An Encryption Scheme using Poly Alphabetic Substitution, Permutation and Diffusion along with 10 Dimensional Quantum Logistic Map

Dikshant Sharma[1], Vyoma Vaish[2], Garv Singhal[3], Hemang Mehra[4], Naveen Tayal[5], Krish Tanwar[6], Himanshi[7], Drishti Singla[8], Dimple[9]
Dikshant112@gmail.com, vyoma873@gmail.com, garvsinghal39@gmail.com, mehra.hemang2023@gmail.com, Naveen.personal.001@gmail.com, krishtanwar9506@gmail.com, himanshig134@gmail.com , drishtisingla83@gmail.com , dimplemuskan123@gmail.com
J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India[1,2,5,6,8]
Indian Institute of Technology, Gandhinagar, Gujarat, India[3]
San Jose State University, San Jose, California, United States of America[4]
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India[7]
Shri Vishkarma University, Gurugram Haryana, India[9]

Abstract
In the modern era, the increasing exchange of multimedia content over public networks heightens the risk of unauthorized access, making it essential to implement encryption mechanisms to prevent piracy, copyright violations, and unauthorized use. Numerous researchers have proposed various encryption mechanisms to address these concerns. An effective encryption scheme must ensure a large key space and demonstrate strong performance in quantitative security metrics (such as PSNR, correlation, SSIM, Entropy, correlation coefficient) Additionally, it must be robust against a various attacks (including noise and geometric attacks), while successfully passing differential attacks. In this work, an encryption scheme that is based on 10-dimensional logistics map with intra polyalphabetic substitution, permutation and diffusion is proposed ensuring the aforementioned security features. The results show that the proposed approach have a large key space, qualitative security analysis, robustness against differential attacks, and resilience to other common attack vectors.

Keywords: Security, Encryption, 10D Quantum Logistic Map, Differential Attacks, Noise and Geometric Attacks, Key Space, Quantitative Security Analysis, Polyalphabetic Substitution, Permutation and Diffusion.

1. Introduction

The need for multimedia content security has grown significantly over the past few decades due to increased connectivity and the rapid expansion of digital media which increased the associated rise in cyber threats. Graph represented in figure-1 has been prepared after combing the reports of security breaches provided by Verizon's Data Breach Investigations Report (DBIR) which is an annual report providing statistics on data breaches as well as on cyber attacks happening globally in a year [1] and Cisco Annual Cybersecurity Report provided by Cisco consisting of analysis of cyber threats, including trends in multimedia attacks [2].The usage of multimedia content in sectors like healthcare, finance, and communication has made it essential to ensure the security of this content. Unauthorized access, tampering, and data breaches can lead to serious consequences, such as loss of privacy, intellectual property theft, and regulatory violations [3].

Over the past few decades these data threats and data breaches have eventually increased, thus creating an insecure environment for the internet users to transfer the information. Hence a requirement for a highly encrypted mechanism that can provide an assurance of high

security during the exchange of information is needed. So various researchers have proposed encryption techniques based on traditional techniques use algorithms that are based on mathematical formulas like RSA and AEF to convert plaintext into cipher text, assuring confidentiality of using either asymmetric or symmetric key-based cryptography. (Vignere cipher, AES, DES, Triple DES, Blowfish, RC4, RC5 and RC6 are some of the techniques that are included in traditional techniques) [4-12], chaotic map based techniques [13-17] and logistic maps [18-21] based techniques Leverages randomness as well as sensitivity of chaotic systems, like logistic maps, in order to generate complex encryption schemes that enhance security against cryptographic attacks. Qubit [22,26] as well as quantum based techniques [27,30] consists of principles of quantum mechanics, using qubits for quantum key distribution and quantum-safe algorithms, to achieve theoretically unbreakable encryption through quantum superposition and entanglement.

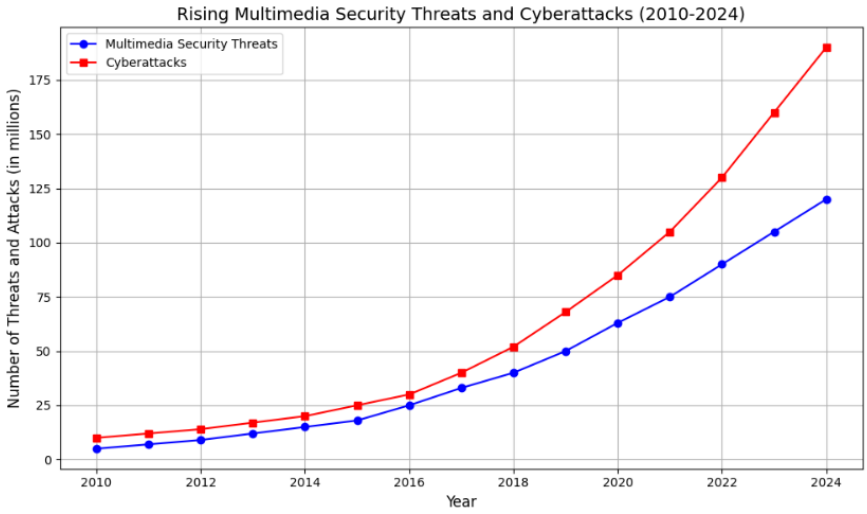


Figure-1: Graphical Representation of increasing security threats in multimedia over the time

Figure-1 shows the rising multimedia security threats over time and the number of cyber attacks which have been significantly increased, due to the growth of social media, smartphone usage, healthcare sector and later, AI-based threats like Deep fakes.

To reduce these security threats, one possible solution is encryption. Numerous researchers have proposed various algorithms starting from classical algorithm (AES, DES etc.) to chaotic based cryptography solutions moving forward towards qubit and quantum mechanics-based cryptography. A good and effective encryption scheme must ensure the following performance parameters:

- **Quantitative Security Analysis:** By conducting this analysis one can measure and compare the different security level of video encryption techniques, which includes evaluating the key space size to ensure it has a large value in order to resist against brute-force attacks, resistance to cryptanalysis etc. [31-35]. For ensuring good quantitative security analysis, the following parameters are used in this category:

  - ❖ **Correlation coefficient Analysis:** This analysis checks the relationship between original and encrypted videos. The ideal value of correlation is 0. In a secure encryption system, the value should be zero or near-zero (approaching towards zero)

correlation among the corresponding pixels in original and encrypted videos, which is considered to be good [31, 33, 34].

