مارس 2024م March 2024 ISSN : 7848-2958 pp. 230 - 257 TURJ العدد (39) Issue (39) مجلة بحوث جامعة تعز Taiz University Research Journal

A Colour Video Encryption Scheme Based on Chaotic Maps

Dr. Abdullah Jaafar [1] and Ahmed Abdulqader Mohammed [1,2]
[1] Computer Science Department, Faculty of Computers and Information Technology (Turba Branch), University of Taiz, Taiz, Yemen
[2] Computer Science Department, Faculty of Applied Sciences, University of Taiz, Taiz, Yemen
dr.abdullahjaafar@yahoo.com [1], ahmed772338911@gmail.com [2]

Abstract

The development of wireless and mobile communications has led to the expansion of digital multimedia (video and image) exchange over computer networks. So, there is a need to protect video content in many digital applications such as confidential video conferencing, medical imaging systems and payTV, etc. Video encryption is widely used as a way to provide security for digital video. In this paper, we introduce a secure scheme for colour video encryption using chaotic maps. Video encryption divides the file into a group of images called frames, and then the video encryption process goes through five stages: shuffle the video frames, permutation, confusion, diffusion on frame level and diffusion on video space components level. The proposed scheme uses four chaotic maps to shuffle, permutation, confuse, and diffuse the video frames. It also uses two other chaotic maps to initialize and control the above chaotic maps. The key space of our scheme is 2^{128} which makes the brute force attack impracticable. To analyze security and robustness of the proposed scheme, several security tests are used such as peak signal-to-noise ratio (PSNR), signal to noise ratio (SNR), histogram analysis, correlation coefficient analysis, information entropy, differential attacks, keyspace, and key sensitivity. The results of the different types of analyses indicate that the proposed video encryption scheme has high sensitivity and

security.

Keywords: Video Encryption, Chaos Encryption, Chaotic Maps, Symmetrical Encryption, Information security.

1 Introduction

The development of wireless and mobile communications has led to the expansion of digital multimedia (video and image) exchange in many digital applications such as confidential video conferencing, medical imaging systems and pay-TV, etc. But, such applications encounter many security threats such as the unauthorized listening and illegal use over shared and open networks. Therefore, it has become necessary to protect sensitive video content from such threats. Video encryption is an approach to protect the sensitive video content [1]. Encryption of digital video is more complex and difficult than text encryption due to some intrinsic features of videos such as bulk data capacity and high redundancy, which are generally difficult to handle by traditional methods [2,3,4,5]. Chaotic maps have been used recently in cryptography for large scale data encryption such as text, image, video and audio data [5,6,7,8,9] et al, due to their strong properties such as its extreme sensitivity to initial conditions, unpredictability and random like behaviors [1,10].

In [2], the authors present a chaotic video encryption scheme (CVES). Chaotic maps are used to generate pseudo-random signal to mask the video, and to make pseudo-random permutation of the masked video. CVES encrypts video stream as frame by frame by XORing the data stream first and then substituted by a pseudo random s-box.

In [11], the authors present a video encrypting scheme based on the chaotic maps, the encrypting scheme uses the 2D cat map and the piecewise linear chaotic map together, these chaotic maps are used to generate key sequence in order to be used for encryption and to generate the permutation list. In order to provide video content protection, first, selective encryption is completed by encrypting (FLC), using chaotic stream cipher. Then, for each frame, macro block shuffling is executed in video bit stream, using chaos-based permutation.

In [1], the authors present video encryption based on chaotic system and stream cipher, the video encryption method is developed using chaotic system for key generator and stream cipher, it uses chaotic map as one-time key generator which produce key used for encryption process. Initial values of the key generator are randomly selected and set to all the equations of key generator. Key generator will generate key sequence as long as the video data. One-time pad stream cipher is performed on the data to be encrypted by xoring key sequence with the video data.

In [5], the authors present different scenarios from three combinations of algorithms from four types of chaotic maps (Arnold map, Lorenz map, Chebyshev map, Logistic map): The first one is two stages of encryption from two different types of chaotic maps. In this type, the signal becomes an input to the first stage of encryption. The encrypted signal output is the input to the second encryption stage. So, the signal is encrypted twice in a series manner by different types of chaotic algorithm. The second one is the sum of the encrypted signal by one chaotic map by the encryption key and the result is the final encrypted signal.

In [4], the authors present a colour video encryption/ decryption scheme based on hybrid chaotic maps. A video scrambling is to split the file into several I-frames, then shuffle the frames and confuse and diffuse the frame content. In this method, the confusion and diffusion procedures are integrated.

In [12], the authors present a chaotic video encryption scheme based on hyperchaotic system, the proposed encryption scheme uses different strength encryption algorithms for the Y channel (the brightness component), C_b channel (the blue chromaticity component), and C_r channel (the red chromaticity component). The Y channel is encrypted by the Arnold transformation to change the position of the pixel points in the original image, after Arnold encryption transformation, Y channel uses a DNA encoding algorithm. The C_b and C_r channels are encrypted by the Lorenz hyperchaotic map.

In [13], the authors present a novel chaotic-based encryption scheme using 1-D and 2-D iteration models of the video stream. The proposed algorithm is based on the cat map and the logistic map. The cat map transforms the dataset into a pseudorandom state, while the logistic map provides external key to replace the pixels value during encryption.

In [14], the authors present a new one round video encryption scheme based on 1D chaotic maps. The proposed algorithm is based on the logistic map and the tent map. The video frame is first selected based on the input FS and sent to the permutation block. In permutation block, intra-frame pixels positions are permutated according to the PO received from the key generator block. The permuted frame is then given to the diffusion block.

In [15], the authors present a video encryption scheme, the proposed encryption scheme uses 3D intertwining logistic map (ILM) with cosine transformation to generate a chaotic sequence. The video frame is first selected and sent to the permutation process. In the process of permutation, intra-frame pixels positions are permutated, and then each frame is rotated 90 degrees in the anticlockwise direction. The changes then are made in the image, in terms of rows and columns. Finally, all the encrypted frames are shuffled according to a frame selection key.

In [16], the authors present a selective encryption scheme based on singular value decomposition (SVD) and chaotic maps. The proposed scheme uses SVD to identify the most significant parts of each frame of the video feed for encryption. To encrypt this selected part of the frame, the logistic chaotic map generates 80 pseudo-random values, then these values are quantified to 0s and 1s to construct an 80-bits long key, which will represent the secret key of the present block cipher. Finally, the pixels of the frames are shuffled using the cat map.

A large number of chaos-based video cryptosystems have been proposed. However, chaos-based video encryption still has security problems of different degree from the viewpoint of modern cryptology [2]. In this paper, we propose a colour video encryption scheme based on the chaotic maps. The proposed scheme uses four chaotic maps to shuffle, permutation, confuse and diffuse of the video frames, and uses two other chaotic maps to initialize and control the above chaotic maps. The proposed video encryption scheme can provide high security with rather fast encryption speed.

The rest of the paper is organized as follows: Section 2 presents the chaotic maps employed in the proposed encryption scheme and Section 3 presents the proposed algorithm. Security analysis and discussion of various results obtained from testing the system based on the proposed algorithm, with various sizes of the video files is explained in section 4. Section 5 is the conclusion.

