lyapunov exponents for the new system (7) are calculated using Wolf algorithm with simulation time 15000 sec and step size 0.01 as follows:

$$LE_1 = 0.21613, LE_2 = 0, LE_3 = -2.517$$

Since LE_1 has the positive value, LE_2 is zero and LE_3 has the negative value, the proposed system (7) has chaotic nature itself.

The Lyapunov Dimension (D_L) can be obtained as follows:

$$D_L = 2 + \frac{LE_1 + LE_2}{|LE_3|} = 2.086 \tag{15}$$

Equation (15) indicates the fractional dimension of the proposed system (7). The chaotic attractors of the new system (7) are given in Figure 1.

3. Dynamic Analysis

In this section, the dynamical analysis of the new system (7) is conducted using bifurcation plot and Lyaponent spectra. The bifurcation plot is obtained by increasing and decreasing the parameter value using forward continuation (Blue) and backward continuation (Red) method as given in Figure 2(a). It is noted from Figure 2a, the system (7) holds different behaviours such as periodic, n-period, chaotic, etc., particularly ranges from $a = 5$ to $a = 10$, coexistence of attractors observed.

The Lyapunov spectrum of the system (7) for the range of $-10 \le a \le 10$ is presented in Figure 2b which shows there are some ranges with one positive Lyapunov exponents, which confirms the existence of chaotic oscillations of the system (7). For better understanding we plotted the maximum Lyapunov exponent spectrum separately as given in Figure 2c. In order to clarify the bistability phenomena range, we plotted the forward continuation and backward continuation of Maximum Lyapunov exponents in Figure 2d. It is very clear that from $a = 6.8$ to $a = 7.5$ during forward continuation positive Lyapunov exponents and during backward continuation there are no positive Lyapunov exponents for the same range. This confirms the existence of bistable behaviour in the system. We highlighted the bistable region with a window in red color. The coexisting periodic and chaotic attractors of the system (7) are given in Figure 3 where blue indicates the initial condition $X_0 = (-1, 0, 1)$ and red indicates the initial condition $Y_0 = (1, 0, -1)$.

The bifurcation plot and Lyapunov exponent spectrum of the system (7) for the variation of parameter *b* are given in Figure 4. Figure 4a shows that the system (7) holds the chaotic state in the region $b \in [0, 0.03]$, $b \in [0.035, 0.085]$ and $b \in [0.95, 0.145]$. The Lyapunov exponent spectrum for the range of $b \in [0, 0.315]$ is given in Figure 4b. Figure 4b shows that the system (7) has at least one positive Lyapunov exponent in the region $b \in [0, 0.03]$, $b \in [0.035, 0.085]$, and $b \in [0.95, 0.145]$ which indicates the existence of chaotic nature in the system (7). To realize the bistability phenomena, the bifurcation diagram under the parameter *b* is plotted with X_0 (Blue) and Y_0 (Red). It is also noted from Figure 4a that there is no overlapping in the region $b \in [0.14, 0.315]$ which affirms the presence of the coexisting attractors and bistability in the system (7). Figure 5 represents the chaotic and periodic coexisting attractors under the variation of parameter *b*.

The bifurcation diagram and Lyapunov spectrum of the system (7) for the variation of parameter $c \in [1.6, 3]$ is presented in Figure 6. Figure 6a indicates that the system (7) has chaotic states in the region $c \in [1.6, 2]$ and $c \in [2.25, 2.6]$. It also indicates that the system (7) holds period - 4, period - 2 and limit cycle oscillations beyond $c \in 2.6$. The bistability and coexisting attractors are observed in the regions $c \in [1.6, 1.7]$ and $c \in [2.5, 3]$. Figure 6b shows the Lyapunov spectrum of the system (7) for the variation of *c*. Figure 6b also confirms the existence of the chaotic dynamics in the region $c \in [1.6, 2]$. Figure 7 represents coexisting attractors under the various values of the parameter *c*.

4. Amplitude Control

The chaotic system with amplitude control [29] has many engineering applications where the desired amplitude level is required. Equation (16) represents the amplitude-controllable system along the *y* dimension while the amplitude of other variables is unchanged. If we take $x \to x, y \to y/\alpha, z \to z$ then the system (16) becomes similar to the system (7). It means that the introduction of control parameter α in the system (7) does not modify its chaotic dynamics.

$$\begin{cases} \dot{x} = \alpha y \\ \dot{y} = \alpha^{-1} z \\ \dot{z} = ax - x^3 - b\sinh(\alpha y) - cz \end{cases} \tag{16}$$

The equilibrium points of the system (16) are same as that of the system (7). The Jacobian matrix of the system (16) is given as follows:

$$J = \begin{bmatrix} 0 & \alpha & 0 \\ 0 & 0 & \alpha^{-1} \\ a - 3x^2 & -b\alpha\cosh(\alpha y) & -c \end{bmatrix} \tag{17}$$

The characteristic polynomial equation of Equation (17) at E_1 and $E_{2,3}$ are similar to that of the system (7). It indicates that the control parameter α does not modify the stability of the system (7).

Figures 8 (a-b) and Figure (c-d) show the amplitude-controlled attractor along *y* dimension when $\alpha = 0.001$ and $\alpha = 1000$ respectively. Comparing Figure 8 with Figure 1, it can be understood that the amplitude of the state signal *y* is increased to 10^3 times its original value when $\alpha = 0.001$ and decreased to 10^3 times its original value when $\alpha = 1000$. Figure 9 confirms that the chaotic nature of the system (7) is not modified by the control parameter α.

5. Offset Boosting Control

The offset boosting control [30, 31] in the system (7) is achieved by adding a constant β with the signal *z* as given in Equation (18). The offset boosting control is achieved in the system (7) without affecting its stability, dissipativity and Lyapunov exponent values.

$$\begin{cases} \dot{x} = y \\ \dot{y} = z + \beta \\ \dot{z} = ax - x^3 - b\sinh y - c(z + \beta) \end{cases} \tag{18}$$

The equilibrium points of the system (18) can be calculated as, $E_1 = [0, 0, -\beta]$, $E_{2,3} = [\pm\sqrt{\alpha}, 0, -\beta]$. The Jacobian matrix of the system (18) is similar to Equation (10) which is independent of the *z* variable. Thus, the introduction of offset booster β with the *z* variable does not affect the stability of the system (7).

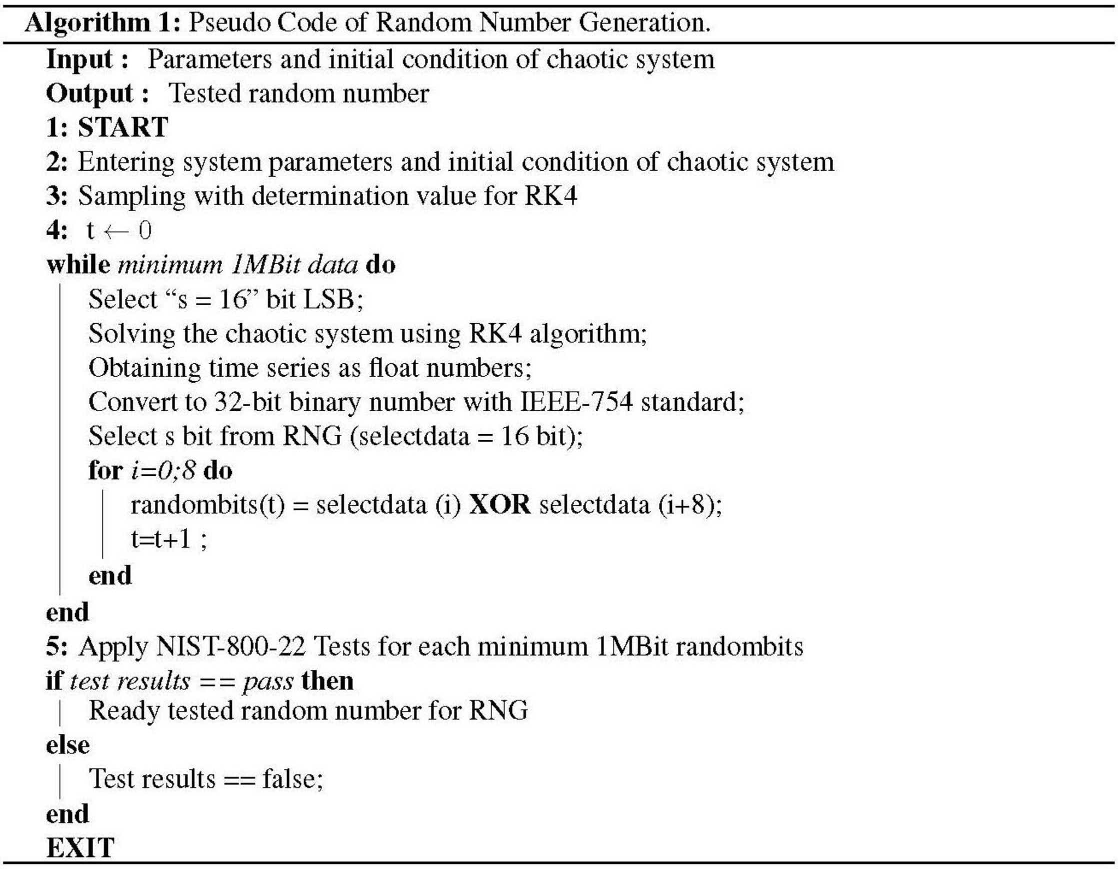
Figure 10 shows the offset boosted attractors of the system (18) along the *z* direction with $\beta = 1$ (blue), $\beta = 20$ (red) and $\beta = -20$ (green). Figure 11 confirms that the Lyapunov exponent values of the system (7) are not modified by the parameter β.

6.Random Number Generator and Statistical Tests

6.1. Random Number Generator Design

In this section, Random Number Generator (RNG) design is examined. The pseudocode of the RNG is given in Algorithm 1. After the initial conditions and system parameters of the chaotic system are determined, the equation phase to be used for RNG is selected. To make the system discrete-time, the RungaKutta-4 solution method is used by selecting the appropriate step interval. Thus, raw values are obtained. Raw values are converted to 32-bit binary form using the IEEE-754 standard.

When the 32-bit numbers with 15 different values in Figure 12 are examined, it can be observed that the values start to become the same as they approach the 0-bit. Toward the end, independent values are obtained. The least significant bit, s = 16 (LSB), was chosen. To increase the randomness in the RNG design, the 16-32 bit sequence was selected. The 16-bit values obtained from the x phases are subjected to an XOR operation as $x[n] - x[n+8] (0 \le n < 8)$. After this process, 8-bit sequences are obtained. The resulting bit sequences are then combined to obtain a total of 10 different bit sequences, each consisting of 1,000,000 bits. Finally, the obtained random bit sequences are subjected to NIST 800-22, FIPS 140-1, and ENT tests.



```
Algorithm 1: Pseudo Code of Random Number Generation.
  Input :  Parameters and initial condition of chaotic system
  Output :  Tested random number
  1: START
  2: Entering system parameters and initial condition of chaotic system
  3: Sampling with determination value for RK4
  4: t ← 0
  while minimum 1MBit data do
      Select "s = 16" bit LSB;
      Solving the chaotic system using RK4 algorithm;
      Obtaining time series as float numbers;
      Convert to 32-bit binary number with IEEE-754 standard;
      Select s bit from RNG (selectdata = 16 bit);
      for i=0;8 do
          randombits(t) = selectdata (i) XOR selectdata (i+8);
          t=t+1 ;
      end
  end
  5: Apply NIST-800-22 Tests for each minimum 1MBit randombits
  if test results == pass then
      Ready tested random number for RNG
  else
      Test results == false;
  end
  EXIT
```

6.2. RNG Statistical Tests

6.2.1. NIST 800-22 Test Suit

The NIST 800-22 test package is an internationally recognized statistical test. The test package is described in detail in the article given in Reference [32]. The test package is used to evaluate that random number generators do not have randomness properties statistically. The NIST 800-22 test includes 15 different statistical test sets and must pass all tests successfully for the number to be considered random.