- ❖ **Peak to Signal Noise Ratio (PSNR) Analysis:** It is a quantitative measure which is used in assessing the quality of video or an image after compression, encryption, or other types of processing. PSNR value for video should be less than 8. PSNR is used to quantify the quality of a processed image or video compared to its original form. PSNR values indicate better quality [31-35].

- ❖ **Structural Similarity Index** (**SSIM):** it's a widely used metric for evaluating structural similarity between 2 videos or images which includes three components i.e. structure, luminance as well as contrast. SSIM ranges from -1 and 1, where 0 or approaching towards 0 is the perfect for this parameter [32-34].

- ❖ **Entropy (H):** it quantifies the unpredictability or uncertainty in a random variable or set of outcomes. In essence, entropy measures the amount of "information" or "surprise" contained in data, making it crucial for fields like data compression, communication systems, cryptography, and machine learning. Its value should be equal or near to 8 (approximately 7.99) [31-35].

- ❖ **Mean Square Error (MSE):** metric used in machine learning, statistical analysis as well as signal processing for assessing the quality of a predictor, model, or estimator. It measures average of squares of differences between predicted values (or estimated values) and actual values (or observed data points). MSE is important in applications where the accuracy of predictions or estimations plays a critical role, such as in control systems, image processing, and regression analysis [32, 34, 35].

- ❖ **Bit Error Rate (BER):** is a critical performance metric in digital communication systems, representing the ratio of the number of bit errors to the total number of bits transmitted during a given time interval. It provides a quantitative measure of the reliability and accuracy of data transmission over a communication channel. BER is crucial in the design and evaluation of systems such as wireless communications, optical fiber networks, satellite communications, and any digital data transmission scheme [34, 35].

- **Key Space:** Key space refers to the total number of possible keys that can be used by an encryption algorithm. For medical image encryption, a large key space is essential to resist brute-force attacks while protecting sensitive medical images [36-39].
- **Histogram Analysis:** A powerful technique for analysing the distribution of data by visually representing the frequency of data points across different intervals or bins. It is commonly used in digital signal processing (DSP), image processing, machine learning, and various other fields. A histogram is particularly useful in understanding the underlying characteristics of a dataset, such as its distribution, central tendency, spread, and presence of outliers. For encrypted image histogram should be flattened and should be scattered [36, 38-40].

- **Key Sensitivity:** Key sensitivity refers to how sensitive an encryption algorithm is to small changes in the key. A small alteration in the encryption key (e.g., changing just one bit), result in a completely different encrypted output, thus ensuring that similar keys produce drastically different encrypted images. High key sensitivity in medical image encryption protects against attacks that try to make minor changes in keys, thus making the encrypted data useless to intruders [36, 37, 40].
- **Differential Attack Analysis:** It is a cryptanalysis technique which is used to evaluate the security of an encryption algorithm against attacks which can manipulate slight differences between two closely related inputs, such as frames in a video. This includes the following parameters [36-40]:
  - **Number of Pixels Change Rate (NPCR):** It is metric commonly used to evaluate the strength of image and video encryption algorithms, particularly in terms of their resistance to differential attacks. For a strong encryption algorithm, the NPCR value should be near to 99.60% or above, indicating that nearly all pixels change in response to a small change, ensures high security against differential attacks [36-40].

  - **Unified Average Changing Intensity (UACI):** It is metric commonly used to evaluate the average intensity of the differences between two encrypted images (or video frames) when there is a small change in the input. For a strong encryption algorithm, the UACI value should be approximately 33.33%, which are the indicative of secure encryption scheme against differential attacks [36-40].

- **Imperceptibility:** Imperceptibility is the crucial parameter which ensures secure hidden of sensitive information or encrypted data within the image without affecting the visual appearance or quality of the image. To ensure the confidentiality of crucial content, some of the important techniques have been used to ensure the imperceptibility in encrypted images. This includes Spatial Domain Technique, Frequency Domain Technique, Chaos based encryption etc. These techniques not only ensure imperceptibility but also ensure image retaining and data security [37, 39, 40].

- **Robustness**: The method should introduce significant randomness into the encrypted video, ensuring that it can withstand various types of attacks and noises, compression and transmission errors. This robustness ensures the video can be accurately reconstructed even under adverse conditions [41-45]. This includes:

  - **Noise Attack:** This attack aims to degrade the quality of the video or obscure important details by intentionally introducing noise to encrypted video data to disrupt the original content. And thus, making it difficult to reconstruct the original content. Various types of noise include Salt-and-Pepper Noise, Gaussian Noise and Poisson Noise [41-45].

  - **Geometric Attack:** This attack aims to manipulate the geometric properties of an image or video, such as rotation, searing, translation or scaling, to distort the original content. The goal to disrupt the visual coherence of the video, pose the significance challenge to interpret the original content after encryption [41-45].

- **Encryption Time:** Different encryption algorithms have different levels of complexity. The mechanism should take minimal time to encrypt the video, ensuring quick processing. [41-45]]

In this paper 10D quantum logistic map is proposed that inculcates the following parameters mentioned above for maintaining security. A 10D logistic map is used that ensures high key sensitivity and high key space. An adaptive diffusion process is used to ensure resistance to differential attacks. An intra-polyalphabetic substitution is used to ensure good randomness and imperceptibility. An adaptive diffusion process is used to ensure resistance towards differential attacks. It helps in achieving better values of key sensitivity.

The rest of the paper is organized as follows: Section 2 presents the literature survey. Section 3 provides a detailed explanation of the proposed scheme (encryption as well as decryption mechanism). The simulation parameters are outlined in Section 4, followed by the results in Section 5. Section 6 consists of a comparative study with other state of arts techniques and Section 7 concludes the findings of this research.

2. Literature Survey

The evolution of multimedia security has led to diverse encryption techniques, including traditional algorithms (AES, RSA), chaotic systems, and quantum cryptography, each with unique strengths and limitations. Chaotic methods like logistic maps offer robustness against various attacks, while quantum cryptography promises unconditionally secure communication but faces practical challenges. Recent studies employ metrics such as PSNR, Entropy, MSE, Correlation Coefficient, histogram analysis, UACI, SSIM and NPCR to assess encryption performance, balancing security, efficiency, and imperceptibility. Below mentioned Table -1 consists of the values of the range in which the parameter qualifies a particular test and Table-2 consists of the abbreviations used throughout in this paper.