2 Chaotic Maps Employed in the Proposed Encryption Scheme

In this section, we will discuss the chaotic maps used in the proposed algorithm which used to produce random sequences. Then these random sequences are used to shuffle the position of frames in the video file, shuffle the position of pixels in each frame, and confuse the pixel values of the frame as well as in the process of diffusion at the frame level and diffusion at the level of the video file.

2.1 Arnold Cat Map

Arnolds Cat map was discovered by Russian mathematician Vladimir Arnold in 1960. Consider a N×N video frame and *x* and *y* be the row and column number of the pixels in the video frame. Thus *x* and *y* both ranges from 1 to N. Arnolds Cat map transformation of the video frame is obtained by implementing the below formula [17,18,5]:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \ mod \ N \qquad (1)$$

where *a* and *b* are positive integers, and determinant of matrix is 1.

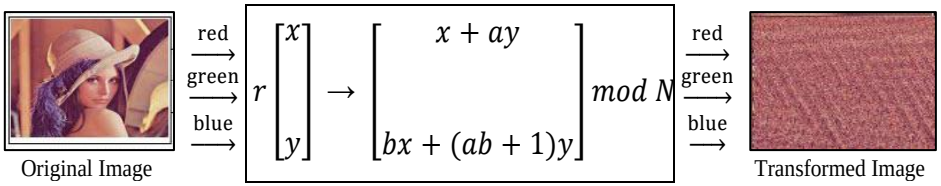


Figure 1: Arnold cat map implementation

For cat map in our proposed video encryption scheme, it takes the pixel values and shuffles them using simple matrix multiplication as shown in figure 1 where the red, green and blue pixel values are transformed individually.

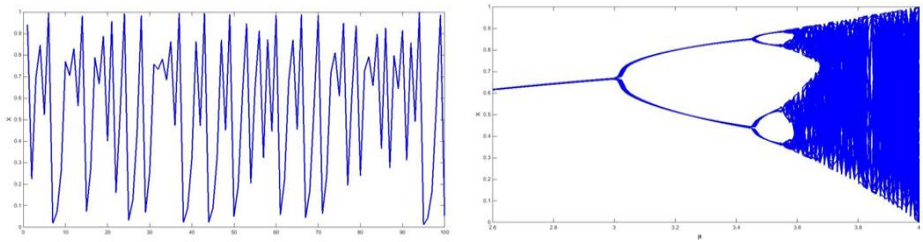
2.2 The Logistic Map

Proposed by Pierre Verhulst in 1845, the logistic map was one of the simplest and the most famous maps. Mathematically, the logistic map is written as follows [19]:

$$x_{n+1} = \mu\, x_n\, (1 - x_n\,) \qquad (2)$$

where *n* is the number of iterations and μ is the system parameter $3.57<\mu<4$ and x_0 is the initial condition of the map. Figure 2a, distinct regime for the logistic map,

is shown through 100 iterations of the logistic map (x_n, n) with μ=3.99, where n is the number of the iteration. Figure 2b shows the bifurcation diagram of logistic map.



(a) (b)

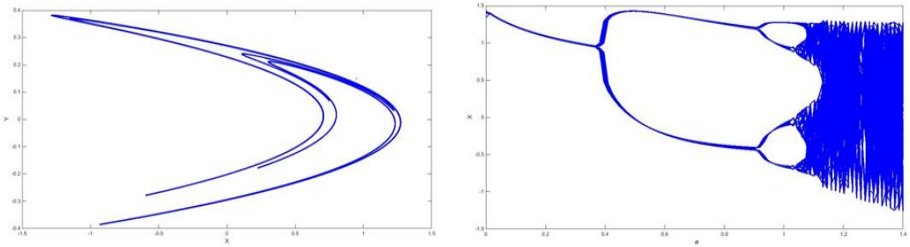
Figure 2: (a) Attractors for the Logistic Map with μ=3.99. (b) Bifurcation Diagram for the Logistic Map

2.3 Henon Map

In 1976, Henon proposed a two-dimensional non-linear discreet chaotic map. The Henon chaotic map takes two inputs (x_i, y_i) to give as an output two sets of random values. It was a simplified solution of the Poincare map for the Lorenz system [20], represented by the state equations with a chaotic attractor [21]. With keeping the same essential properties as the Lorenz system, Henon developed a system which was more accurate, faster and can be more mathematically analyzed. Mathematically, Henon chaotic map can be written as follows [20]:

$$x_{(n+1)} = 1 - \alpha x_n^2 + y_n$$
$$y_{(n+1)} = \beta x_n \quad (3)$$

where x_0 and y_0 are the initial conditions of the map and n is the number of iterations and α∈(0, 1.4] and β∈(0.2, 0.314] are the control parameters of the map. In Figure 3a, highlights the diagram of the Henon map attractor map on the x-y plane. In Figure 3b, exhibits the bifurcation behavior of the Henon map with β=0.3. The Henon chaotic map shows good chaotic behavior when α≥1.1, β=0.3 [22]. Due to this chaotic behavior, we use the Henon chaotic map in our proposed encryption scheme to produce two random sequences used in the diffusion process.



(a) (b)

Figure 3: (a) Attractors for the Henon Map with α=1.4 and β=0.3. (b) Bifurcation Diagram of the Henon Map for β=0.3

2.4 Chepyshev Chaotic Map

The Chepyshev chaotic map is a one dimensional chaotic system with one initial condition y_0 and one control parameter k. Mathematically, Chepyshev chaotic map can be written as follows [23]:

$$y_{(n+1)} = \cos(k \cos^{-1}(y_n) \qquad (4)$$

where $y_n \in [-1, 1]$ is the initial conditions of the map and n is the number of iterations and $k \in [2,\infty)$ is the control parameter of the map. The bifurcation diagram of the chaotic Chepyshev map in Figure 4 shows that all the (y_0, k) where $y_0 \in [-1,1]$ and $2 \leq k < \infty$ can be used as secret keys. In the proposed algorithm, Chebyshev chaotic map is employed for producing random sequences used to shuffle the position of the frames in the video file.

2.5 Chaotic Standard Map

The properties of chaos of the standard chaotic map were established by Boris Chirikov in 1969. The standard chaotic map has a large key space [24], mathematically, standard chaotic map can be written as follows [18]:

$$\begin{aligned} p_{(n+1)} &= p_n + c \sin(q_n) \\ q_{(n+1)} &= q_n + p_{(n+1)} \end{aligned} \qquad (5)$$

where p_0 and q_0 are the initial conditions of the map and n is the number of iterations and c is the control parameter of the map. The values of the p_n and q_n are taken modulo 2π. Figure 5, highlights the diagram of the Standard map attractor map on the p-q plane.

2.6 Lorenz Chaotic Map

Lorenz chaotic has three deferential equations known as Lorenz equations form, which is modelled by Edward Norton Lorenz [25]. Equation 6 represents Lorenz chaotic map [26]:

$$\begin{aligned} \frac{dx}{dt} &= \alpha\,(y - x) \\ \frac{dy}{dt} &= \gamma\, x - x\, z - y \\ \frac{dz}{dt} &= x\, y - \beta\, z \end{aligned} \qquad (6)$$

where α, γ, β are system parameters. The initial conditions of the Lorenz system are x_0, y_0, z_0. When α=10 and β=8/3, γ=28, the Lorenz system is in a chaotic state. In the proposed algorithm Lorenz map is used to generate the initial values of the logistic map. Figure 6 shows the numerical solution of Lorenz system with parameters values α=10 and β=8/3, γ=28 and initial conditions (x_0 =7.29, y_0 =15.16, z_0=25) gives the corresponding chaotic signals x, y, z and the different attractors of the system chaotic Lorenz.