The random numbers used in the test were obtained from the x phase of the system. A total of 10 million bit sequences were obtained by using 10 sequences of 1,000,000 bits. In addition, the encrypted image was converted into a binary sequence and subjected to the NIST 800-22 test. The significance level α was set to 0.01 so that the result of each test could be considered random at 99% confidence level.

Table 3 presents the NIST 800-22 test results for each of the obtained sequences and all of the P-values exceed the threshold value of the randomness statistical test. This means that the proposed image encryption algorithm shows strong resistance to statistical attacks.

6.2.2. FIPS 140-1 Test

As part of the Federal Information Processing Standards (FIPS), published by the National Institute of Standards and Technology (NIST), FIPS 140-1 covers the security requirements of cryptographic modules and recommends statistical tests for random number generators. The FIPS 140-1 test consists of four different tests: monobit, poker, run and long-term test. The bit string of 20 Kbits in the binary number system is subjected to these four different tests. For the bit sequence obtained from the RNG output to be counted randomly, it must pass four defined tests [33]. In Table 4, the success criteria of each test and the test results of the number sequences obtained from the x phase are given. When the results are examined, it is seen that the bit strings have passed all tests successfully.

6.2.3 ENT Test

The ENT test applies statistical analyses to evaluate the randomness properties of bit sequences. This test, developed by John Walker, plays an important role in the field of computer science and cryptology [34]. The ENT test includes 5 different tests, namely Arithmetic Mean, Entropy, Correlation, Chi-Square and Monte Carlo values, to define the randomness of bit sequences. The ENT test results of the generated bit array are given in Table 5, and the bit array has successfully passed all tests.

7. Image Encryption and Security Analysis

In this section, a biomedical image encryption application is detailed using the generated bit sequences with proven randomness. The biomedical image encryption application was designed in the Google Colaboratory environment. Google Colaboratory is a cloud-based Jupyter Notebook environment using the Python programming language [35]. The security analysis of the encrypted image is performed on the same platform, and the efficiency of the encryption algorithm and the security of the encrypted image are evaluated.

7.1. Encryption and Decryption

This section implemented an image encryption application using the randomly generated bit sequence obtained from the x phase. The application performs encryption and decryption on a 512x512x1 biomedical image. The pseudo-codes for the steps followed during encryption and decryption are provided in Algorithm 2 and Algorithm 3, respectively.

Algorithm 2: Pseudo code of Biomedical Image Encryption Algorithm

```
Input :  Tested random bit sequence and image
Output :  Encrypted image
1: START
2: Entering random bit sequence (randombits) and image data (image)
3: Get the dimensions of the image (w=512,h=512)
4: Convert image to GrayScale
5: Resize image
    img = image.reshape(w * h)
6: t ← 0
for i=0; w * h do
    decimalrandomseq(i)=bintodecimal(randombitseq (t → $t + 8$))
    t = t+8
end
7: Sort decimal array (decimalrandomseq) and get indexes (idxdecrandomseq)
    idxdecrandomseq=argsort(decimalrandomseq)
8: for i=0; w * h do
    confusionimg(i) = img(idxdecrandomseq(i))
end
9: for i=0; w * h do
    encryptedpixel(i) = confusionimg(i) XOR decimalrandomseq(i)
end
10: Resize encrypted image to size w x h
    encryptedimage = encryptedpixel.reshape((w,h))
EXIT
```

Algorithm 3: Pseudo code of Biomedical Image Decryption Algorithm

```
Input :  Tested random bit sequence and encrypted image
Output :  Decrypted image
1: START
2: Entering random bit sequence (randombits) and encrypted image data (encimage)
3: Get the dimensions of the image (w=512,h=512)
4: Resize image
    encimg = encimage.reshape(w * h)
5: t ← 0
for i=0; w * h do
    decimalrandomseq(i)=bintodecimal(randombitseq (t → $t + 8$))
    t = t+8
end
6: Sort decimal array (decimalrandomseq) and get indexes (idxdecrandomseq)
    idxdecrandomseq=argsort(decimalrandomseq)
7: for i=0; w * h do
    confimage(i) = encimg(i) XOR decimalrandomseq(i)
end
8: for i=0; w * h do
    decryptimagepx(idxdecrandomseq(i)) = confimage(i)
end
9: Resize encrypted image to size w x h
    decryptedimage = decryptimagepx.reshape((w,h))
EXIT
```

In Algorithm 2, 'randombits' represents the random bits generated, w and h represent the length and width of the image, respectively, in Step 5 the resizing of the image, in Step 6 'decimalrandomseq' represents the bits converted to integers, in Step 7 'idxdecrandomseq' represents the index sequence obtained as a result of sorting, 'confusionimg' in step 8 represents the image pixels obtained as a result of mixing with the indices, and 'encryptedimage' in step 10 represents the encrypted image. The algorithm steps are given below.

Step 1: The image to be encrypted and the tested random bit sequence are taken into the system.

Step 2: The image is converted to grayscale.

Step 3: The size of the image to be encrypted is transformed into a one-dimensional form.

Step 4: The received random bit sequence is converted into 8-bit random integers.

Step 5: In Algorithm 2, as in step 7, the obtained integers are sorted in ascending order to obtain index numbers.

Step 6: The image is shuffled using the index numbers to create a one-dimensional rearranged image.

Given the original pixel list [p_0, p_1, p_2, ..., p_{n-1}] and the index list to be used for confusing these pixels [i_0, i_1, i_2, ..., i_{n-1}], we can follow these steps to obtain the confused pixel list:

For each index, select the pixel $p_{i_k}$from the original pixel list. Create a new list containing these selected pixels. This list represents the confused pixel list.

Using these steps, we can perform the shuffle process for a given n, changing the original position of each pixel.

For example, with pixels = [123, 132, 112, 80] and indices = [3, 1, 2, 0]:

i_0 = 3, so the first element for the confused list will be p_3= 80.

i_1 = 1, so the second element for the confused list will be p_1 = 132.

i_2 = 2, so the third element for the confused list will be p_2 = 112.

i_3 = 0, so the fourth element for the confused list will be p_0 = 123.

When applying these steps, the confused pixel list will be [80, 132, 112, 123].

Step 7: The obtained integers are subjected to an XOR operation with the confused image pixels.

Step 8: The encrypted image is obtained.

In algorithm 3, a biomedical image is decrypted and a random bit sequence is imported into the system, then the conversion process to an integer at the encryption stage and the index acquisition process is performed, and then, as the reverse of the order in the encryption process, the pixels of the first encrypted image and the integers obtained from the random bit sequence are subjected to the XOR process, the resulting image is a mixed image, the confusion phase is eliminated using the indexes obtained as a result of sorting in this image so that the image is decrypted and the solved image is obtained. The matrix values of the source image before encryption are given in Figure 13 (a), the matrix values after the encryption process (with x phase) are given in Figure13 (b), and the matrix value of the decrypted image is given in Figure13 (c). The security analysis of the encryption process will be discussed in the next section.

7.2. Security Analysis

The reliability of the biomedical image encryption process depends on the performance of the encryption algorithm. The image encryption algorithm should be resistant to brute force, side channel attacks and cryptanalysis attacks. Statistical tests such as histogram, correlation, entropy and differential attack analyses are used in the literature to measure the performance of the image encryption process [36-38]. These tests help to determine the security level of the encryption algorithm by evaluating the statistical properties of the encrypted image.

7.2.1. Histogram Analysis

In this section, histogram analysis of the image encoded with the x phase of the source image is performed. While the histogram graph of the source image is irregular, the histogram graph of the encrypted image is equal and homogeneous. This shows that the encrypted image is resistant to differential attacks [39]. Figure 14 (a) shows the histogram distribution of the source image, while Figure 14 (b) shows that the histogram distribution of the encrypted image is even and homogeneous. This shows that the encryption process is successful and the encrypted image is resistant to differential attacks.

7.2.2. Correlation Analysis

In this section, correlation analyses were conducted on the source and encrypted images, and within the scope of the analysis, correlation coefficients and correlation maps were examined. In a successful encryption process, the correlation between adjacent pixels in the encrypted image should be close to zero [40]. This indicates that the pixel values in the encrypted image are unrelated and independent from each other. Additionally, as observed in the correlation maps in Figure 15, the encrypted images exhibit a homogeneous distribution among adjacent pixels (vertical, horizontal, and diagonal) with no apparent correlation.

In Table 6, the correlation coefficient of the source image is around 1, while the correlation value of the x-phase encrypted image is close to 0. Table 6 shows some correlation coefficient values that are currently in the literature. It is seen that the results obtained are compatible with the studies in the literature. These results show that the encryption process was performed successfully.

7.2.3. NPCR and UACI

This section examines the NPCR (number of pixel change rates) and Unified Average Changing Intensity (UACI) values of the encrypted image. While the NPCR parameter expresses the pixel change rate; The UACI parameter shows the average rate of change in density. In previous studies, it is known that the accepted NPCR value in a good encryption method is greater than 99.6%, and the UACI value is 30% or higher [46]. The NPCR and UACI values of some of the studies obtained in the study and found up-to-date in the literature are given in Table 7. It has been seen that the results of the analysis are compatible with the literature, and it is concluded that the proposed system is also resistant to differential attacks.

7.2.4. Entropy Analysis

Entropy analysis is a method used to measure the complexity of encrypted data. As the complexity of encrypted data increases, obtaining information about the original data becomes more difficult. 8 is accepted as an ideal information entropy value for encryption [50]. As the calculated entropy value approaches the integer 8, the encryption quality increases. The entropy values of some of the studies obtained in the study and found up-to-date in the literature are given in Table 8. When Table 8 is examined, the entropy value of the encrypted image is very close to 8, showing that successful encryption is provided against attacks. The information entropy is calculated as follows: [51]:

$$E = \sum_{i=0}^{2^k - 1} \left[p(s_i) \log_2 \frac{1}{p(s_i)} \right] \tag{19}$$

Here, $p(s_i)$ represents the probability of s_i, L is the number of bits for s_i and is equal to 8.

7.2.5. Key Space Analysis

It is important for an image encryption algorithm to have a large enough security key space to resist brute force attacks. Key space size refers to the total number of different keys that can be used in a cryptosystem. For an ideal encryption algorithm, this number should be greater than 2^{100} [53]. According to the IEEE floating point standard [54], the computational precision of a 64-bit double precision number is about 10^{15}. In our encryption process, the key parameters are x_0, y_0, z_0, a, b, c ‘dir. Given these, the total number of possible secret keys is approx,

$$key = (10^{15\times 6}) \cong 2^{298} > 2^{100} \tag{20}$$

is calculated, indicating that it is resilient to a brute force attack.

7.2.6. Key Sensitıvity Analysis

Key sensitivity analysis is a type of analysis used to detect a change in the key used in an encryption algorithm. The encryption algorithm must be sensitive to the modification of secret keys. A small change in the secret key should result in a large change in the output result. This analysis involves both the encryption and decryption processes. First, in the encryption phase, the image is encrypted using the original key and the encrypted image is obtained. Subsequently, the image is encrypted using a key modified from the original key by a weak change of $t = 10^{-15}$. In the second stage, during the decryption step, the encrypted image is decrypted using the original key to obtain the original image. The decryption key is used to decrypt the image encrypted using the modified key. Differences between images are compared. Figure 16 and Figure 17 respectively show the results of the key sensitivity test performed in the encryption and decryption processes. Based on the differences between images, the images corresponding to different keys are significantly different, ensuring the success of our key sensitivity test.

The developed encryption algorithm uses random numbers generated by a chaos-based RNG, it has a very sensitive dependence on the initial conditions and system parameters of the chaotic system used as a key.

7.2.7. Robustness Analysis

During communication, errors or interruptions can occur at times. In such situations, a portion of the encrypted image may be lost during transmission. The robustness of the system is determined by whether the information can be accurately recovered. A suitable image encryption algorithm should be resistant to noise and data loss, transforming a noisy encrypted image into a recognizable clear image.

Figure 18 shows the results of salt and pepper noise attacks applied to the encrypted image and the images decrypted after the attack. The decrypted image remains recognizable and understandable even when affected by different levels of noise intensity.