Table-1: Value or range to qualify the test

S.No.	Parameter	Value or Range to Qualify test/ parameter
1	Visual Analysis	Encrypted Image should be distorted and non identical to the original image
2	Correlation Coefficient (CC)	Nearly equal to 0 either in positive side or negative side
3	Histogram	Graph with flattened edges in entire range (in this proposed work 256x256x3) is taken
4	Peak Signal to Noise Ratio (PSNR)	in between 7 and 8
5	Entropy (H)	Nearly equal to 8 (>7.99)
6	Structural Similarity Index (SSIM)	Nearly equal to 1 (between original and decrypted images) Nearly equal to 0 (between original and encrypted images)
7	Number of Pixel Change Rate (NPCR)	Above or equal to 99.60%
8	Unified Average Change Intensity (UACI)	Above or equal to 33.33%

Table-2: Abbreviation Table

S.No.	Long Form	Abbreviated Form	S.No.	Long Form	Abbreviated Form
1	Visual Analysis	VA	7	Entropy	H
2	Histogram Graphical Analysis	HGA	8	Number of Pixel change rate	NPCR
3	Correlation Coefficient	r	9	Unified Average Change Intensity	UACI
4	Mean Squared Error	MSE	10	Noise Attack Analysis	NAA
5	Peak Signal to Noise Ratio	PSNR	11	Geometric Attack Analysis	GAA
6	Structural Similarity Index	SSIM	12	Anti -occlusion attack	AOAA

Table-3: Literature Survey Table

S.No.	Reference	Human Eye Analysis		Quantitative Security Analysis					Differential Attack Analysis		Robustness/ Other Attack Analysis		
		VA	HGA	r	MSE	PSNR	SSIM	H	UACI	NPCR	NAA	GAA	AOAA
1	Khalid M Honsy 2022 [46]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	--	✓
2	Heping Wen 2023 [47]	✓	✓	✓	✓	✓	✓	✓	✓	✓	--	--	--
3	Deepti Dhingra 2023 [48]	✓	✓	✓	✓	✓	--	✓	✓	✓	--	--	--
4	Yang Yang 2023 [49]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	--	--
5	Bharti Ahuja 2023 [50]	✓	✓	✓	--	✓	✓	✓	✓	✓	✓	--	✓
6	Mohamad Gabr 2023 [51]	✓	✓	✓	✓	✓	--	✓	✓	✓	✓	--	✓
7	Wang Jin 2023 [52]	✓	✓	--	✓	✓	✓	--	--	--	--	--	--
8	Doang Jiang 2024 [53]	✓	✓	✓	--	--	--	✓	✓	✓	✓	--	✓
9	Suo Gao 2024 [54]	✓	✓	✓	--	--	--	--	✓	✓	--	--	--
10	Monu Singh 2024 [55]	✓	✓	✓	--	✓	✓	✓	✓	✓	--	--	--
11	Tanveer Qayyum 2024 [56]	✓	✓	✓	--	✓	--	✓	✓	✓	--	--	✓
12	Himanshu Kumar Singh 2024 [57]	✓	--	✓	--	✓	✓	--	✓	✓	--	✓	--
13	Sweta Kumari 2024 [58]	✓	✓	✓	--	✓	--	✓	✓	✓	--	--	--
14	Parul Saini 2024 [59]	✓	✓	✓	--	✓	✓	✓	✓	✓	--	--	--
15	Deepti Dhingra 2024 [60]	✓	✓	✓	--	✓	--	✓	✓	✓	--	--	--
16	N.Ramesh Babu 2024 [61]	✓	✓	✓	--	✓	--	✓	--	--	--	--	--
17	R. Roselinkiruba 2023 [62]	✓	✓	--	--	✓	✓	✓	--	--	✓	--	--
18	Deyang Wu 2015 [63]	✓	--	✓	--	✓	--	--	--	--	✓	--	--

Table-3 above ✓ means that the implemented methodology passes the mentioned parameter and is within the limit such that it qualifies the test of that particular parameter, -- means that the value of parameter doesn't qualify the test. The qualifying range is mentioned in Table-1. Accordingly the failing and passing of a parameter is decided in Table-3.

3. Proposed Scheme (Encryption as well as Decryption)

In the proposed work the encryption and decryption is performed in 2 steps. Step 1 consists of the confusion process that is performed as a combination of permutation followed by substitution. Both permutation and substitution are performed at inter frame level as well as at the intra frame level (within blocks of 8x8). In Step 2 diffusion is performed in which XORing of bits is done of the permutated and substituted image with the keys generated from Quantum Logistic Map (QLM).

3.1. Key Generation Process:

In order to generate the keys using the quantum logistic maps [64,65] are used as its 10 dimensional quantum logistic maps that means that 12 parameters are involved and utilized for the generation of the keys that are used to perform the encryption process, the QLM equations used are:

$$x_{(n+1)} = (u \,.64\,. y_n \,.(1 - \, x_n) \, + \, z_n \,)\, mod\, 1 \qquad \text{eqn (1)}$$

$$y_{(n+1)} = (u \,.64\,. y_n \, + \, z_n \,.(1 - \, x_{n+1}^2 \,)\,)\, mod\, 1 \qquad \text{eqn (2)}$$

$$z_{(n+1)} = (u \,.(\, y_{n+1} \, + \, (\, x_{n+1}) \, + \, 64\,)\,.(z_n - \tfrac{1}{6}\, z_n^3 \,))\, mod\, 1 \qquad \text{eqn (3)}$$

Where:

x_n, y_n, z_n are state variables representing the system's quantum states at iteration n.

x_{n+1}, y_{n+1}, z_{n+1} are the next state variables representing the system's quantum states at $(n+1)^{th}$ iteration.

u is the control parameter influencing the system's evolution.

mod operation ensures that the state variables remain within the unit interval [0, 1], maintaining bounded and normalized state space. Initial values: x_n = 0.19274124785, y_n = 0.71213452355, z_n = 0.54345235464 and u = 2.1234432441.

$$p_{new} = \sqrt{(\sin(\pi r b(1-b))) + (\cos(\pi r b(1-b)))} \qquad \text{eqn (4)}$$

Where, r=2.1234432441, b=0.23456876543 are initial values.