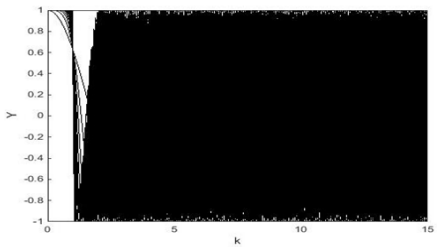


Figure 4: Bifurcation diagram of the chaotic Chepyshev map at x_0= 0.02.

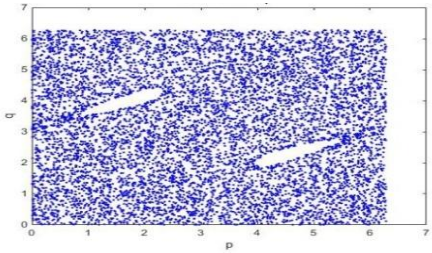


Figure 5: Attractors for the chaotic Standard map with c=4.6 and p=0.666 and q=0.282.

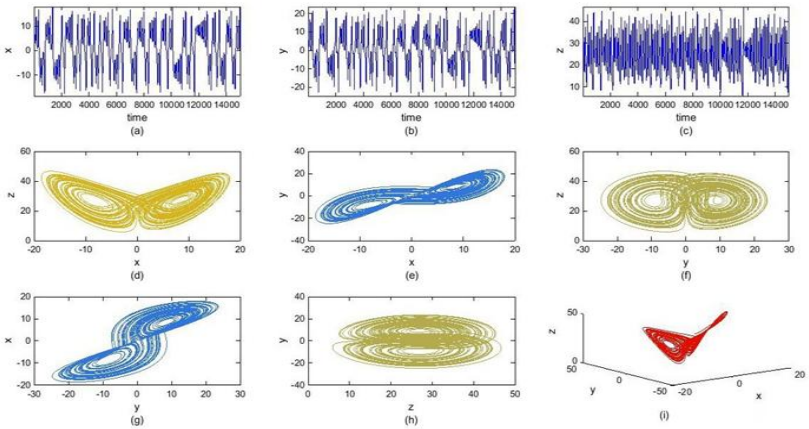


Figure 6: Simulation results of Lorenzs chaotic: (a) x chaotic signal, (b) y chaotic signal and (c) z chaotic signal, (d) (x-z) attractor, (e) (x-y) attractor, (f) (y-z) attractor, (g) (y-x) attractor, (h) (z-y) attractor, (i) (x-y-z) attractor.

3 The Proposed Scheme

In this paper, we introduce a colour video encryption scheme based on chaotic maps. Video encryption divides the file into a group of images called frames, and then the video encryption process goes through five stages: shuffling the video frames, permutation, confusion, diffusion on frame level, and diffusion on video space components level. The shuffling process shuffles video frames, and after that, these frames are handed down to the proposed video encryption scheme. Shuffling provides additional security before encrypting the video frames. In the permutation stage, intra-frame pixels positions are permutated and controlled by the keys which are generated by the cat chaotic map. The purpose of this stage is to reduce the correlation between the adjacent pixels of the encrypted video frame. In the confusion stage, the Logistic chaotic map generates three different chaotic sequences used to confuse the pixel values of the frame. In the diffusion stage, the Henon chaotic map generates two different chaotic sequences used to diffuse the frame. In the last stage of video encryption scheme, diffusion on video file level, the diffusion will be applied between the frame (i) resulting from the diffusion step and the encrypted frame (i-1) and the keys which are generated by Henon map. The overall methodology diagram is shown in Figure 7.

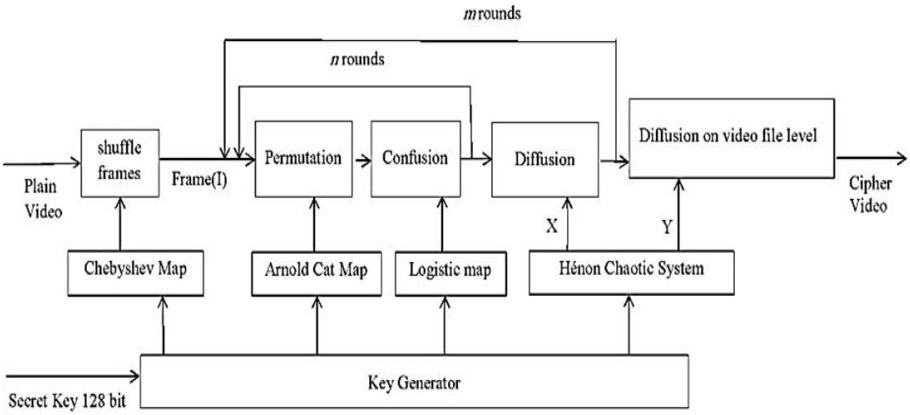


Figure 7: The architecture and the methodology of the proposed encryption

3.1 Key Generation

The proposed scheme use a proportional large size of secret key with 128 bit represented in binary number system. The binary sequence is divided into 13 segments, as shown in Table 1. The secret key consists of a set of 4 floating point numbers and 9 integers. The subkeys K_1, K_2, K_3, K_4, K_5, K_6, K_{11}, K_{12}, K_{13} that are converted from binary values to positive integers (Eq 7), where the subkeys K_7, K_8, K_9, K_{10} are converted from binary values to decimal values (Eq 8).

$$K_{int} = \sum_{i=0}^{n-1} k_n\,(i)2^i \qquad (7)$$

$$K_{doc} = \sum_{i=1}^{n} k_n\,(i)2^{i-1} \qquad (8)$$

where $K_n(i)$ is bit i in the sequence K_n.

- The subkeys K_1, K_2, K_3, K_4, K_5, K_6 are assigned to secret parameters a and b of Arnold Cat map in Eq 1, as follows:

$$a_r = K_1 \oplus K_3 \oplus K_5 \;,\; b_r = K_2 \oplus K_4 \oplus K_6$$

$$a_g = K_3 \;,\; b_g = K_4 \;,\; a_b = K_5 \text{ and } b_b = K_6 .$$

- The subkey K_7 is assigned to the initial value of the Chepyshev map in Eq 4, $x_0 = K_7$ and $k \in [2,\infty)$.
- Put $x_0 = K_8$, $y_0 = K_9$, $z_0 = K_{10}$, $\alpha = 10$, $\beta = 8/3$ and $\gamma = 28$ in Eq 6, to get the initial value of the Logistic map in Eq 2, as follows:

$$[X, Y, Z] = Lorenz\_map(x_0, y_0, z_0, NR)$$

$$XR_0 = (X \bmod 1)$$
$$XG_0 = (Y \bmod 1)$$
$$XB_0 = (Z \bmod 1) \quad (9)$$

where $XR_0 \neq 0$, $XG_0 \neq 0$ and $XB_0 \neq 0$.