Figure 19 shows the decrypted results of images with 25% and 50% data loss in different regions of the encrypted images. The results show that the original image is accessible from the data loss images. Moreover, the assessment of decrypted images affected by pollution and data loss involves the use of PSNR (Peak Signal-to-Noise Ratio).

$$PSNR = 10\log\frac{512\times 512}{\frac{1}{W\times H}\sum_{i=1}^{W}\sum_{j=1}^{H}\left[P(i,j)-D(i,j)\right]^{2}} \tag{21}$$

PSNR values for Figure 18 (d-e-f) and Figure 19 (d-e-f) images are shown in Table 9.

8. Conclusions

In this paper, a new 3-dimensional chaotic jerk system with one cubic and hyperbolic sine nonlinearities is introduced. The proposed system is compared with the existing systems and concluded that the new system has highest positive Lyapunov exponent value. The bifurcation diagrams are plotted with two different initial conditions and realized that the new system can able to produce coexisting attractors. The Lyapunov exponent spectrum of the amplitude controlled and offset boosting-controlled systems indicate that the chaotic nature of the new system is not modified by the introduction of control parameters. Future work is considered as to employ a memristor as the nonlinearity in the proposed system [55,56]. As engineering application, a random number generator is designed using a new chaotic jerk system. After confirming that random numbers can be used safely with statistical tests such as NIST 800-22, FIPS 140-1 and ENT, these numbers were used for biomedical image encryption. To measure the reliability of the encryption process, Histogram, Correlation, NPCR-UACI, entropy, key space analysis, key sensıtıvıty analysis and robustness analyzes were performed and successful results were obtained from each test. Considering the studies in the literature, it can be said that an encryption application using random numbers generated based on chaos is sufficiently secure against attack attacks. As a result of all these research and analyses, a source of information is presented for studies in areas such as random number generation, analysis of random numbers, biomedical image encryption and security analysis. New encryption algorithms for the encryption of video-assisted biomedical data can be proposed and the study can be implemented in embedded card platforms.

References

[1] Vaidyanathan, S., Benkouider, K. and Sambas, A. "Modelling, bifurcation analysis and multistability of a new 4D hyperchaotic system with no balance point, its master-slave synchronisation and MultiSim circuit simulation," *International Journal of Modelling, Identification and Control,* **40 (4)**, pp. 279-293, 2022.

[2] Lai, Q. and Chen, Z. "Grid-scroll memristive chaotic system with application to image encryption," *Chaos, Solitons and Fractals*, **170**,113341, 2023

[3] Lai, Q., Wan, Z., Kuate, P.D.M. et al. "Coexisting attractors, circuit implementation and synchronization control of a new chaotic system evolved from the simplest memristor chaotic circuit," *Communications in Nonlinear Science and Numerical Simulation,* **89**, Article ID 105341, 2020.

[4] Lu, Z., Dieu, N.J.D., Jiang, D. et al. "Novel Duffing chaotic oscillator and its application to privacy data protection," *Physica Scripta,* **98**, Article ID 085248, 2023.

[5] Jiang, D., Njitacke, Z.T., Nkapkop, J.D.D. et al. "A new cross ring neural network: Dynamic investigations and application to WBAN," *IEEE Internet of Things Journal,* **10 (8),** pp. 7143-7152, 2022.

[6] Telem, A.N.K., Fonzin, T.F., Sone, M.E. et al. "A chaos-based DS-CDMA transmission and synchronization for multi-leads medical ECG/EEG through AWGN and Rayleigh channels," *Multimedia Tools and Applications,* 2023.

[7] Ahmad, I. and Srisuchinwong, B. "Simple chaotic jerk flows with families of self-excited and hidden attractors: Free control of amplitude, frequency, and polarity," *IEEE Access,* **8**, pp. 46459–46471,2020.

[8] Vaidyanathan, S., Sambas, A., Zhang, S. et al. "A chaotic jerk system with three cubic nonlinearities, dynamical analysis, adaptive chaos synchronization and circuit simulation," *Journal of Physics: Conference Series, IOP Publishing,* **1179**, Article ID 012083, 2019.

[9] Rajagopal, K., Pham, V.T., Tahir, F.R. et al. "A chaotic jerk system with non-hyperbolic equilibrium: Dynamics, effect of time delay and circuit realisation," *Pramana,***90**, Article ID 52, 2018.

[10] Rajagopal, K., Kingni, S.T., Kom, G.H. et al. "Self-excited and hidden attractors in a simple chaotic jerk system and in its time-delayed form: analysis, electronic implementation, and synchronization," *Journal of the Korean Physical Society,***77**, pp. 145–152,2020.

[11] Kom, G., Kengne, J., Mboupda Pone, J.R. et al. “Asymmetric double strange attractors in a simple autonomous jerk circuit,” *Complexity*, Article ID 4658785, 2018.

[12] Negou, A.N., Kengne, J. and Tchiotsop, D. “Periodicity, chaos and multiple coexisting attractors in a generalized Moore–Spiegel system,” *Chaos, Solitons &Fractals*,**107**, pp. 275–289,2018.

[13] Lai, Q., Wan, Z., Zhang, H. et al. “Design and analysis of multiscroll memristive Hopfield neural network with adjustable memductance and application to image encryption,” *IEEE Transactions on Neural Networks and Learning Systems*, **34 (10)**, pp. 7824-7837, 2023.

[14] Lai, Q., Wan, Z. and Kuate, P.D.K“Generating grid multi-scroll attractors in memristive neural networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **70**, pp. 1324-1336, 2023.

[15] Lai, Q. and Yang, L. “Discrete memristor applied to construct neural networks with homogeneous and heterogeneous coexisting attractors,” *Chaos, Solitons and Fractals*, **174**, Article ID 113807, 2023.

[16] Lai, Q., Hua, H., Zhao, X.W. et al. “Image encryption using fission diffusion process and a new hyperchaotic map,” *Chaos, Solitons and Fractals*, **175**, Article ID 114022, 2023.

[17] Kengne, J., Njitacke, Z.T., Nguomkam Negou, A. et al. “Coexistence of Multiple Attractors and Crisis Route to Chaos in a Novel Chaotic Jerk Circuit,” *International Journal of Bifurcation and Chaos*,**26(5)**, Article ID1650081, 2016.

[18] Joshi, M. and Ranjan, A. “An autonomous simple chaotic jerk system with stable and unstable equilibria using reverse sine hyperbolic functions,” *International Journal of Bifurcation and Chaos*,**30(05)**, Article ID 2050070, 2020.

[19] Volos, C., Akgul, A., Pham, V.T. et al. “A simple chaotic circuit with a hyperbolic sine function and its use in a sound encryption scheme,” *Nonlinear Dynamics*,**89**, pp. 1047–1061,2017.

[20] Hu, X., Sang, B. and Wang, N. “The chaotic mechanisms in some jerk systems,” *AIMS Mathematics*,**7(9)**, pp. 15714–15740, 2022.

[21] Liu, J., Sprott, J.C., Wang, S. et al., “Simplest chaotic system with a hyperbolic sine and its applications in DCSK scheme,” *IET Communications*, **12(7)**, pp. 809–815,2018.

[22] Vaidyanathan, S., Volos, C., Pham, V.T. et al. “Adaptive backstepping control, synchronization and circuit simulation of a 3-D novel jerk chaotic system with two hyperbolic sinusoidal nonlinearities,” *Archives of Control Sciences*, **24(3)**, pp. 375-403, 2014.

[23] Sun, F., Lv, Z. and Wang, C. “Pseudo-Random Number Generator Based on Generalized Spatial Surface Chaotic System,” pp. 1–27, 2023. Available at http://dx.doi.org/10.2139/ssrn.4414973.

[24] Gong, L.H., Luo, H.X., Wu, R.Q. et al. “New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG,” *Physica A: Staistical Mechanics and Its Applications*, **591**, p. 126793, 2022.

[25] Adhikari, S. and Karforma, S. “An Efficient Image Encryption Method Using Henon-Logistic-Tent Chaotic Pseudo Random Number Sequence,” *Wireless Personal Communications*, **129(4)**, pp. 2843–2859, 2023.

[26] Mondal, B., Singh, S. and Kumar, P. “A secure image encryption scheme based on cellular automata and chaotic skew tent map,” *Journal of Information Security and Applications*, **45**, pp. 117–130, 2019.

[27] Çavuşoğlu, Ü., Panahi, S., Akgül, A., et al. “A new chaotic system with hidden attractor and its engineering applications: analog circuit realization and image encryption,” *Analog Integrated Circuits and Signal Processessing*, **98(1)**, pp. 85–99, 2019.

[28] Ismail, S. M., Said, L.A., Rezk, A.A.et al. “Image encryption based on double-humped and delayed logistic maps for biomedical applications,” *2017* 6th *International Conference on Modern Circuits and Systems Technologies. MOCAST 2017*, pp. 6–9, 2017.

[29] Li.,C. and Sprott, C. “Amplitude control approach for chaotic signals,” *Nonlinear Dynamics*, **73**, pp. 1335–1341, 2013.

[30] Xu, Q., Huang, L., Wang, N. et al. “Initial-offset-boosted coexisting hyperchaos in a 2D memristive Chialvo neuron map and its application in image encryption,” *Nonlinear Dynamics*, **111**, pp. 20447-20463, 2023.

[31] Chen, X., Wang, N., Wang, Y. et al. “Memristor initial-offset boosting and its bifurcation mechanism in a memristive FitzHugh-Nagumo neuron model with hidden dynamics,” *Chaos, Solitons and Fractals*,**174**, Article ID 113836, 2023.

[32] Bassham, L., Rukhin, A., Soto, J. et al., “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications”, *Special Publication (NIST SP), National Institute of Standards and Technology*, Gaithersburg, MD, 2010. [Online]. Available:https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762

[33] Sun, F. and Liu, S. “Cryptographic pseudo-random sequence from the spatial chaotic map,” *Chaos, Solitons and Fractals*, **41(5)**, pp. 2216–2219, 2009.

[34] Walker, J. “ENT: a pseudorandom number sequence test program.” 2008. [Online]. Available: https://www.fourmilab.ch/random/

[35] Bisong, E. “Google Colaboratory BT - Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners,” E. Bisong, Ed. Berkeley, CA: Apress, 2019, pp. 59–64. doi: 10.1007/978-1-4842-4470-8_7.

[36] Akgul, A., Moroz, I., Pehlivan, I. et al. “A new four-scroll chaotic attractor and its engineering applications,” *Optik*,**127(13)**, pp. 5491–5499, 2016.

[37] Zhu, S., Deng, X., Zhang, W. et al. “Secure image encryption scheme based on a new robust chaotic map and strong S-box,” *Mathematics and Computers in Simulation,* **207**, pp. 322–346, 2023.

[38] Liu, L. and Wang, J. “A cluster of 1D quadratic chaotic map and its applications in image encryption,” *Mathematics and Computers in Simulation,***204**, pp. 89–114, 2023.

[39] Elkhalil, N., Weddy, Y.C. and Ejbali, R. “Image encryption using the new two-dimensional Beta chaotic map,” *Multimedia Tools and Applications,* **82**, pp. 31575–31589, 2023.

[40] Gupta, M., Singh, V.P., Gupta, K.K. et al. “An efficient image encryption technique based on two-level security for internet of things,” *Multimedia Tools and Applications,* **82(4)**, pp. 5091–5111, 2023.

[41] Man, Z., Li, J., Di, X. et al. “A novel image encryption algorithm based on least squares generative adversarial network random number generator,” *Multimedia Tools and Applications*, **80(18)**, pp. 27445–27469, 2021.

[42] Maddodi, G., Awad, A., Awad, D. et al. “A new image encryption algorithm based on heterogeneous chaotic neural network generator and dna encoding,” *Multimedia Tools and Applications*, **77(19)**, pp. 24701–24725, 2018.

[43] Ogras, H. and Tur, M.R. “An Effective Image Encryption Algorithm Using Bit Reversal Permutation and a New Chaotic Map,” *Gazi University Journal of Science,* **35(2)**, pp. 542–556, 2022.