These Equations 1-4 of quantum mechanics are used to generate the randomized keys that are used to carry out the encryption process.

Algorithm - 1: Random number generation using Quantum Logistics Map

```
INPUT : Initial_State_Variables = x, y, z, u, r, b; Generated Random_Numbers;

OUTPUT: Generated Random Numbers: 3D NumPy Array, SHAPE (HEIGHT, WIDTH, 3)

STEP 1: INITIALIZE AN EMPTY LIST 'random_numbers'[ ] TO STORE THE GENERATED
        RANDOM NUMBERS
STEP 2: INITIALIZE 'n' as 'n = HEIGHT * WIDTH * 3'
STEP 3: FOR EACH 'I' IN RANGE '180000 // 3':
            (a)   GENERATE RANDOM NUMBERS x1,y1 and z1 between 0 to 255.
```

```
        (b)   x1 = (u * 64 * y * (1 - x) + z) % 1
              y1 = (u * 64 * y + z * (1-x1)^2) % 1
              z1 = (u * (y1 + x1 + 64) * (z - (1/16) * z^3) % 1
              pnew   =   (math.sin(r*x*(1-x)*math.pi)+   math.cos(r*x*(1-
              x)*math.pi))% math.sqrt(2)
        (c)   key_list.extend ([abs (Num1). abs (Num2). abs (Num3)])
STEP 4: CONVERT 'key_list' TO A 3D Numpy Array 'Updated_Key' WITH SHAPE
'(Height, Width, 3)'
```

3.2. Confusion Process: In this first of all the extracted frames from the video are taken and then permutation process is applied followed by the substitution process on the images extracted from the video. Complete process of confusion is shown in Figure-2.

Permutation Process: In this first of all a Permutation box or P-Box of size (256x256x3) is generated using the Quantum Logistic Map which is further used to perform inter frame permutation in which the bits are shuffled randomly in the entire image block of each video fram. Once inter frame permutation is completed intra frame permutation is performed within the block of (8x8) In each frame of the video. On completing both the steps we get a permutated image that is further forwarded for carrying out the substitution process. In order to carry out the permutation at both inter as well as intra bit levels the keys of size 256x256x3 is used that is generated using the quantum logistic maps.

Algorithm - 2: Permutation Process

```
INPUT: EXTRACTED FRAMES OF IMAGE FROM VIDEO, Random_Numbers, KEY;

OUTPUT: PERMUTATED IMAGE of SHAPE (HEIGHT, WIDTH, 3)
STEP 1: INPUT taken as EXTRACTED IMAGES FROM INPUT VIDEO, RANDOM NUMBERS
        taken as KEY
STEP 2: FLATTEN IMAGE AS FLAT_IMAGE = IMAGE.FLATTEN()
STEP 3: FOR k IN RANGE (256):
STEP 4: PT1[K] = PT [PBOX [K]]
STEP 5: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 6: RETURN PERMUTATED IMAGE (PT1). RESHAPE (8,8)
STEP 7: FLATTEN IMAGE AS FLAT_IMAGEPT1 = IMAGE.FLATTEN()
STEP 8: FOR k IN RANGE (32):
STEP 9: PT2[K] = PT1 [PBOX [K]]
STEP 10: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 11: RETURN PERMUTATED IMAGE (PT2)
```

Algorithm - 3: Reverse Permutation Process

```
INPUT: REVERSE_DIFFUSED FRAMES, Random_Numbers, KEY;

OUTPUT: DIFFUSED IMAGE of SHAPE (HEIGHT, WIDTH, 3)
STEP 1: INPUT taken as DIFFUSED IMAGES FROM INPUT VIDEO, RANDOM NUMBERS
        taken as KEY
STEP 2: FLATTEN IMAGE AS FLAT_IMAGE = IMAGE.FLATTEN()
STEP 3: FOR k IN RANGE (256):
STEP 4: PT1[K] = PT [PBOX [K]]
STEP 5: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 6: RETURN PERMUTATED IMAGE (PT1)
STEP 7: FLATTEN IMAGE AS FLAT_IMAGEPT1 = IMAGE.FLATTEN). RESHAPE (8,8)
STEP 8: FOR k IN RANGE (32):
STEP 9: PT2[K] = PT1 [PBOX [K]]
STEP 10: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 11: RETURN PERMUTATED IMAGE (PT2)
```

Substitution Process: In this process we work on the permutated image that we have received from the previous step, Substitution box or S-Box is used here to carry out the substitution process in each frame of the video. First of all inter frame substitution is done followed by the intra frame substitution. In order to carry out the substitution at both inter as well as intra bit levels the keys of size 256x256x3 is used that is generated using the quantum logistic maps. Now this permutated and substituted image which can be termed as a confused image is sent further for diffusion process.

Algorithm - 4: Substitution Process

```
INPUT : Random_Numbers, KEY, PERMUTATED FRAMES;
OUTPUT: SUBSTITUTED ENCRYPTED IMAGE of SHAPE (HEIGHT, WIDTH, 3)

STEP 1: INPUT taken as PERMUTATED IMAGE, RANDOM NUMBERS taken as KEY
STEP 2: FLATTEN the IMAGE into a 1D ARRAY for SUBSTITUTION
STEP 3: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
        QUANTUM LOGISTIC MAPS
STEP 4: FOR k IN RANGE (256):
STEP 5: SUBSTITUTED IMAGE = SBOX [PT[k]]
STEP 6: SUBSTITUTED_ARRAY = NP.ARRAY(PT1).RESHAPE(8,8)
STEP 7: RETURN SUBSTITUTED IMAGE1
STEP 8: FLATTEN the SUBSTITUTED IMAGE1 into a 1D ARRAY for SUBSTITUTION
STEP 9: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
        QUANTUM LOGISTIC MAPS
STEP 10: FOR k IN RANGE (32):
STEP 11: SUBSTITUTED IMAGE1 = SBOX1 [PT1[k]]
STEP 12: SUBSTITUTED_ARRAY1 = NP.ARRAY(PT2)
STEP 13: RETURN SUBSTITUTED IMAGE2
```