- The subkey K_{11} is assigned to the round number for rotation to get the initial values of the chaotic maps, NR = K_{11}.
- Put $p_0 = (K_{12} \bmod 2\pi)$, $q_0 = (K_{13} \bmod 2\pi)$ and c= 4.6 in Eq 5 to get the initial values of the Henon map x_0, y_0 in Eq 3, as follows:

$$[x, y] = Standard\_map(p0, q0, NR)$$
$$x_0 = \frac{x}{100}$$
$$y_0 = \frac{y}{100} \quad (10)$$

Table 1: Subkeys and Parameters

Subkey	size	Range	Parameter	Chaotic Map
K_1	8bit	(0-2^8)	a_r	Arnold Cat Map Parameters
K_2	8bit	(0-2^8)	b_r	
K_3	8bit	(0-2^8)	a_g	
K_4	8bit	(0-2^8)	b_g	
K_5	8bit	(0-2^8)	a_b	
K_6	8bit	(0-2^8)	b_b	
K_7	8bit	(0-1)	X_0	Chepyshev Map
K_8	8bit	(0-1)	x_0	Lorenz Map →Logistic Map
K_9	8bit	(0-1)	y_0	
K_{10}	8bit	(0-1)	z_0	
K_{11}	16bit	(0-2^{16})	NR	Iterations number
K_{12}	16bit	(0-2π)	p_0	Standard Map → Henon Map
K_{13}	16bit	(0-2π)	q_0	

The chaotic key stream (CKS) is generated by iterating the chaotic Chepyshev map, Cat map, Henon map, and logistic map using the secret key, as follows:

- The shuffle key generates using chaotic Chepyshev map. A sequence of size numFrames generates by iterating the Chepyshev map in Eq 4 using the initial condition y_0 and constant k=2, as follows:

$$X = ChepyshevMap(y_0, numFrames)$$

where numFrames is the number of frames in video file.

$$[K_{chepy}, f_x] = sort(X) \qquad (11)$$

where sort () indicates index sequencing role, f_x index values and K_{chepy} is sequences obtained after ascending to X.

- The permutation key generates using chaotic cat map, where generate a sequence of numbers of size $h \times w$ by iterating the cat map using the initial condition a and b, as follows:

$$[X_r, Y_r] = \text{CatMap}(a_r, b_r, \text{h}, \text{w})$$
$$[X_g, Y_g] = \text{CatMap}(a_g, b_g, \text{h}, \text{w})$$
$$[X_b, Y_b] = \text{CatMap}(a_b, b_b, \text{h}, \text{w})$$

where h and w are height and width of frames in video file.

- The confusion key generates using chaotic logistic map, where keep the system parameter value of logistic map μ=3.9999, which corresponds to a highly chaotic case, the initial conditions XR_0, XG_0 and XB_0 are calculated using Lorenz chaotic map as shown in Eq 9. A sequence of size $h \times w$ generates by iterating the logistic map using the initial condition XR_0, XG_0 and XB_0 obtained in Eq 9, and then XR, XG and XB are converted from real numbers to integers, as follows:

$$\text{XR} = \text{LogisitcMap}(\text{XR}_0, \text{h}, \text{w})$$
$$\text{XG} = \text{LogisitcMap}(\text{XG}_0, \text{h}, \text{w})$$
$$\text{XB} = \text{LogisitcMap}(\text{XB}_0, \text{h}, \text{w})$$

$$\begin{aligned} \text{K}(:,:,1) &= (\lfloor \text{XR} \times 10^{14} \rfloor) \bmod 256 \\ \text{K}(:,:,2) &= (\lfloor \text{XG} \times 10^{14} \rfloor) \bmod 256 \\ \text{K}(:,:,3) &= (\lfloor \text{XB} \times 10^{14} \rfloor) \bmod 256 \end{aligned} \qquad (12)$$

Here $\lfloor \rfloor$ represents the floor function.

- The diffusion key generates using chaotic Henon map, where keep the system parameter value of Henon map α=1.4 and β=0.4, which corresponds to a highly chaotic case, the initial conditions x_0 and y_0 are calculated using Standard map as shown in Eq 10. A sequence of size $h \times w \times 3$ generates by iterating the Henon map using the initial condition x_0 and y_0 obtained in Eq

10, and then X and Y are converted from real numbers to positive integers, as follows:

$$[X, Y] = HenonMap(X_0, Y_0, h, w, t)$$
$$X_{Henon} = (\lfloor X \times 10^{14} \rfloor) \bmod 256$$
$$Y_{Henon} = (\lfloor Y \times 10^{14} \rfloor) \bmod 256 \quad (13)$$

where t the number of channels, h and w are the height and width of frames in video file.

3.2 Encryption Algorithm

The proposed encryption algorithm will generate cipher video (C) from plain video (P). The full video encryption system is realized with the Algorithm 1 and Figure 7, which can be summarized up as follows:

- **Step 1**: Load the video file and stores it as a group of images called frames.
- **Step 2**: Key generation: Select a sequence of 128 bits as the key, and split them into 13 groups, which are used to initialize the chaotic maps, as discussed in section 3.1.
- **Step 3**: Perform Shuffle the plain video frames using Chepyshev map Eq 4 and Eq 11 with key y_0. The frames positions of the video file are shifted to reduce both the correlation between adjacent frames of an encrypted video file. Swap the frame (i) and frame (x(i)), where $x=K_{chepy}$, i= 1, 2, 3, ..., numFrames.
- **Step 4**: Put the f=1.
- **Step 5**: Put the C = frame(f)
- **Step 6**: Shuffle the pixels in each frame using the 2D cat chaotic map Eq 1 with keys: a_r, b_r, a_g, b_g, a_b, b_b. Convert the frame (h,w,3) into R(h,w), G(h,w), B(h,w) channels, swap the pixels I(i,j) and I(x,y) to permutate the f-frame, as follows:

$$R(i,j) \leftrightarrow R(X_r(i,j), Y_r(i,j))$$
$$G(i,j) \leftrightarrow G(X_g(i,j), Y_g(i,j))$$
$$B(i,j) \leftrightarrow B(X_b(i,j), Y_b(i,j))$$

where i= 1, 2, 3, ..., h and j= 1, 2, 3, ..., w.

- **Step 7**: Perform confusion using 3D logistic map Eq 2 and Eq 12 with keys XR_0, XG_0, XB_0 , μ=3.99, as follows:

$$C = ((C + K) \ mod \ 256)$$

- **Step 8**: Repeat **Step 6** to **Step 7** (N round).
- **Step 9**: Diffusion on frame level: Perform diffusion on confused frame using Henon map Eq 3 and Eq 13 with keys X_0, Y_0, α=1.4 and β=0.3, perform "XOR NOT plus mod" operations on C(index), C(index-1) and chaotic key stream K= X-Henon, as the following:

$$t = (K_{chepy}(f) \ mod \ 4)$$

$$C(i) = \begin{cases} ((C(i) + K(i)) \ mod \ 256) \ \oplus \mathrm{C}(i-1) & if \ t = 0 \\ C(i) \oplus \mathrm{K(i)} \oplus \mathrm{C}(i-1) & if \ t = 1 \\ NOT(C(i) \oplus \mathrm{K}(i)) \oplus \mathrm{C}(i-1) & if \ t = 2 \\ ((C(i) + K(i)) \ mod \ 256) \oplus \mathrm{K}(i) \oplus \mathrm{C}(i-1) & if \ t = 3 \end{cases} \quad (14)$$

where index i=1,2,3,..., (h×w×3) and C(0)=K(1).