[44] Njitacke, Z.T., Nkapkop, J.D.D., Signing, V.F. et al. “Novel extreme multistable Tabu learning neuron: circuit implementation and application to cryptography,” *IEEE Transactions on Industrial Informatics*,**19(8),** pp. 8943-8952, 2023.

[45] Lai, Q., Hua, H., Zhao, X.W. et al. “Image encryption using fission diffusion process and a new hyperchaotic map,” *Chaos, Solitons and Fractals,* **175,** Article ID 114022, 2023.

[46] Praveenkumar, P., Amirtharajan, R., Thenmozhi, K. et al. “Pixel scattering matrix formalism for image encryption-A key scheduled substitution and diffusion approach,” *AEU - International Journal of Electronics and Communications,* **69(2)**, pp. 562-572, 2015.

[47] Velliangiri, S., Krishna Lavakumar, G. and Sathya, K. “Image Encryption using Chaotic Sorting Fortified with DNA Sequencing,” *8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, pp. 352-357, 2022.

[48] Thomas, M.Y.S., Krishna, V.N.A. and Varghese, M.B. “Image Encryption Algorithm with Block Scrambling Based on Logistic Map,” *Indian Journal of Science and Technology*,**16(14)**, pp. 1045–1055, 2023.

[49] Elkhalil, N., Zahmoul, R., Ejbali, R. et al, “A Joint Encryption-Compression Technique for Images Based on Beta Chaotic Maps and SPIHT Coding,” *ICSEA 2019: The Fourteenth International Conference on Software Engineering Advances,* ISBN: 978-1-61208-752-8, pp. 118–122, 2019.

[50] Zhou, G., Zhang, D., Liu, Y. et al. “A novel image encryption algorithm based on chaos and Line map,” *Neurocomputing,* **169**, pp. 150–157, 2015.

[51] Wang, X.Y., Zhang, Y.Q. and Bao, W.M. “A novel chaotic image encryption scheme using DNA sequence operations”, *Optics and Lasers in Engineering.***73**, 53–61, 2015.

[52] Som, S. and Kotal, A. “Confusion and diffusion of grayscale images using multiple chaotic maps,” *2012 National Conference on Computing and Communication Systems,* Durgapur, India, pp. 1-5, 2012.

[53] Zhu, H., Zhao, C., Zhang, X. et al. “An image encryption scheme using generalized Arnold map and affine cipher,” *Optik*,**125**, pp. 6672-6677, 2014.

[54] Chen, J.X., Zhu, Z.L., Fu, C. et al. “A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism,” *Communications in Nonlinear Science Numerical Simulation,* **20**, pp. 846-860, 2015.

[55] Xu, Q., Wang, K., Shan, Y. et al. “Dynamical effects of memristive electromagnetic induction on a 2D Wilson neuron model,” *Cognitive Neurodynamics,* 2023.

[56] Xu, Q., Wang, Y., Iu, H.H.C. et al. “Locally active memristor-based neuromorphic circuit: Firing pattern and hardware experiment,”*IEEE Transactions on Circuits and Systems I: Regular Papers,* **70 (8),** pp. 3130-3141, 2023.

Biographies

Rameshbabu Ramar received the Ph.D degree in electronics and communication engineering from St.Peter’s university, Chennai, Tamilnadu, India in 2021. He is currently a Associate Professor, V.S.B. Engineering college, Karur, Tamilnadu, India. He has published many research paper in international journals. His current research interest includes nonlinear dynamical systems, chaotic and hyperchaotic systems, and chaos synchronization.

Sundarapandian Vaidyanathan received the D.Sc. degree in electrical and systems engineering from Washington University in St. Louis, St. Louis, MO, USA, in 1996. He is currently a Professor and the Dean of the Research and Development Centre, Vel Tech University, Chennai, India. He has published over 600 Scopus-indexed research publications. He has delivered many Keynote Addresses in Control Systems, Chaos Theory, Data Science and Scientific Modelling in International Conferences. His current research interests include linear and nonlinear control systems, chaotic and hyperchaotic systems, data science, circuits, intelligent control, optimal control, mathematical modelling, and scientific computing.

Akif Akgul received the B.Sc. degree in electronics-computer education from Kocaeli University, in 2009, the B.Sc. degree in electrical-electronics engineering from Sakarya University, in 2013, and the M.S. and Ph.D. degrees in electronics computer education and electrical-electronics engineering from Sakarya University, in 2011 and 2015, respectively.

He joined the Institute of Electronics, Communications and Information Technology (ECIT), Queen’s University Belfast, U.K., as a Visiting Researcher, in 2015. His current interests include analog electronics, chaos theory, chaotic systems, chaos-based engineering applications (cryptography, steganography, and pseudo and true random number generators), experimental chaotic synchronization, analysis and design of analog circuits, and microcomputer-based applications.

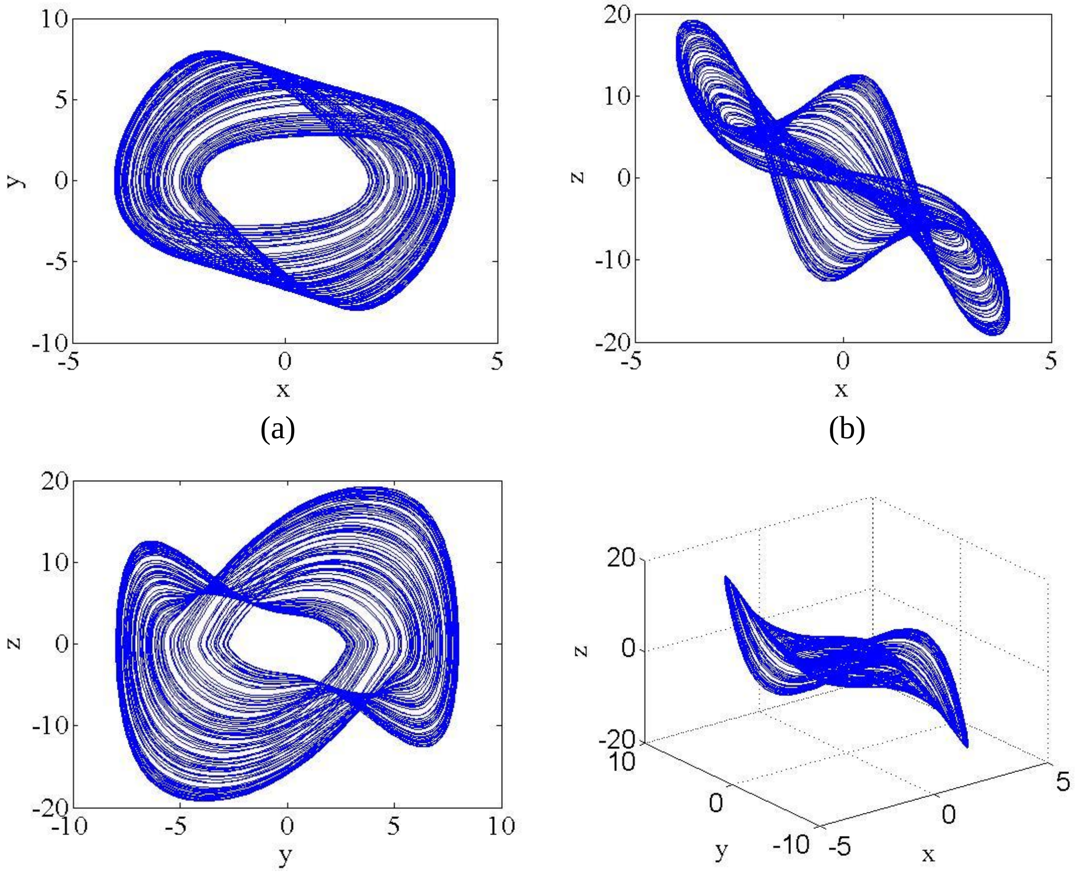
Berkay Emin received the B.Sc. degree in electrical and electronics engineering from Bozok University, in 2017, and the M.Sc. degree in electrical and electronics engineering from Nevsehir Haci Bektas Veli University, in 2019. He is currently pursuing the Ph.D. degree at Yozgat Bozok University. He has been a lecturer at Hitit University Osmancık Ömer Derindere Vocational School. His research interests include chaotic systems, cryptology and information security.

List of Figure Captions:

15. **Figure 15** Correlation analysis (a) Horizontal correlation of source image; (b) Horizontal correlation of encrypted image; (c) Vertical correlation of source image; (d) Vertical correlation of encrypted image; (e) Diagonal correlation of source image; (f)Diagonal correlation of encrypted image
16. **Figure 16** Key sensitivity test in the encrypted stage(a) source image (b) image encrypted with the correct key (c) image encrypted by modifying x0 + t
17. **Figure 17** Key sensitivity test in the decrypted stage(a) encrypted image (b) image decrypted with the correct key (c) image decrypted by modifying x0+ t
18. **Figure 18** Results of noise interference experiments: (a) with 0.01 Salt & Pepper Noise; (b) with 0.05 Salt & Pepper Noise; (c) with 0.1 Salt & Pepper Noise; (d,e,f) are the decrypted images of (a,b,c), respectively
19. **Figure 19** Results of clipping experiments: (a-b) 25% data loss; (c) 50% data loss; (d,e,f) Decrypted images of (a,b,c), respectively

List of Table Captions:

1. **Table 1:** Comparison of the proposed system with existing systems which have hyperbolic sine nonlinearity
2. **Table 2:** Equilibrium points and stability of the new system (7)
3. **Table 3:** NIST-800-22 test results of x phase random numbers from 10 sequences
4. **Table 4:** Random numbers FIPS 140-1 success criterions and test results
5. **Table 5:** ENT test results of random numbers
6. **Table 6:** Correlation coefficient of the source image and encrypted images
7. **Table 7:** Encrypted images NPCR and UACI analysis
8. **Table 8:** Entropy of source image and encrypted images
9. **Table 9:** The PSNR of decrypted image (unit dB)



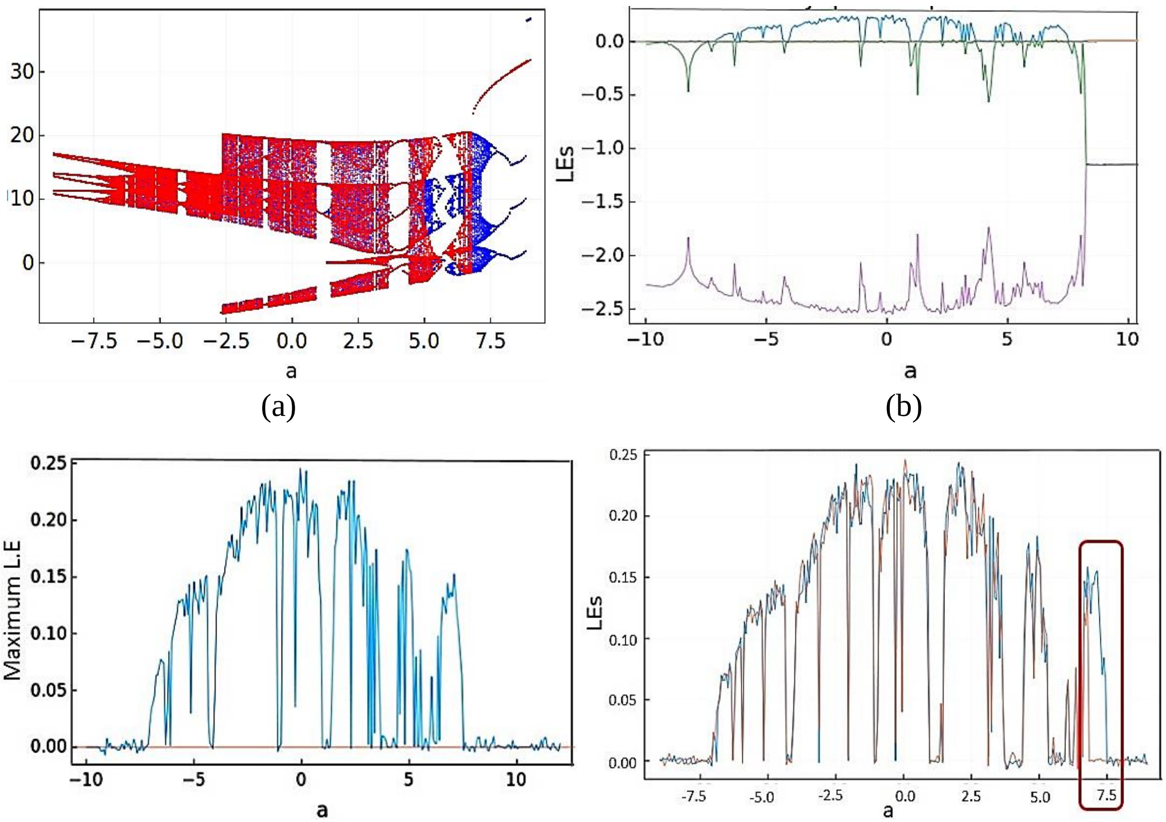
(a)