Algorithm - 5: Reverse Substitution Process

```
INPUT : Random_Numbers, KEY, PERMUTATED FRAMES;
OUTPUT: REVERSE_PERMUTATED_IMAGE of SHAPE (HEIGHT, WIDTH, 3)

STEP 1: INPUT taken as REVERSE_PERMUTATED_IMAGE, RANDOM NUMBERS taken as KEY
STEP 2: FLATTEN the IMAGE into a 1D ARRAY for SUBSTITUTION
STEP 3: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
        QUANTUM LOGISTIC MAPS
STEP 4: FOR k IN RANGE (256):
STEP 5: SUBSTITUTED IMAGE = SBOX [PT[k]]
STEP 6: SUBSTITUTED_ARRAY = NP.ARRAY(PT1).RESHAPE(8,8)
STEP 7: RETURN SUBSTITUTED IMAGE1
STEP 8: FLATTEN the SUBSTITUTED IMAGE1 into a 1D ARRAY for SUBSTITUTION
STEP 9: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
        QUANTUM LOGISTIC MAPS
STEP 10: FOR k IN RANGE (32):
STEP 11: SUBSTITUTED IMAGE1 = SBOX1 [PT1[k]]
STEP 12: SUBSTITUTED_ARRAY1 = NP.ARRAY(PT2)
STEP 13: RETURN ORIGINAL IMAGE
```

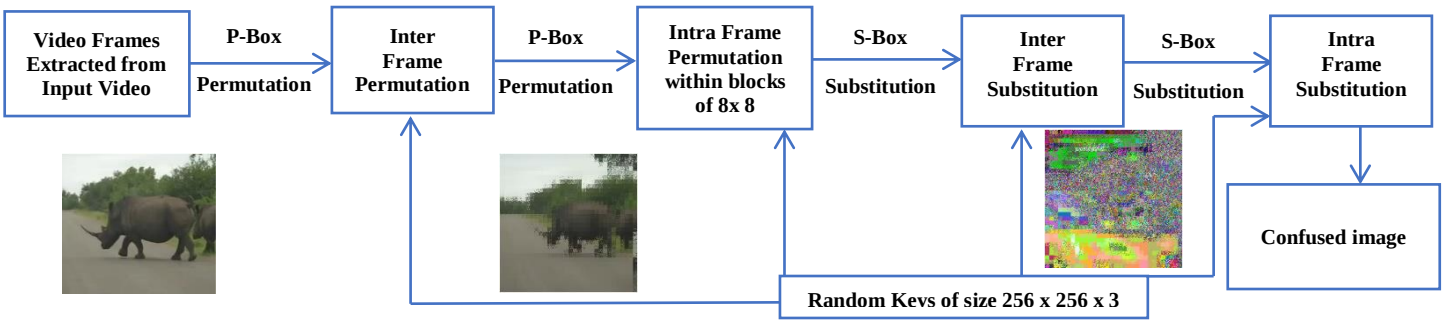


Figure-2: Block Diagram Showing the Confusion Process for encryption

3.3. Diffusion Process: In order to perform the diffusion process the confused image is taken and XORing is done in between the random keys generated from the quantum logistic map of size 256x256x3 and the confused image. After performing the XORing operation encryption process is completed and encrypted frames are received, combining which we get an encrypted video. Complete process of diffusion is shown in Figure-3.

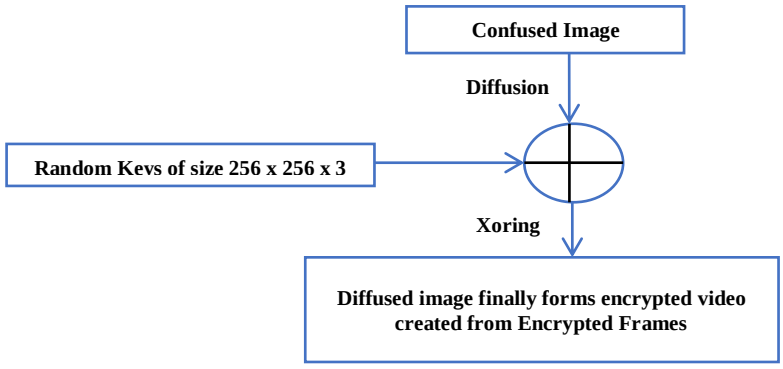


Figure-3: Block Diagram Showing the Diffusion Process for encryption

Algorithm - 6 : Diffusion Process

```
INPUT : SUBSTITUTED IMAGE, Random_Numbers, KEY;
OUTPUT: DIFFUSED IMAGE of SHAPE (HEIGHT, WIDTH, 3)

STEP 1: INPUT taken as PERMUTATED IMAGE,RANDOM NUMBERS taken as KEY
STEP 2: FLATTEN SUBSTITUTED IMAGE AS FLAT_IMAGE = SUBSTITUED_IMAGE.FLATTEN()
STEP 3: FOR ch in range (3):
STEP 4: FOR I in range (frame.shape[0]):
STEP 4: FOR J in range (frame.shape[1]):
STEP  4:  DIFFUSEDFRAME  [I,J,  ch]=  (FRAME  [I  ,  J,  ch]  ^  KEY  [I  %
        KEY.SHAPE[0],J % KEY.SHAPE [1],ch])
STEP 4: DIFFUSED_IMAGE = DIFFUSED_IMAGE.RESHAPE (IMAGE.SHAPE)
STEP 5: RETURN ENCRYPTED_IMAGE
```

Algorithm - 7 : Reverse Diffusion Process

```
INPUT : SUBSTITUTED IMAGE, Random_Numbers, KEY;
OUTPUT: DIFFUSED IMAGE of SHAPE (HEIGHT, WIDTH, 3)

STEP 1: INPUT taken as PERMUTATED IMAGE,RANDOM NUMBERS taken as KEY
STEP 2: FLATTEN SUBSTITUTED IMAGE AS FLAT_IMAGE = SUBSTITUED_IMAGE.FLATTEN()
STEP 3: FOR ch in range (3):
```

```
STEP 4: FOR I in range (frame.shape[0]):
STEP 4: FOR J in range (frame.shape[1]):
STEP 4: DIFFUSEDFRAME [I,J, ch]= (FRAME [I , J, ch] ^ KEY [I %
        KEY.SHAPE[0],J % KEY.SHAPE [1],ch])
STEP 4: DIFFUSED_IMAGE = DIFFUSED_IMAGE.RESHAPE (IMAGE.SHAPE)
STEP 5: RETURN REVERSE_DIFFUSED_IMAGE
```