- **Step 10**: Repeat **Step 6** to **Step 9** (M round).
- **Step 11**: Put the frame (f)=C and f=f+1, if f≤NumFrames then go to **Step 5**.
- **Step 12**: Finally, diffusion on video file level. Perform diffusion between the encrypted frame (i) and the encrypted frame (i-1), as follows:

$$frame\ (i) = ((frame\ (i) \oplus frame\ (i-1)) + K) \ mod \ 256$$

where i=1, 2, ..., numFrames and K=Y-Henon Map, frame (0)=K.

3.3 Decryption Algorithm

The proposed decryption algorithm will regenerate the plain video (P) from the cipher video (C) . This algorithm consist of the following steps:

- **Step 1**: Load the cipher video file and stores it as a group of images called frames.
- **Step 2**: Key generation: Select a sequence of 128 bits as the key, and split them into 13 groups, as discussed in section 3.1.
- **Step 3**: Inverse diffusion on video file level. Perform inverse diffusion between the frame (i) and frame (i-1) and K= Y-Henon, as follows:

$$frame\ (i) = ((frame\ (i) - K) \ mod \ 256) \oplus frame\ (i-1)$$

where i= numFrames, ...,2,1 and frame (0) = K.

- **Step 4**: Put the f=1.
- **Step 5**: Put the image = frame (f)
- **Step 6**: Inverse diffusion frame-level. Perform inverse diffusion using Henon map Eq 3 and Eq 13 with keys X_0, Y_0, α =1.4 and β= 0.3, as follows:

$$C = image, t = (K_{chepy}(f) \bmod 4),\ K = X - Henon$$

$$image(i) = \begin{cases} (((C(i) \oplus C(i-1)) - K(i)) \bmod 256 & if\ t = 0 \\ C(i) \oplus \mathrm{C(i-1)} \oplus \mathrm{K}(i) & if\ t = 1 \\ NOT\big(C(i) \oplus \mathrm{C}(i-1)\big) \oplus \mathrm{K}(i) & if\ t = 2 \\ \big(\big(C(i) + C(i-1) \oplus K(i)\big) - K(i)\big) \bmod 256 & if\ t = 3 \end{cases} \quad (15)$$

Algorithm 1: Full video encryption system

input : Plain Video
output: Cipher Video

```
Begin
    [h,w]=size of (Video frames)
    Select a sequence of 128 bits as the key.
    Key generation, as discussed in section 3.1 .
    t = (K_chepy mod 4)
        /*Shuffle Frame Process*/
    for i = 1 to numFrames do
        x=K_chepy (i)
        SwapFrame(frame(i), frame(x))
    end
    for i = 1 to numFrames do
        C=frame(i)
        for m = 1 to M_rounds do
            for n = 1 to N_rounds do
                /*Permutation Process */
                C= SwapPixel(C,X_Cat,Y_Cat)
                /*Confusion Process */
                C= ((C+K_Logistic) mod 256)
            end
            /* Diffusion process on frame level (Eq 14) */
            C=Diffusion(C, X_Henon , t(i))
```

```
        end
        frame(i)=C
    end
            /*Diffusion process onvideo file level*/
    frame(1) = ((frame(1) ⊕ Y_Henon ) + Y_Henon) mod 256
    for i = 1 to numFrames-1 do
      frame(i+1) = ((frame(i+1)⊕ frame(i))+Y_Henon) mod 256
    end
end
```

- **Step 7**: Perform inverse confusion using Logistic Map Eq 2 and Eq 12, with keys XR_0, XG_0, XB_0 and μ = 3.99, as follows:

$$image = (image - K)\ mod\ 256$$

- **Step 8**: Perform inverse permutation using Arnold Cat map Eq 1 with keys a and b.
- **Step 9:** Repeat **Step 7** to **Step 8** (N round).
- Step **10**: Repeat **Step 6** to **Step 9** (M round).
- **Step 11**: Put the frame (f)=image and f=f+1, if f≤NumFrames then go to **Step 5**.
- **Step 12**: Perform inverse shuffle the video frames using Chepyshev map Eq 4 and Eq 11 with key y_0. Inverse swap the frame(i) and frame(x(i)), where x=K_{chepy} and i= 1, 2, 3, ..., numFrames.

4 The Result of the Proposed Experiment

In this section, we discuss the security analysis of the proposed encryption scheme such as statistical analysis, sensitivity analysis, key space analysis, and time analysis to prove that the proposed cryptosystem is robust against the chosen/known plaintext attacks and is suitable for real-time applications. To examine our encryption scheme, we manipulated several experiments on different video sequences, with (M-rounds=1, N-rounds=1), the employed video samples are taken from the Internet.

4.1 Histogram Analysis

The video frame histogram indicates the intensity of the pixels in the video frame that the histogram represents. In the original video frame, the pixel density

is different, and for good video frame encryption, the density distribution should be almost uniform. Figures 8 and 9 show that the encrypted video frame have fairly uniform histogram and significantly different from the histogram of original video frame. Hence, it does not provide any useful information for the attackers to perform any type of statistical attack on the encrypted video.

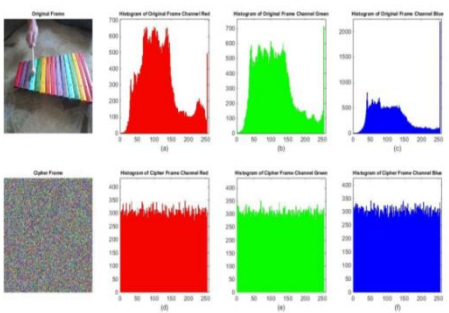


Figure 8: Histogram Xylophone Video Frame (30): (a,b,c) Plain Frame, (d,e,f) Cipher Frame

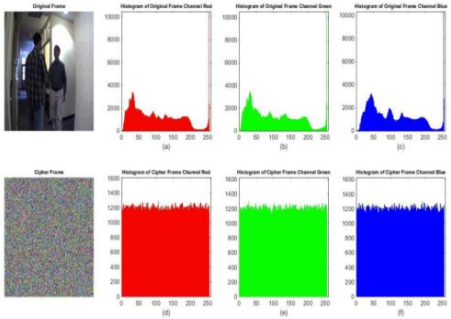


Figure 9: Histogram Handshake Video Frame (60): (a,b,c) Plain Frame, (d,e,f) Cipher Frame

4.2 Statistical Analysis

Statistical analysis is performed on the proposed video encryption scheme, demonstrating its superior confusion and diffusion properties which strongly resist statistical attacks. This is evaluated by calculating the correlation coefficient, signal noise ratio (SNR), mean square error (MSE) and peak signal noise ratio (PSNR) of ciphered video files.

4.2.1 Correlation Coefficient (CC)

The correlation coefficient between the original video frame and the encrypted video frame indicates the extent to which the original video frame is related to the encrypted video frame. It is used to find a relationship between the encrypted video frame and the original video frame and to access to the key or secret message. If CC is equal to 1, it indicates a correlation between the two values. -1 indicates that there is an inverse correlation between the two values, whereas 0 indicates no correlation [27], and computes the correlation coefficient between the pixel values [4,9]:

$$CC = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2}} \quad (16)$$

where $E(x)=\frac{1}{N}\sum_{i=1}^{N} X_i$ and x_i, y_i are the values of the original and cipher video frame pixel. Results for different video samples are shown in Table 4. The correlation coefficients are very low (C C≈0), which shows that the plain video file is nearly independent from the encrypted video file. In Table 2, the values of CC are very low, they indicate no correlation between the two adjacent pixels of encrypted video frame. In Table 3, the values of CC are very low, and it indicates no correlation between the two adjacent frames of encrypted video file.