(b)

(c)

(d)

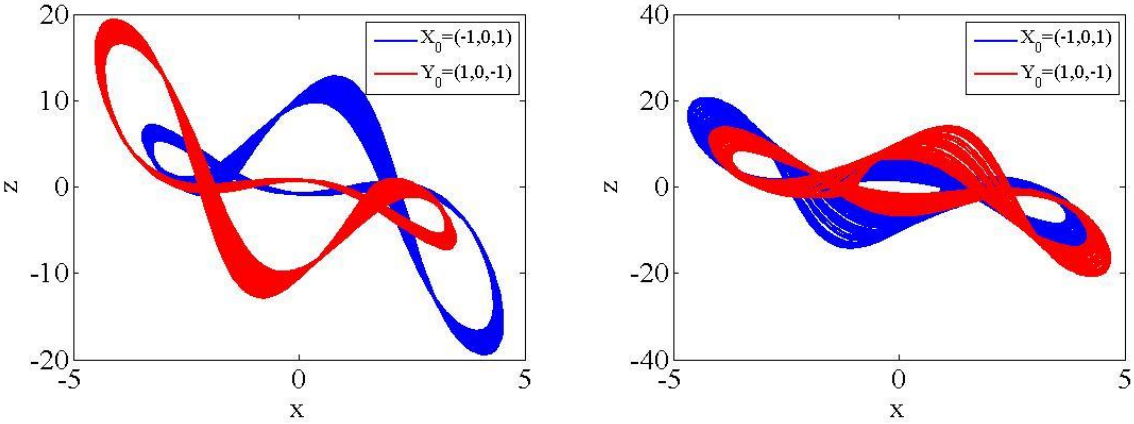
Figure 1. Attractors of the proposed system (7) with the initial conditions (1,0,-1). (a) *xy* plane, (b) *xz* plane, (c) *yz* plane and (d) *xyz* plane



(a) (b)

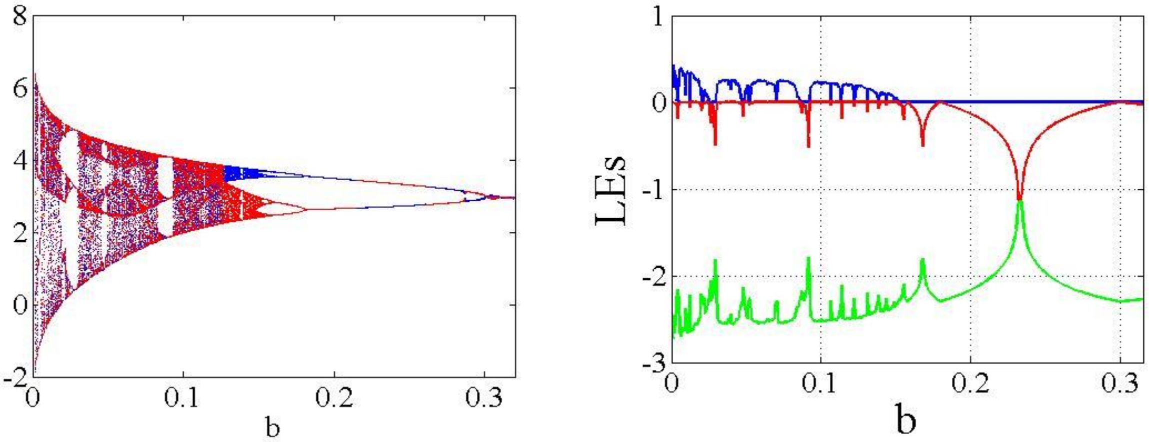
(c) (d)

Figure 2. (a) Bifurcation diagram with Forward continuation (Blue), backward continuation (Red), (b) Lyapunov exponent plot, (c) Maximum Lyapunov exponent Plot, (d) Maximum Lyapunov exponent plot with Forward continuation (Blue), backward continuation (Red) for Parameter "*a*" variation



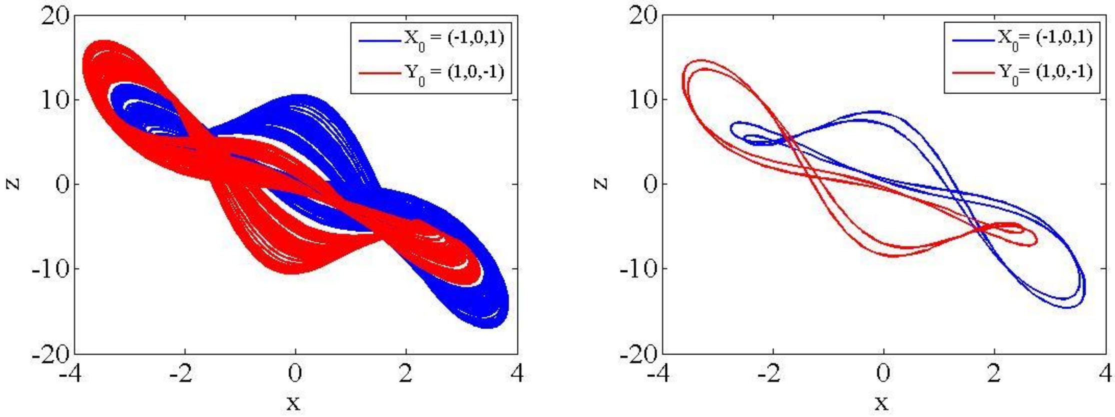
(a) *a* = 5.5 (b) *a* = 6.5

Figure 3. (a) Coexisting periodic attractors, (b) Coexisting chaotic attractors for various values of the parameter *a*



(a) (b)

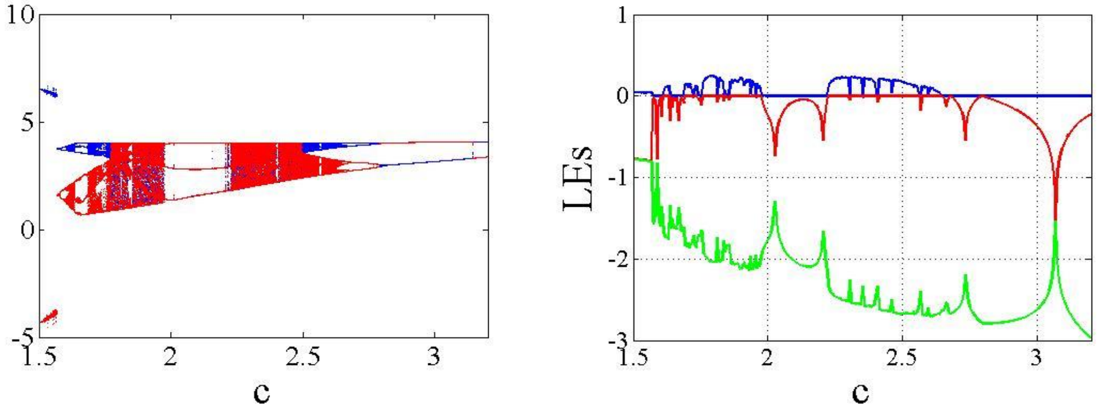
Figure 4. (a) Bifurcation diagram with initial condition (-1,0,1) (Blue) and (1,0,-1) (Red) and (b) Lyapunov spectrum for the parameter *b* variation



(a) *b* = 0.13

(b) *b* = 0.17

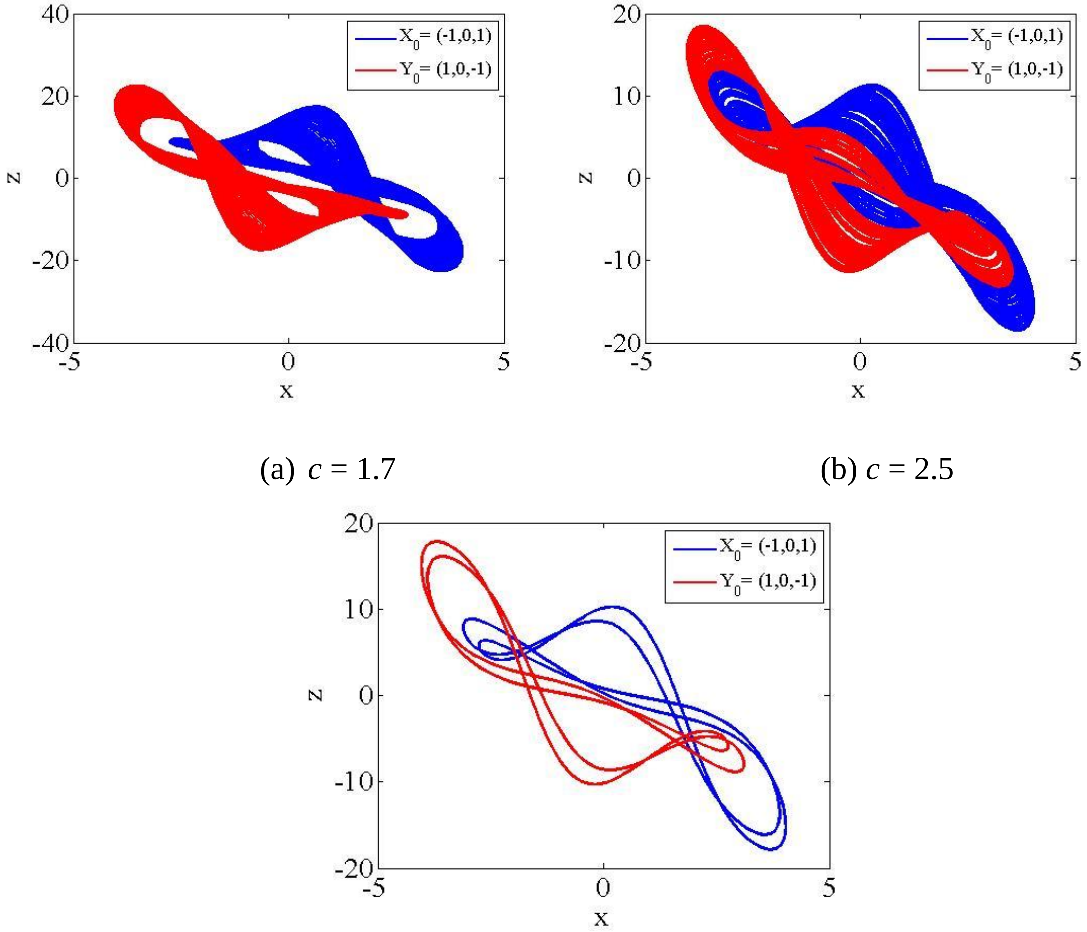
Figure 5 (a) Chaotic coexisting attractor (b) Periodic coexisting attractor for the various values of the parameter *b*.