3.4. Encryption: Figure-4 below shows the block diagram explaining the process used for encrypting the video. First of all the frames are extracted from the input video, after that P-Box is used in order to apply inter frame permutation followed by intra frame permutation within the blocks of 8x8, after that S-box substitution is applied in which inter frame substitution is applied followed by intra frame substitution and then diffusion process is applied in which XORing is done between the generated random keys generated from the quantum logistic map and the permutated, substituted frames (confused image). Random keys are generated using the 12 dimensional quantum based logistic map. Key size is 256x256x3. As a output of entire process described in Figure 2 an encrypted video is created after combining the encrypted frames generated as a combination of confusion and diffusion process.

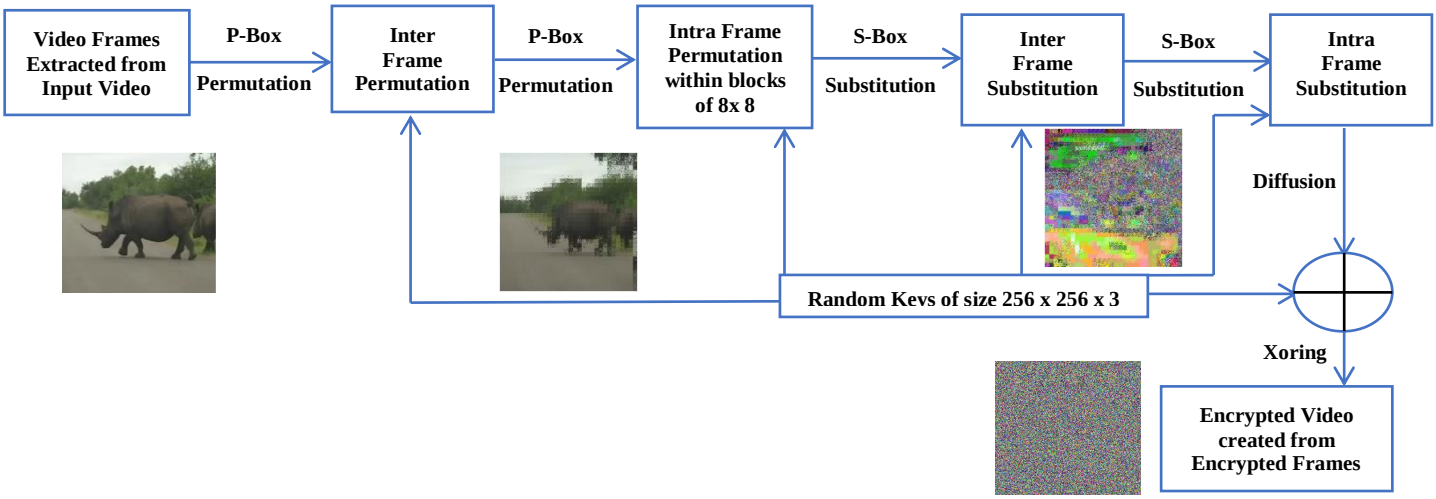


Figure-4: Block Diagram Showing the overall process at encryption side

3.5. Decryption: Figure-5 below shows the block diagram explaining the process used for decrypting the video. Here key thing to note is that proposed scheme is applicable on a video of size 256x256x3 only. First of all the frames are extracted from the encrypted video, after that reverse diffusion is applied by XORing the encrypted video and the generated random keys. After that reverse substitution is applied using the S-Box in which first of all reverse intra frame substitution is applied followed by reverse inter frame substitution. After that reverse intra frame permutation is applied using P-Box followed by reverse intra frame permutation is done. Random keys are generated using the 12 dimensional quantum based logistic map. Key size is 256x256x3. As a output of entire process as described in Figure-5 the original video is created after combining the frames.

Algorithm - 8: Overall Encryption Process

INPUT : Initial_State_Variables = x, y, z, u, r, b; Generated Random_Numbers;
OUTPUT: Encrypted Video after combing the encrypted frames
STEP 1: INITIALIZE AN EMPTY LIST 'random_numbers'[ ] TO STORE THE GENERATED RANDOM NUMBERS
STEP 2: INITIALIZE **'n'** as **'n = HEIGHT * WIDTH * 3'**
STEP 3: FOR EACH **'I'** IN RANGE **'180000 // 3'**:
(a) GENERATE RANDOM NUMBERS x1,y1 and z1 between 0 to 255.
(b) $x1 = (u * 64 * y * (1 - x) + z) \% 1$
$y1 = (u * 64 * y + z * (1-x1)^2) \% 1$
$z1 = (u * (y1 + x1 + 64) * (z - (1/16) * z^3) \% 1$
pnew = (math.sin(r*x*(1-x)*math.pi)+ math.cos(r*x*(1-x)*math.pi))% math.sqrt(2)
(c) key_list.extend ([abs (Num1). abs (Num2). abs (Num3)])
STEP 4: CONVERT **'key_list'** TO A 3D Numpy Array **'Updated_Key'** WITH SHAPE **'(Height, Width, 3)'**