4.2.2 Peak Signal-to-Noise Ratio (PSNR)

PSNR used to find out how distorted the original video frame when it was encrypted. For a good encryption algorithm, the PSNR values should be low. The PSNR is defined as follows [9]:

$$PSNR = 10 \times \log_{10}\left[\frac{(2^L-1)^2}{MSE}\right] \quad (17)$$

where L is the number of bits per pixel of the video frame.

4.2.3 Mean Squared Error (MSE)

MSE was used to measure the performance of the decryption procedure, the smaller MSE value, is the better video frame quality is recovered. On the contrary, the greater the MSE value, is the worse video frame quality is recovered, MSE is defined as follows [9]:

$$MSE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(C_1(i,j) - C_2(i,j))^2}{N \times M} \quad (18)$$

where M is the width and N is the height of the video frame, $C_1(i,j)$ and $C_2(i,j)$ illustrate the pixel position at *i* row and *j* column for the plain frame and cipher frame respectively.

Table 2: Correlation coefficients of two adjacent pixels in the plain and cipher video files (Vertically (V.), Horizontally (H.), Diagonal (D.))

File name	Original video			Cipher video		
	V.	H.	D.	V.	H.	D.
xylophone	0.9789	0.9612	0.9505	0.0016	0.0021	0.0034

viptraffic	0.855	0.8459	0.7348	0.0055	0.0058	0.0077
handshake-right	0.9948	0.9869	0.9823	0.0013	0.0014	0.0019
vipbarcode	0.9857	0.9231	0.8926	0.0019	0.0017	0.0023
singleball	0.9916	0.9927	0.9857	0.0024	0.0021	0.0024
vipmosaicking	0.9854	0.9947	0.9842	0.0027	0.0027	0.0038
viplanedeparture	0.9869	0.9957	0.9859	0.0028	0.0027	0.0033
viptrain	0.9653	0.9547	0.9545	0.0027	0.0028	0.0033
shuttle	0.9749	0.9505	0.9221	0.0019	0.0017	0.0021

Table 3: Correlation Between Frame (i) and Frame (i+1) in Cipher and Plain Video Files

File name	Encryption frame(i) & (i+1)	Original frame(i) & (i+1)
xylophone	0.00161	0.9661
viptraffic	0.00327	0.83251
handshake	0.00098	0.9904
vipbarcode	0.00012	0.7611
singleball	0.00107	0.9968
vipmosaicking	0.00151	0.9534
viplanedeparture	0.00158	0.9721
viptrain	0.00120	0.9848
shuttle	0.00122	0.9798

Table 4: Correlation Between Plain and Cipher Video Files

File name	Correlation Coefficients
xylophone	0.000191
viptraffic	0.000456
Handshake_right	0.000783
vipbarcode	0.001785
singleball	0.001234
vipmosaicking	0.001860
viplanedeparture	0.001622
viptrain	0.001577
shuttle	0.0013071

4.2.4 Signal to Noise Ratio (SNR)

The signal to noise ratio (SNR) measures the noise content in the encrypted data signal, the SNR values of encrypted video files are calculated based on the Eq (19) [2]:

$$SNR = 10 \times \log_{10} \frac{\sum_{i=1}^{N} |x_i^2|}{\sum_{i=1}^{N} (|x_i| - |y_i|)^2} \qquad (19)$$

where x_i and y_i denote the color intensities of pixel in the plain frame and encrypted frame respectively. In Table 5, the values of MSE are very large and the values of PSNR and SNR are very low. Thus the algorithm proposed has a high resistance against statistical attacks.

4.3 Entropy Analysis

The entropy is the measure of randomness of encryption video. Since the video frame pixel carries 8 bit information, the expected ideal entropy value should be 8. By means of the entropy of the information, it is possible to know how the video pixel values are distributed. The entropy H is calculated using the equation 20 [4, 5]:

$$H(m) = -\sum_{i=0}^{2^N-1} P(m_i) \log_2[P(m_i)] \qquad (20)$$

where $p(m_i)$ is the probability that a pixel has a specific gray value m_i and N denotes the total number of gray values. Video frame with 256 colors would achieve an entropy of 8 bits; an encrypted video frame should thus get close to that value. In the proposed video encryption scheme, the information entropy is H(m)=7.99, which is very close to the ideal value. This means a high diffusion and substitution is achieved by the proposed algorithm. So, the algorithm proposed has a high resistance against entropy attacks.

Table 5: PSNR , SNR and MSE values of encrypted video files

File name	PSNR	SNR	MSE
xylophone	8.743	2.513	8687
viptraffic	9.747	4.985	6907
handshake-right	8.102	2.123	10069
vipbarcode	9.254	4.803	7811
singleball	7.411	-2.764	11803
vipmosaicking	8.103	3.211	10068
viplanedeparture	8.980	3.716	8225
viptrain	8.063	2.105	10158
shuttle	9.026	5.544	8146

Table 6: Entropies of plain and cipher video files

File name	Original video	cipher video
xylophone	7.587	7.999
viptraffic	7.081	7.997
handshake-right	7.654	7.999
vipbarcode	7.062	7.999
singleball	6.057	7.999
vipmosaicking	7.590	7.999
viplanedeparture	6.923	7.999
viptrain	7.495	7.999
shuttle	6.687	7.999

4.4 Differential Attack Analysis

We have done many measures to check the security of the proposed video encryption scheme. These measures consist of key, plain-text sensitivity analysis and key space analysis. Each of these measures is shown in detail in the following subsections. The value of number of changing pixel rate (NPCR) between two video frames C_1 and C_2 of size N×M is calculated by [28,9,5]:

$$NPCR = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} D(i,j)}{N \times M} \times 100\% \qquad (21)$$

where $D(i,j) = \begin{cases} 0 & if \quad C_1(i,j) = C_2(i,j) \\ 1 & if \quad C_1(i,j) \neq C_2(i,j) \end{cases}$

Like NPCR, the unified averaged changed intensity (UACI) is also used to measure the difference between two video frame C_1 and C_2 of size N×M and is calculated by:

$$UACI = \frac{1}{N \times M} \times \left[\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{|C_1(i,j) - C_2(i,j)|}{2^L - 1}\right] \times 100\% \qquad (22)$$

where $C_1(i,j)$ and $C_2(i,j)$ denotes the video frame pixel at (i,j) position of the encrypted video frame and L is the number of bits per pixel of the video frame.

4.4.1 Plainvideo Sensitivity Analysis

The number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) are two most common quantities used to evaluate the strength of video encryption algorithms with respect to differential attacks [29]. Tests have been performed on the proposed scheme, about the one-pixel change influence on video file. We encrypt the original video file using the (*k*) key to get the encrypted video file C_1 and we change one bit in the original video file and encrypt it by using the same (*k*) key to get the encrypted video file C_2. In addition, we measure the similarity of both encrypted video files C_1 and C_2 using NPCR. Table 7, gives NPCR and UCAI of encrypted videos with one bit differ from the plain video. The results show that a swiftly change in the original video file will give a large change in the ciphered video file. Therefore, our proposed video encryption scheme has a very high resistance against differential attacks.