(a)

(b)

Figure 6 (a) Bifurcation diagram with initial condition (-1,0,1) (Blue) and (1,0,-1) (Red) and (b) Lyapunov spectrum for the parameter *c* variation

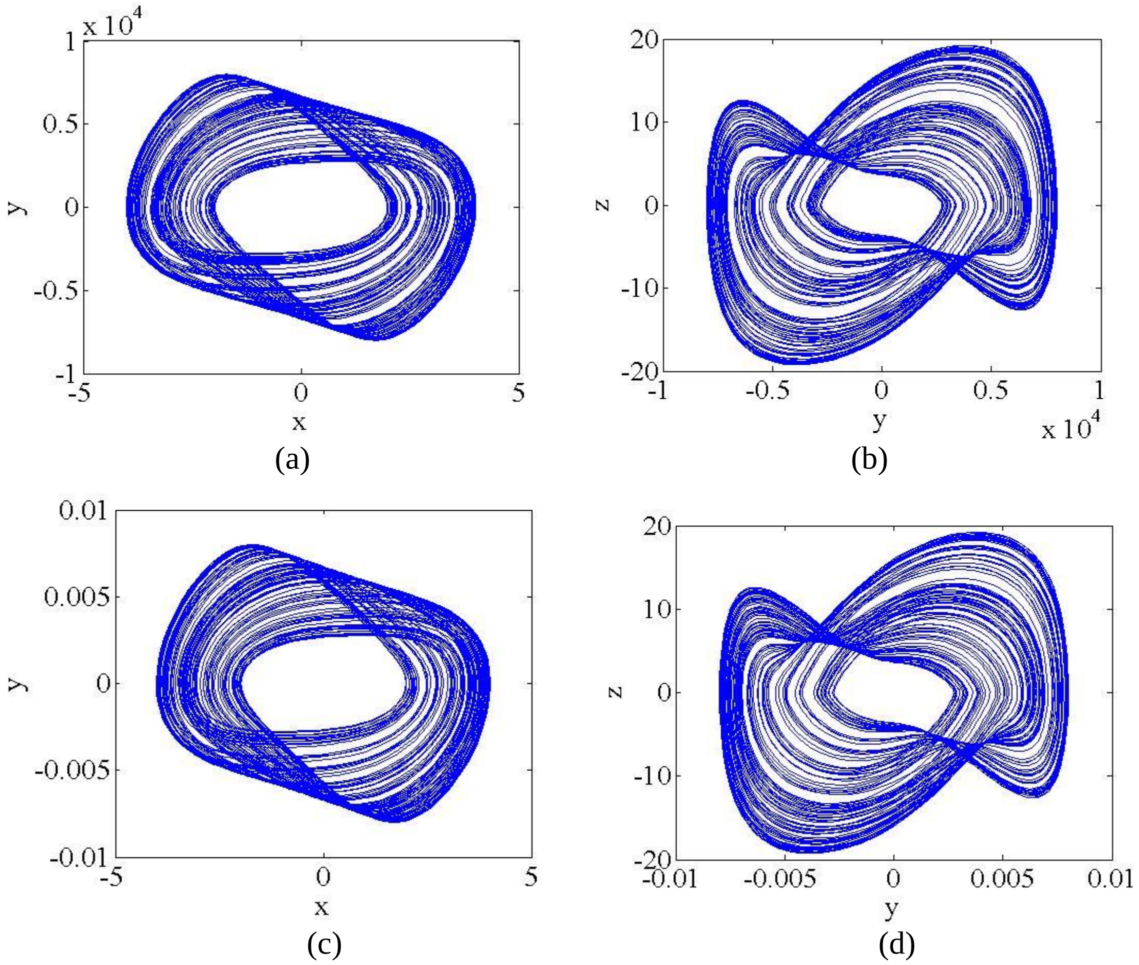


(a) $c = 1.7$

(b) $c = 2.5$

(c) $c = 2.7$

Figure 7. (a-b) Chaotic coexisting attractors (c) Periodic coexisting attractors for the variation of parameter *c*



(a) (b)

(c) (d)

Figure 8. Partially amplitude-controlled attractors along y direction. (a-b) when $\alpha = 0.001$ and (c-d) when $\alpha = 1000$.

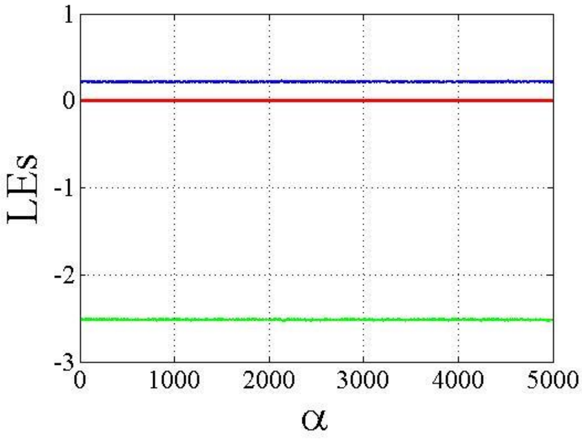


Figure 9. Constant Lyapunov exponent spectrum of the system (16) for the variation of control parameter $\alpha \in [0, 5000]$.

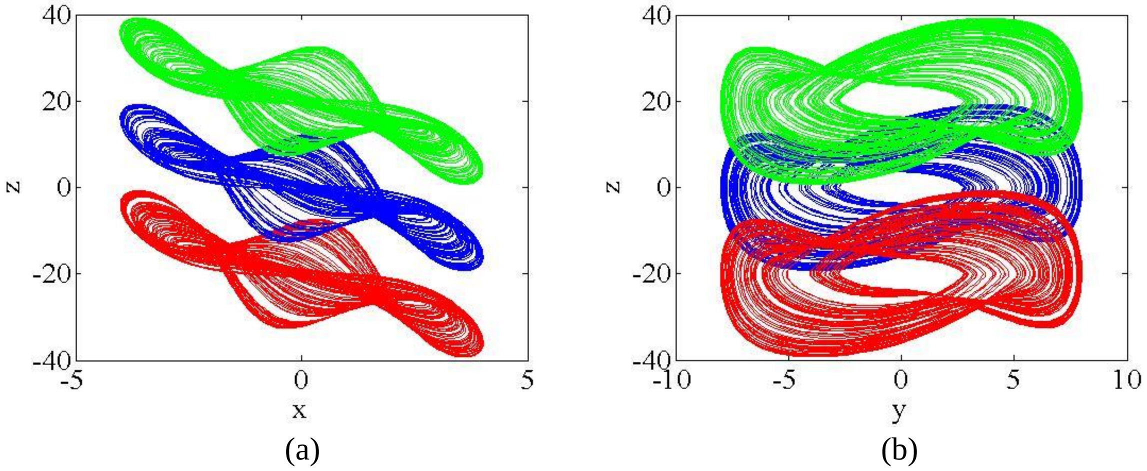


Figure 10. Offset boosting-controlled attractors along the *z* direction with $\beta = 1$ (blue), $\beta = 20$ (red) and $\beta = -20$ (green)

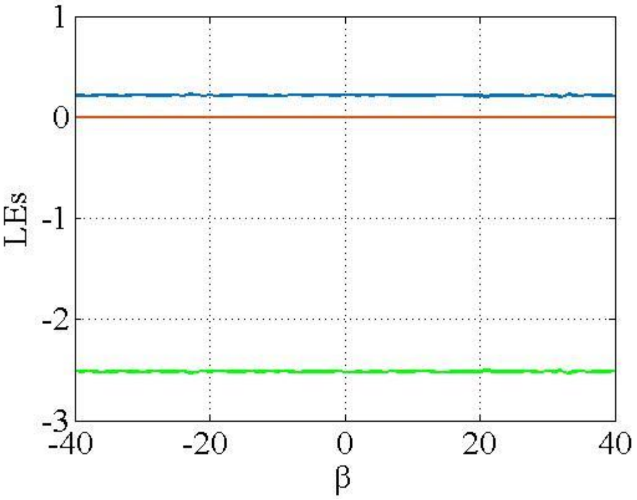
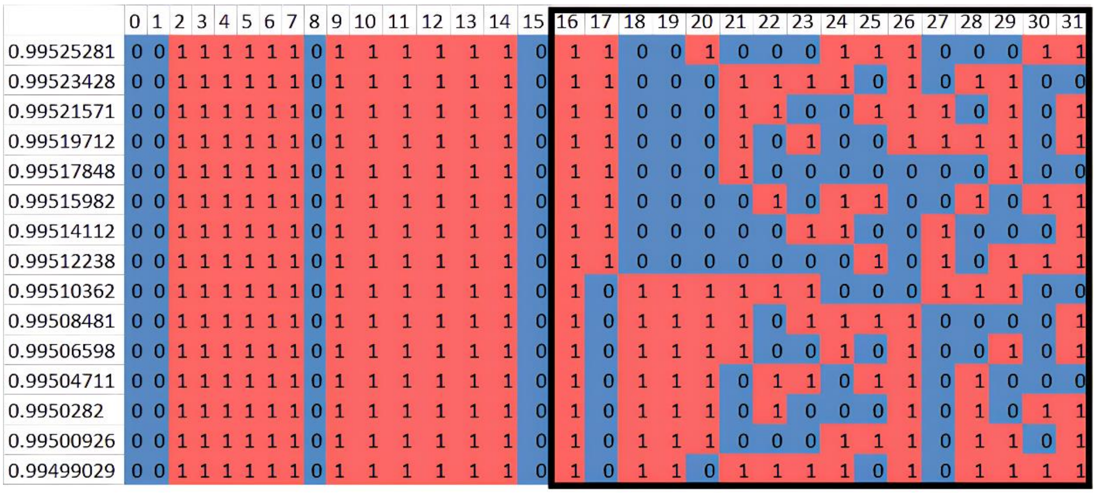
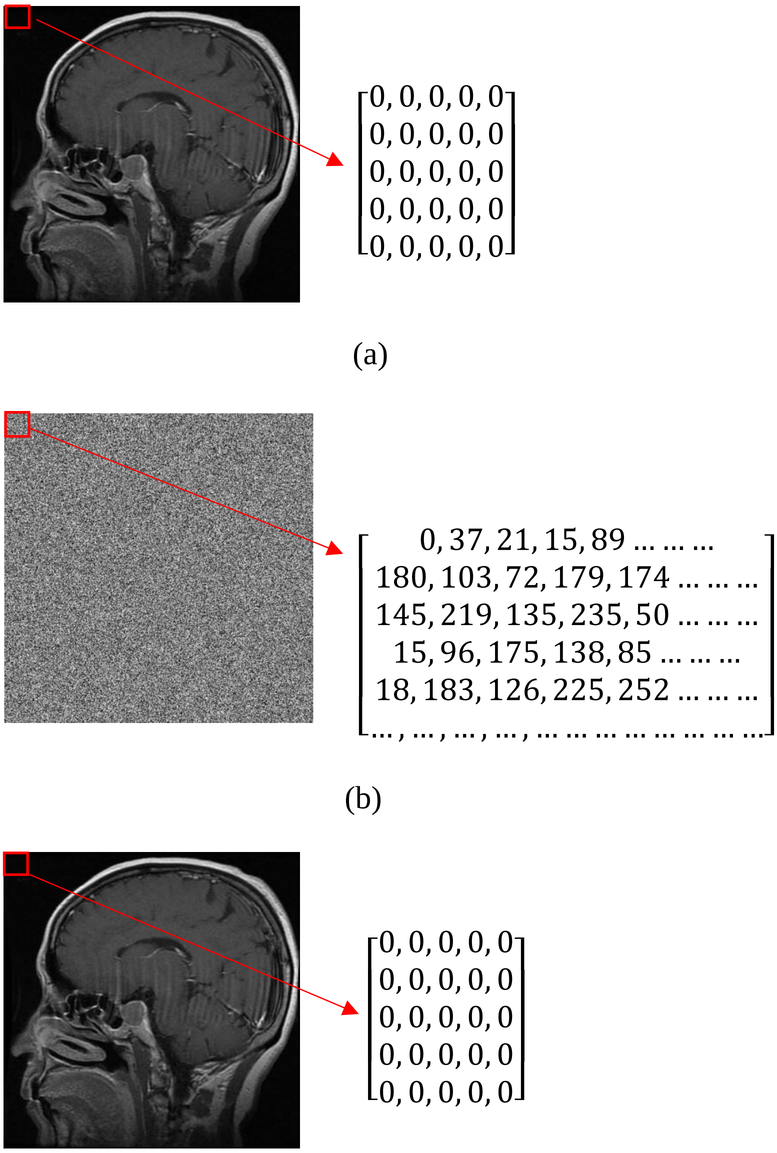