STEP 5: INPUT taken as EXTRACTED IMAGES FROM INPUT VIDEO, RANDOM NUMBERS taken as KEY
STEP 6: FLATTEN IMAGE AS FLAT_IMAGE = IMAGE.FLATTEN()
STEP 7: FOR k IN RANGE (256):
STEP 8: PT1[K] = PT [PBOX [K]]
STEP 9: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 10: RETURN PERMUTATED IMAGE (PT1). RESHAPE (8,8)
STEP 11: FLATTEN IMAGE AS FLAT_IMAGEPT1 = IMAGE.FLATTEN()
STEP 12: FOR k IN RANGE (32):
STEP 13: PT2[K] = PT1 [PBOX [K]]
STEP 14: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 15: RETURN PERMUTATED IMAGE (PT2)
STEP 16: INPUT taken as PERMUTATED IMAGE, RANDOM NUMBERS taken as KEY
STEP 17: FLATTEN the IMAGE into a 1D ARRAY for SUBSTITUTION
STEP 18: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM QUANTUM LOGISTIC MAPS
STEP 19: FOR k IN RANGE (256):
STEP 20: SUBSTITUTED IMAGE = SBOX [PT[k]]
STEP 21: SUBSTITUTED_ARRAY = NP.ARRAY(PT1).RESHAPE(8,8)
STEP 22: RETURN SUBSTITUTED IMAGE1
STEP 23: FLATTEN the SUBSTITUTED IMAGE1 into a 1D ARRAY for SUBSTITUTION
STEP 24: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM QUANTUM LOGISTIC MAPS
STEP 25: FOR k IN RANGE (32):
STEP 26: SUBSTITUTED IMAGE1 = SBOX1 [PT1[k]]
STEP 27: SUBSTITUTED_ARRAY1 = NP.ARRAY(PT2)
STEP 28: RETURN SUBSTITUTED IMAGE2
STEP 29: INPUT taken as PERMUTATED IMAGE,RANDOM NUMBERS taken as KEY
STEP 30: FLATTEN SUBSTITUTED IMAGE AS FLAT_IMAGE = SUBSTITUED_IMAGE.FLATTEN()
STEP 31: FOR ch in range (3):
STEP 32: FOR I in range (frame.shape[0]):
STEP 33: FOR J in range (frame.shape[1]):
STEP 34: DIFFUSEDFRAME [I,J, ch]= (FRAME [I , J, ch] ^ KEY [I % KEY.SHAPE[0],J % KEY.SHAPE [1],ch])
STEP 35: DIFFUSED_IMAGE = DIFFUSED_IMAGE.RESHAPE (IMAGE.SHAPE)
STEP 36: RETURN ENCRYPTED_IMAGE