4.4.2 Key Sensitivity

The cipher video bytes produced by the cryptosystem should be sensitive to the secret key, to validate this property, we apply NPCR. We encrypt the same video under two different keys k_1, k_2 and we measure the similarity of both cipher-text using NPCR, the difference between k_1 and k_2 are one bit flip at a random position. For the proposed algorithm, the value of percentage difference is between video-1 encrypted and encrypted video-2. It is greater than 99.60% for all the test video

files with a small change in key, as shown in Table 8. Hence, slight change in the value of the key will give different output, this makes the system more secure.

4.5 Key Space Analysis

The key space is the total number of trial keys that can be used to break the cryptosystem. In the proposed video encryption scheme, a secret key of 128-bits long is used. Thus, it has 2^{128} different combinations (3.40×10^{38}) which is practically huge key space to protect the system from brute force attack.

Table 7: Sensitivity to Plainvideo frame: NPCR and UACI of encrypted video files with one bit difference from the plain video file

File name	NPCR	UACI
xylophone	99.995	33.463
viptraffic	99.979	33.299
Handshake_right	99.995	33.298
vipbarcode	99.959	33.456
singleball	99.985	33.313
vipmosaicking	99.957	33.550
viplanedeparture	99.977	33.991
viptrain	99.997	33.416
shuttle	99.955	33.510

Table 8: Comparison of pixel difference between video files encrypted by key with one-bit difference

File name	NPCR	UACI
xylophone	99.609	33.464
viptraffic	99.614	33.478
Handshake_righ	99.609	33.465
vipbarcode	99.609	33.459
singleball	99.609	33.470
vipmosaicking	99.607	33.462
viplanedeparture	99.609	33.464
viptrain	99.610	33.461
shuttle	99.592	33.463

4.6 Time Analysis

To check performance of the proposed video encryption scheme, we have analyzed the speed of the proposed video encryption technique on an Intel Core i3 CPU 2.53GHz and 1.86GB of RAM and MATLAB R2014b programming. Table 9 shows the encryption/decryption and key generation time for the proposed scheme and different video samples.

Table 9: Encryption/ Decryption and key generation time (in seconds) for the proposed scheme and different video samples

File name	Size in byte	Encrypt time	Decrypt time
xylophone	32486400	2.650	2.870
viptraffic	6912000	0.656	0.671
Handshake_right	133632000	12.907	12.453

vipbarcode	61056000	5.042	4.996
singleball	23328000	2.594	2.549
vipmosaicking	16358400	1.496	1.558
viplanedeparture	87350400	7.860	7.499
viptrain	162259200	13.695	14.257
shuttle	53526528	4.234	4.591

4.7 Comparison with Other Works

In this section, we made a comparison between our scheme and other previous works. We made a comparison between the NPCR, UACI and correlation coefficient, information entropy, PSNR of our scheme and other previous works, Table 10-11-12.

Table 10: PSNR Comparison Between Our Scheme and Other Works

Scheme	PSNR
Our Scheme	7.41 - 9.746
M.K. Ibrahem, L.A.Hamood [1]	8.184 - 11.978
R.I. Abdelfatah et al [5]	6.344 - 8.36
Li, Xiaodong et al [12]	7.66 - 27.60
M.Dua, D.Makhija et al [15]	31.14 - 36.67

Table 11: Entropy Comparison Between Our Scheme and Other Works

Scheme	Entropy
Our Scheme	7.9967 - 7.999
R.I. Abdelfatah et al [5]	7.997 - 7.999
M.M. Eid et al [4]	7.9913 - 7.9978
Li. Xiaodong et al [12]	7.9998
M.Dua, D.Makhija et al [15]	7.9972 - 7.9979
O. Benrhouma et al [16]	7.911-7.945

Table 12: Correlation Coefficient Comparison Between Our scheme and Other Works

Scheme	CC H.	CC V.	CC D.
Our Scheme	0.001 - 0.005	0.001 - 0.002	0.002 - 0.007
Roayat et al [5]	0.00057 - 0.0042		
R.R. kumar [14]	0.032 - 0.047	0.021 - 0.026	0.017 - 0.046

M.Dua et al [15]	0.001 - 0.016	0.003 - 0.026	0.0001 - 0.01
Oussama et al [16]	0.002 - 0.009	0.003 - 0.009	0.004 - 0.008

Table 13: NPCR & UACI Comparison Between Our Scheme and Other Works

Scheme	NPCR	UACI
Our Scheme	99.95 - 99.996	33.297 - 33.99
R.I. Abdelfatah et al [5]	99.59 - 99.65	29.8 - 39.8
M.M. Eid et al [4]	99.687	33.47
R.R. kumar [14]	99.51 - 99.61	33.50 - 33.62
M.Dua et al [15]	99.60 - 99.61	31.98 - 37.66
O. Benrhouma et al [16]	99.64 - 99.87	30.18 - 33.78

5 Conclusion

In this paper, we propose a secure scheme for video encryption based on chaotic maps. Experiments conducted and comparison with other video encryption schemes demonstrate the efficiency and security of the proposed video encryption scheme. The key space of our scheme is 2^{128}. It makes the brute force attack impracticable. The entropy values of the encrypted video frames are greater than 7.99. The correlation coefficient values of the encrypted video frames using the proposed scheme are close to zero and also encryption video frames pixel values have random uniform distribution. The peak signal to noise ratio of encrypted video frames reached 7db. In addition, the proposed algorithm is sensitive to plaintext and secret key. NPCR values are greater than 99.90% and UACI values are 33.46%. Hence, these results show the efficiency and security of our scheme. The detailed analysis of our scheme can implemented by simple hardware and software.

REFRENCES

[1] Mahmood K.Ibrahem, Laith Abdulhussein Hamood, “Video encryption based on chaotic system and stream cipher”, Journal of Information and Communications Technology, vol. 1, no. 2, 2018.

[2] Shujun Li, Xuan Zheng, Xuanqin Mou, Yuanlong Cai, “Chaotic encryption scheme for real-time digital video”, Proc. SPIE 4666, Real-Time Imaging VI, pp. 149–160, 2002.

[3] Hephzibah Kezia, Gnanou Florence Sudha, "Encryption of digital video based on lorenz chaotic system", In: Proceedings of the 16th International Conference on Advanced Computing and Communications, IEEE Computer Society Press, pp. 40–45, 2008.

[4] Marwa M.Eid, El-Sayed M. El kenawy, Abdelhameed Ibrahi, "A new hybrid video encryption technique based on chaos cryptography", Journal of Computer Science and Information Systems, vol. 2, 2021.

[5] Roayat Ismail Abdelfatah, Mohamed E. Nasr, Mohammed A. Alsharqawy, "Encryption for multimedia based on chaotic map: Several scenarios", Multimedia Tools and Applications, vol. 79, pp. 19717–19738, 2020.

[6] Ibrahim Yasser, Mohamed A.Mohamed, AhmedS. Samra and Fahmi Khalifa, "A chaotic-based encryption/decryption framework for secure multimedia communications", Entropy, vol. 22, no. 1253, 2020.