Figure 11. Constant Lyapunov spectrum of the system (18) for the variation of booster parameter $\beta \in [-40, 40]$



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0.99525281	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	1	1
0.99523428	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	1	0	1	1	0	0
0.99521571	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	0	1	0	1
0.99519712	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	0	0	1	1	1	1	0	1
0.99517848	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0.99515982	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	0	1	0	1	1	0	0	1	0	1	1
0.99514112	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1	1	0	0	1	0	0	0	1
0.99512238	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1
0.99510362	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0	0	1	1	1	0	0
0.99508481	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0	1
0.99506598	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1
0.99504711	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0
0.9950282	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	1
0.99500926	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	0	0	1	1	1	0	1	1	0	1
0.99499029	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	0	1	1	1	1

Figure 12 Conversion of float numbers binary number format.



$$\begin{bmatrix} 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \end{bmatrix}$$

(a)

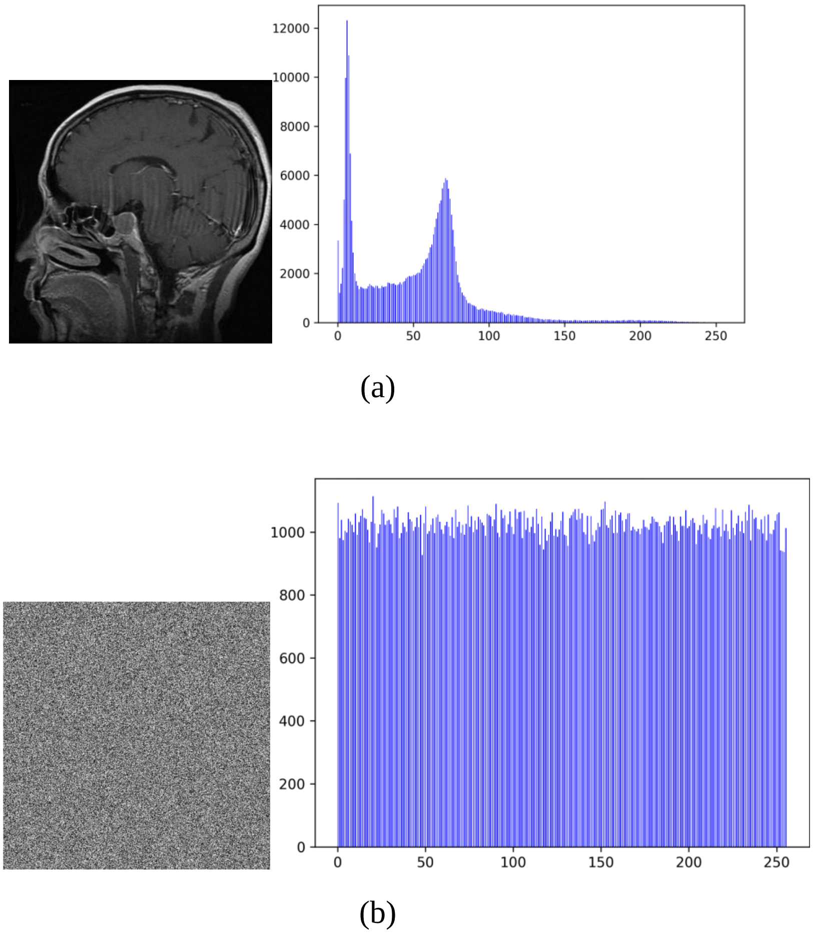
$$\begin{bmatrix} 0,37,21,15,89\ \ldots\ \ldots\ \ldots \\ 180,103,72,179,174\ \ldots\ \ldots\ \ldots \\ 145,219,135,235,50\ \ldots\ \ldots\ \ldots \\ 15,96,175,138,85\ \ldots\ \ldots\ \ldots \\ 18,183,126,225,252\ \ldots\ \ldots\ \ldots \\ \ldots,\ldots,\ldots,\ldots,\ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \end{bmatrix}$$

(b)

$$\begin{bmatrix} 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \end{bmatrix}$$

(c)

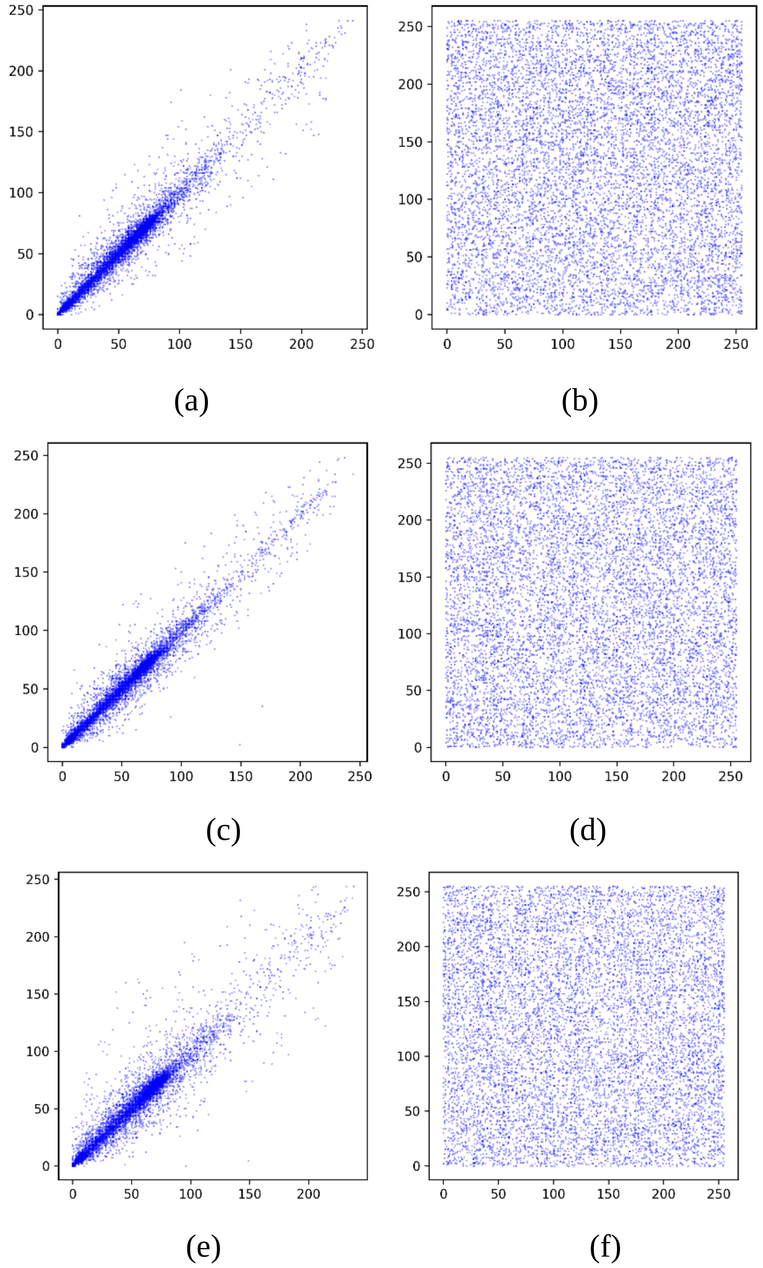
Figure 13 Matrix values; a) source image b) encrypted image c) decrypted image



(a)

(b)

Figure 14. Histogram analysis: (a) Source image; (b) Encrypted image

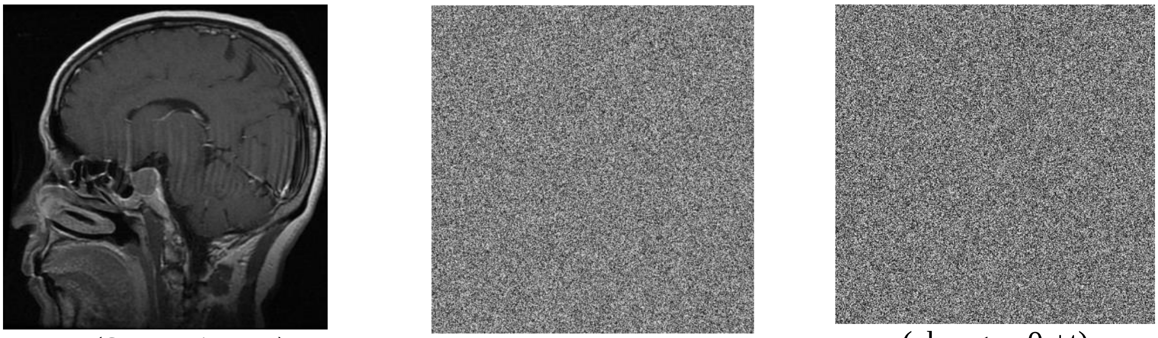


(a) (b)

(c) (d)

(e) (f)

Figure 15 Correlation analysis (a) Horizontal correlation of source image; (b) Horizontal correlation of encrypted image; (c) Vertical correlation of source image; (d) Vertical correlation of encrypted image; (e) Diagonal correlation of source image; (f) Diagonal correlation of encrypted image

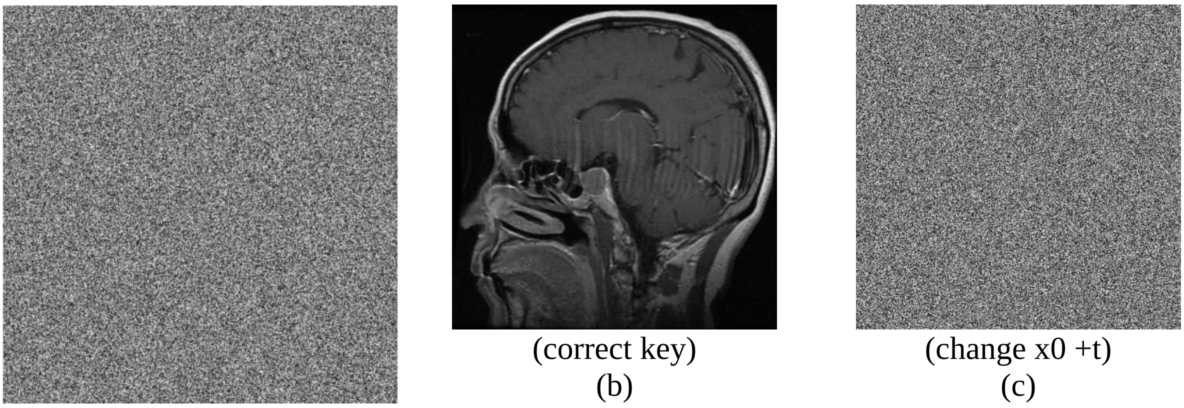


(Source image)
(a)

(correct key)
(b)

(change x0 +t)
(c)

Figure 16. Key sensitivity test in the encrypted stage(a) source image (b) image encrypted with the correct key (c) image encrypted by modifying x0 + t



(encrypted image)
(a)

(correct key)
(b)

(change x0 +t)
(c)

Figure 17. Key sensitivity test in the decrypted stage(a) encrypted image (b) image decrypted with the correct key (c) image decrypted by modifying x0+ t

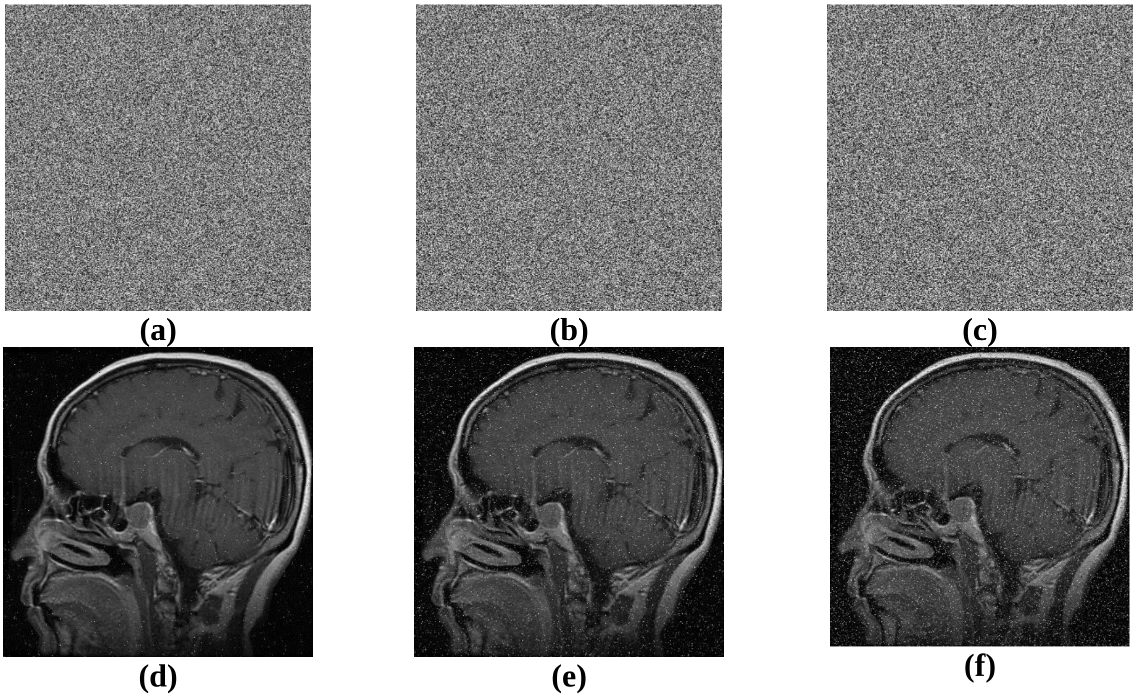


Figure 18. Results of noise interference experiments: (a) with 0.01 Salt & Pepper Noise; (b) with 0.05 Salt & Pepper Noise; (c) with 0.1 Salt & Pepper Noise; (d,e,f) are the decrypted images of (a,b,c), respectively.

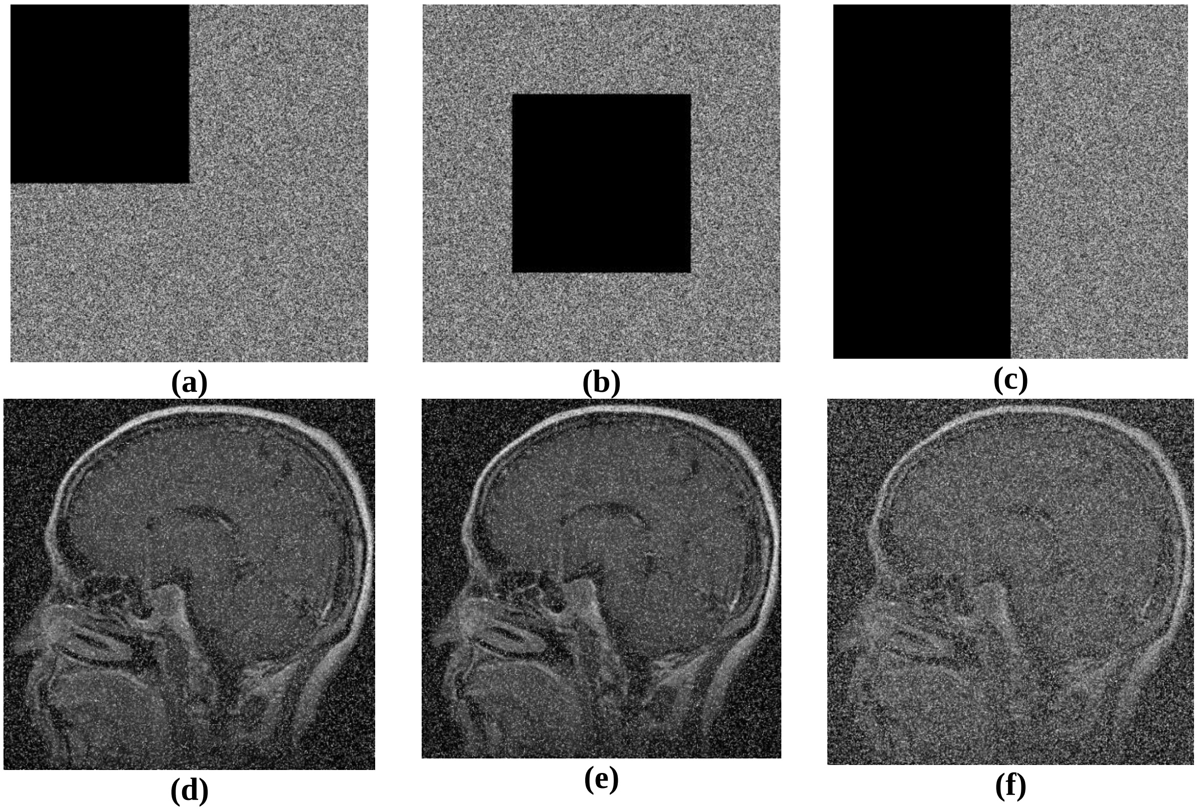


Figure 19 Results of clipping experiments: (a-b) 25% data loss; (c) 50% data loss; (d,e,f) Decrypted images of (a,b,c), respectively

Table 1: Comparison of the proposed system with existing systems which have hyperbolic sine nonlinearity

S.No	Existing systems	*LE₁*	Coexisting attractors	Amplitude control	Offset boosting
1	Kengne et al. [17]	-	Yes	-	-
2	Joshi and Ranjan [18]	0.037	-	-	-
3	Volos et al. [19]	0.21244	Yes	-	-
4	Hu et al. [20]	0.1071	Yes	-	-
5	Liu et al. [21]	0.1652	-	-	-
6	Sundarapandian et al. [22]	0.0777	-	-	-
	Proposed system	**0.21613**	**Yes**	**Yes**	**Yes**

Table 2: Equilibrium points and stability of the new system (7)

Equilibrium points	Eigen values	Nature of stability
$E_1 = [0, 0, 0]$	$\lambda_1 = -2.192, \lambda_2 = 0.428, \lambda_3 = 0.3198$	Unstable node
$E_{2,3} = [\pm\sqrt{a}, 0, 0]$	$\lambda_1 = -2.365, \lambda_{2,3} = 0.0325 \pm j0.503$	Saddle unstable point

Table 3 NIST-800-22 test results of x phase random numbers from 10 sequences

Statistical Tests	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8	Seq 9	Seq 10	Encrypted Image	Result
Frequency (Monobit) Test	0.8041	0.1591	0.1275	0.4248	0.9076	0.5961	0.8212	0.7413	0.7383	0.5538	0.0929	Pass
Block-Frequency Test	0.1570	0.1286	0.2746	0.8754	0.5854	0.2228	0.3539	0.2573	0.7416	0.8389	0.4374	Pass
Cumulative-Sums Test	0.8894	0.1738	0.2317	0.6825	0.6146	0.5385	0.8916	0.6662	0.6976	0.7016	0.1205	Pass
Runs Test	0.7308	0.2601	0.8710	0.1607	0.7702	0.1790	0.1942	0.1713	0.2730	0.0763	0.1429	Pass
Longest-Run Test	0.9395	0.6493	0.1063	0.3100	0.7807	0.6888	0.6700	0.2763	0.7739	0.8500	0.1146	Pass
Binary Matrix Rank Test	0.7310	0.1959	0.1274	0.7465	0.4039	0.0842	0.3356	0.1203	0.5736	0.1564	0.7310	Pass
Discrete Fourier Transform Test	0.2829	0.6073	0.2829	0.6397	0.5756	0.4684	0.4628	0.1371	0.5882	0.6397	0.1323	Pass
Non-Overlapping Templates Test	0.9914	0.9944	0.7974	0.7618	0.8648	0.3662	0.6242	0.0781	0.2470	0.2032	0.2578	Pass
Overlapping Templates Test	0.2149	0.5010	0.6205	0.2283	0.2615	0.8065	0.9923	0.2132	0.8860	0.3141	0.2092	Pass
Maurer's Universal Statistical Test	0.7277	0.3075	0.4457	0.9653	0.9956	0.9419	0.8100	0.8893	0.5377	0.0867	0.6756	Pass
Approximate Entropy Test	0.4431	0.3736	0.2020	0.1760	0.8405	0.6821	0.3827	0.0895	0.3959	0.7635	0.0944	Pass
Random-Excursions Test (x = −4)	0.7056	0.5675	0.9732	0.7875	0.0958	0.9902	0.8065	0.5146	0.2276	0.3991	0.8648	Pass
Random-Excursions Variant Test	0.4358	0.6853	0.7962	0.9159	0.3131	0.2603	0.6055	0.8259	0.6389	0.6159	0.2403	Pass

(x = −9)												
Serial Test-1	**0.9152**	**0.3745**	**0.0809**	**0.0761**	**0.7408**	**0.0126**	**0.9151**	**0.8542**	**0.2892**	**0.4853**	**0.7652**	**Pass**
Serial Test-2	**0.9111**	**0.3385**	**0.5328**	**0.5469**	**0.6551**	**0.0906**	**0.9892**	**0.6729**	**0.1449**	**0.2555**	**0.5562**	**Pass**
Linear-Complexity Test	**0.6135**	**0.9185**	**0.8799**	**0.4900**	**0.1695**	**0.6966**	**0.2892**	**0.7874**	**0.2061**	**0.7432**	**0.5526**	**Pass**

Table 4 Random numbers FIPS 140-1 success criterions and test results

FIPS 140-1 Tests	Success Criterions	Value	Result
Monobit Test Poker	9654 < x < 10346	10036	Pass
Poker Test	1.03 < x < 57.4	10.0223	Pass
Run Test (1)	2267 ≤ x ≤ 2733	2432	Pass
Run Test (2)	1079 ≤ x ≤ 1421	1331	Pass
Run Test (3)	502 ≤ x ≤ 748	633	Pass
Run Test (4)	223 ≤ x ≤ 402	289	Pass
Run Test (5)	90 ≤ x ≤ 223	147	Pass
Long Run Test	34 > Run	3	Pass

Table 5 ENT test results of random numbers

Test name	Average	Ideal Results	Result
Arithmetic Mean	127.4652	127,5	Pass
Entropy	7.9985	8	Pass
Correlation	-0.0027159	0	Pass
Chi-Square	256.2657	10% and 90% between	Pass
Monte Carlo	3.1446 (error =0.0009)	Pi Number	Pass

Table 6 Correlation coefficient of the source image and encrypted images

Image	Horizontal Correlation	Vertical Correlation	Diagonal Correlation
Source Image	0.9792	0.9815	0.9591
Our Encrypted Image	-0.0041	-0.0053	-2.7e-04
Man et al. [41]	−0.0113	0.0056	−0.0004
Maddodi et al. [42]	0.0058	0.0072	0.0031
Ogras et al. [43]	-0.0468	-0.0026	0.0149
Njitacke et al. [44]	0.0081	-0.0041	0.0107
Lai et al. [45]	−0.0089	0.0097	0.0060

Table 7 Encrypted images NPCR and UACI analysis.

Image	NPCR	UACI
Our Encrypted Image	99.5868	33.5302
Njitacke et al. [44]	99.73	33.5765
Lai et al. [45]	99.6081	33.4578
Velliangiri et al. [47]	99.6184	33.42
Thomas et al. [48]	99.5571	33.5995
Elkhalil et al. [49]	99.5666	33.3384

Table 8 Entropy of source image and encrypted images

Image	Entropy
Source Image	6.6491
Our Encrypted Image	7.9993
Man et al. [41]	7.9975
Njitacke et al. [44]	7.9980
Lai et al. [45]	7.9992
Velliangiri et al. [47]	7.9651
Som et al. [52]	7.9975

Table 9. The PSNR of decrypted image (unit dB)

Salt & peppers intensity			Data loss		
0.01	0.05	0.1	%25 corner	%25 middle	%50 left
26.9063	19.9883	17.0891	13.0849	13.0909	10.0825