[7] Yussra Majid Hameed, Nada Hussien M.Ali, "An efficient audio encryption based on chaotic logistic map with 3D matrix", Journal of Theoretical and Applied Information Technology, vol. 96, no. 16, pp. 5142–5152, 2018.

[8] Osama S.Faragallah and Hala S.EL−sayed, "Secure opto−audio cryptosystem using XORing mask and hartley transform", IEEE Access, vol. 9, 2021.

[9] M.Y. Mohamed Parvees, "Audio encryption - a chaos-based data byte scrambling technique", Applied Systemic Studies, vol. 8, no. 1, 2018.

[10] Chunhu Li, Guangchun Luo, Ke Qin and Chunbao Li, "Chaotic Image Encryption Schemes: A Review", Advances in Engineering Research, vol. 86, 2017.

[11] Fang Shang, Kehui Sun, Yongqi Cai, "An efficient MPEG video encryption scheme based on chaotic cipher", IEEE Congress on Image and Signal Processing, pp. 12–16, 2008.

[12] Li.Xiaodong, Yu.Haoyang, Zhang Hongyu, Jin.Xin, Sun.Hongbo, Liu.Jing, "Video encryption based on hyperchaotic system", Multimedia Tools and Applications, 2020.

[13] T.Fang X.Huang, D.Arnold and J.Saniie, "A chaotic-based encryption /decryption system for secure video transmis- sion", IEEE International Conference on Electro Information Technology (EIT), pp. 369–373, 2021.

[14] R. Ranjith kumar, D.Ganeshkumar , A.Suresh, "A New One Round Video Encryption Scheme Based on 1D Chaotic Maps", International Conference

on Advanced Computing & Communication Systems (ICACCS), pp. 369–373, 2019.

[15] Drishti Makhija, Pilla Yamini Lakshmi Manasa Mohit Dua and Prashant Mishra, "3D chaotic map-cosine transformation based approach to video encryption and decryption", Open Computer Science, 2022.

[16] Oussama Benrhouma, Ahmad B.Alkhodre, Ali AlZahrani, Abdallah Namoun and Wasim A.Bhat, "Using singular value decomposition and chaotic maps for selective encryption of video feeds in smart traffic management", Appl. Sci, 2022.

[17] Apeksha Waghmare, Abhishek Bhagat, Abhishek Surve, Sanuj Kalgutkar, "Chaos based image encryption and decryption", International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 4, April 2016.

[18] Jun-xin Chen, Zhi-liang Zhu, Li-bo Zhang, Chong Fu, and Hai Yu, "An efficient diffusion scheme for chaos-based digital image encryption", Mathematical Problems in Engineering, 2014.

[19] Alligood KT, Sauer TD, Yorke JA, "Chaos an introduction to dynamical systems", firsted. New York: Springer-Verlag, 1996.

[20] Jansher Khan, Jawad Ahmad, Seong Oun Hwang, "An efficient image encryption scheme based on: Henon map, skew tent map and S-Box", International conference on modeling, simulation, and applied optimization (ICMSAO), pp.1-6, 2015.

[21] M. Henon, "A two-dimensional mapping with a strange attractor", Communications in Mathematical Physics, vol. 50, pp. 69–77, 1976.

[22] Abdullah Qayyum, "Chaos-based confusion and diffusion of image pixels using dynamic substitution", IEEE Access, vol. 8, 2020.

[23] T. Geisel, V. Fairen, "Statistical properties of chaos in chebyshev maps", Phys. Lett.A, vol. 105, no. 6, pp. 263–266, 1984.

[24] Lian S, Sun J, Wang Z, "A block cipher based on a suitable use of chaotic standard map", Chaos Solitons Fract, pp. 17–29, 2005.

[25] Edward Norton Lorenz, "Deterministic non-periodic flow", journal of Atmospheric Sciences, vol. 20, no. 2, pp. 130–141, 1963.

[26] Mahmood Z. Abdullah, Zinah J. Khaleefah, "Design a hybrid cryptosystem based chaos and sharing for digital audio", Iraqi Journal of Computers,

Communication and Control & System Engineering (IJCCCE), vol. 17, pp. 59–70, 2017.

[27] Noha Ramadan, Hossam Eldin H.Ahmed, Said E.Elkhamy, Fathi E.Abd El-Samie, "Chaos-based image encryption using an improved quadratic chaotic map", American Journal of Signal Processing, vol. 6, no. 1, pp. 1–13, 2016.

[28] Song, Xian-Hua, Hui-Qiang Wang, Salvador E. Venegas-Andraca, and Ahmed A.Abd El-Latif, "Quantum video encryption based on qubit-planes controlled-xor operations and improved logistic map", Physica A: Statistical Mechanics and its Applications, vol. 573, no. 122660, 2020.

[29] Yue Wu, Joseph P. Noonan, and Sos Agaian, "NPCR and UACI randomness tests for image encryption", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), vol. 1, no. 2, pp.31–38, 2011.

[30] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps", International Journal of Bifurcation and Chaos in Applied Sciences and Engineering, vol. 8, no. 6, pp. 1259–1284, 1998.

[31] Guanrong Chen, Yaobin Mao, Charles K.Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps", Chaos Solitons and Fractals, vol. 21, no. 3, pp. 749–761, 2004.

[32] Vinod Patidar, N.K. Pareek and K.K.Sud, "A new substitution- diffusion based image cipher using chaotic standard and logistic maps", Communications in Nonlinear Science and Numerical Simulation, vol. 14, no. 7, pp. 3056–3075, 2009.

[33] Bhaskar Mondal, Shrey Singh, Prabhakar Kumar, "A chaotic permutation and diffusion based image encryption algorithm for secure communications", Springer Science+Business Media, LLC, part of Springer Nature, 2018.

[34] Abdullah Qayyum, Jawad Ahmad, Wadii Boulila, Saeed Rubaiee, "Chaos-based confusion and diffusion of image pixels using dynamic substitution", IEEE Access, vol. 8, pp. 140876–140895, 2020.

[35] W.Diffie and M.E.Hellman, "New directions in cryptography", IEEE Trans. Inf. Theory, vol. 22, pp. 644–654, 1976.

[36] X.Y.Yu, J.Zhang, H.E. Ren, G.S.Xu1 and X.Y. Luo, "Chaotic Image Scrambling Algorithm Based on S-DES", Journal of Physics: Conference Series, vol. 48, pp. 349–353, 2006.

[37] N.K. Pareek, Vinod Patidar, K.K.Sud, N.K.Patidar, "Image encryption using chaotic logistic map", Image and Vision Computing, vol. 24, no. 9, pp. 926–934, 2006.

[38] A. Akhshani, A. Akhavan, S.C. Lim, Z. Hassan, "An image encryption scheme based on quantum logistic map", Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 12, pp. 4653–4661, 2012.

[39] Pan Tian gong, Li Da-yong, "A Novel Image Encryption Using Arnold Cat", International Journal of Security and Its Applications, vol. 7, no. 5, 2013.

[40] Parker TS, Chua L, "Chaos : A tutorial for engineers", Proceedings of the IEEE, vol. 75, no. 8, pp. 982–1008, 1995.

[41] Dachselt F, Schwarz W, "Chaos and cryptography", IEEE Trans Circuits Syst, vol. 48, pp. 498–509, 2001.

[42] Wu CW, Rulkov NF, "Studying chaos via 1-D maps-a tutorial", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, vol.40, no.10, pp.07–721, 1993.