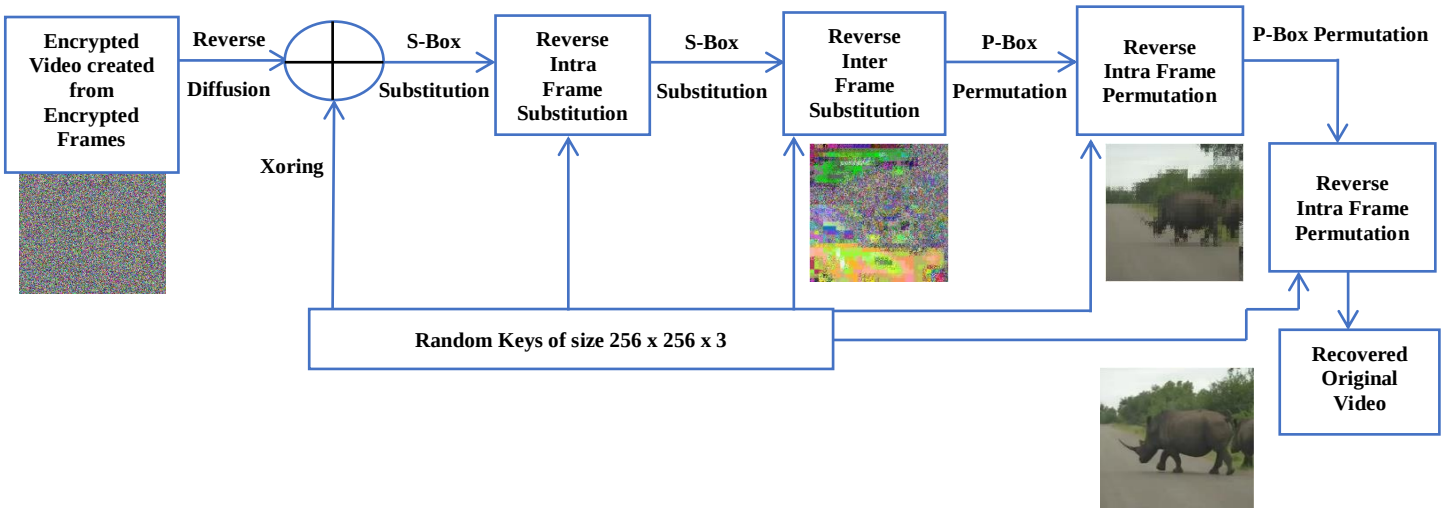


Figure-5: Block Diagram Showing the overall process at the decryption side

Algorithm - 9: Overall Decryption Process

INPUT : Initial_State_Variables = x, y, z, u, r, b; Generated Random_Numbers;
OUTPUT: Decrypted Video after combing the decrypted frames
STEP 1: INITIALIZE AN EMPTY LIST 'random_numbers'[ ] TO STORE THE GENERATED RANDOM NUMBERS
STEP 2: INITIALIZE **'n'** as **'n = HEIGHT * WIDTH * 3'**
STEP 3: FOR EACH **'I'** IN RANGE **'180000 // 3'**:
(a) GENERATE RANDOM NUMBERS x1,y1 and z1 between 0 to 255.
(b) x1 = (u * 64 * y * (1 - x) + z) % 1
y1 = (u * 64 * y + z * $(1-x1)^2$) % 1
z1 = (u * (y1 + x1 + 64) * (z - (1/16) * z^3) % 1
pnew = (math.sin(r*x*(1-x)*math.pi)+ math.cos(r*x*(1-x)*math.pi))% math.sqrt(2)
(c) key_list.extend ([abs (Num1). abs (Num2). abs (Num3)])
STEP 4: CONVERT **'key_list'** TO A 3D Numpy Array **'Updated_Key'** WITH SHAPE **'(Height, Width, 3)'**
STEP 5: INPUT taken as PERMUTATED IMAGE,RANDOM NUMBERS taken as KEY
STEP 6: FLATTEN SUBSTITUTED IMAGE AS FLAT_IMAGE = SUBSTITUED_IMAGE.FLATTEN()
STEP 7: FOR ch in range (3):
STEP 8: FOR I in range (frame.shape[0]):
STEP 9: FOR J in range (frame.shape[1]):
STEP 10: DIFFUSEDFRAME [I,J, ch]= (FRAME [I , J, ch] ^ KEY [I % KEY.SHAPE[0],J % KEY.SHAPE [1],ch])
STEP 11: DIFFUSED_IMAGE = DIFFUSED_IMAGE.RESHAPE (IMAGE.SHAPE)
STEP 12: RETURN REVERSE_DIFFUSED_IMAGE
STEP 13: INPUT taken as DIFFUSED IMAGES FROM INPUT VIDEO, RANDOM NUMBERS taken as KEY
STEP 14: FLATTEN IMAGE AS FLAT_IMAGE = IMAGE.FLATTEN()
STEP 15: FOR k IN RANGE (256):
STEP 16: PT1[K] = PT [PBOX [K]]
STEP 17: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)
STEP 18: RETURN PERMUTATED IMAGE (PT1)
STEP 19: FLATTEN IMAGE AS FLAT_IMAGEPT1 = IMAGE.FLATTEN). RESHAPE (8,8)
STEP 20: FOR k IN RANGE (32):
STEP 21: PT2[K] = PT1 [PBOX [K]]
STEP 22: PERMUTATED_IMAGE = PERMUTATED_IMAGE.RESHAPE (ORIGINAL_SHAPE)

```
STEP 23: RETURN PERMUTATED IMAGE (PT2)
STEP 24: INPUT taken as REVERSE_PERMUTATED_IMAGE, RANDOM NUMBERS taken as KEY
STEP 25: FLATTEN the IMAGE into a 1D ARRAY for SUBSTITUTION
STEP 26: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
         QUANTUM LOGISTIC MAPS
STEP 27: FOR k IN RANGE (256):
STEP 28: SUBSTITUTED IMAGE = SBOX [PT[k]]
STEP 29: SUBSTITUTED_ARRAY = NP.ARRAY(PT1).RESHAPE(8,8)
STEP 30: RETURN SUBSTITUTED IMAGE1
STEP 31: FLATTEN the SUBSTITUTED IMAGE1 into a 1D ARRAY for SUBSTITUTION
STEP 32: SUBSTITUTION is performed using RANDOM NUMBERS GENERATED FROM
         QUANTUM LOGISTIC MAPS
STEP 33: FOR k IN RANGE (32):
STEP 34: SUBSTITUTED IMAGE1 = SBOX1 [PT1[k]]
STEP 35: SUBSTITUTED_ARRAY1 = NP.ARRAY(PT2)
STEP 36: RETURN ORIGINAL IMAGE
```

4. Simulation Parameters

In this section a comprehensive look over the parameters that are employed for experimental purposes. Table-4 outlines the parameters that have role to test the proposed technique, including information about dataset along with the keys utilized in this scheme. Original keys are served for initialing the QLM and to encrypt the videos, Table-4 also details the specifications of machine used for performing the calculations.

Table-4: System Set-up Parameters

S.No.	Name of Parameter	Values
1	Set used for testing	4
2	Total Number of Videos	4
3	Size of Image frames extracted from videos	256 x 256 x 3
4	Types of Images used	Colored
5	Programming Language Used	Python 3.11
6	Intial Parameters for QLM	x_n = 0.19274124785, y_n = 0.71213452355, z_n = 0.54345235464 & u = 2.1234432441
7	Number of Keys Generated from GAN	150000
8	Processor of Hardware	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.60 GHz
9	RAM	8.00 GB (7.89 GB usable)
10	CPU	64-bit operating system, x64-based processor with Windows 10 Pro
11	Model	DESKTOP-SJ75BC4
12	Dataset	https://drive.google.com/drive/folders/183rWwvu2ESQt2Fii1vqgcNovliGcGKzE?usp=sharing

5. Results (Key Findings from Proposed Scheme)

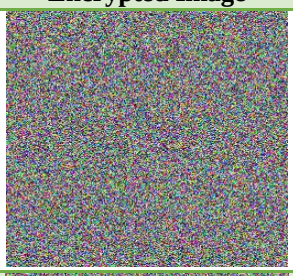
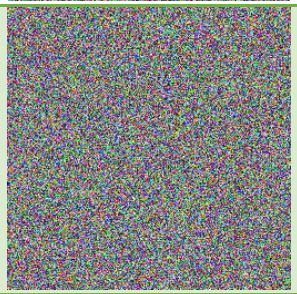
Here we present a thorough analysis and examination of outcomes obtained during this research. The results are computed on dataset, containing 4 standard videos in ".mp4" format with dimensions of (256 × 256 × 3). This section is structured as: 5.1 Screenshots, which includes different plots derived from original, encrypted, and decrypted images of various sizes for better understanding; 5.2 Visual Analysis, providing insights into the robust technique proposed for encryption through inspecting visually and then comparing with other image encryption techniques in available literature; 5.3 Statistical Analysis, discussing results that are dependent on correlation coefficients to understand pixel relationships post-encryption; 5.4 Quantitative Analysis, offering comparison between proposed technique with others mentioned in this literature based on Bit Error Rate (BER), Mean Square Error (MSE), Entropy (H), Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR) 5.5 Other Attack Analysis; and 5.6 Randomness and Robustness Test Analysis.

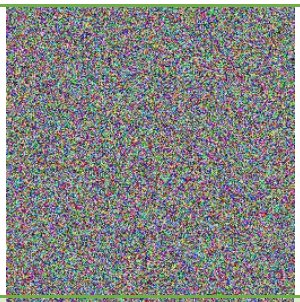
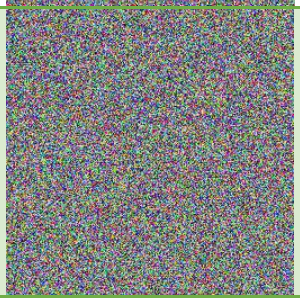
The results below demonstrate that the proposed encryption algorithm achieves high robustness against various attacks, with an average PSNR near to 8 dB entropy 7.99, indicating minimal visual distortion. The NPCR and UACI values exceed 99.60% and 33.33%, respectively, showing strong resistance to differential attacks. Additionally, encryption time analysis confirms the scheme's suitability for real-time applications without compromising security.

5.1. Visual Analysis

Table-5 illustrates the proposed encryption scheme's effectiveness for videos of size (256, 256, 3) from which we extract the frames and then encrypt them. Original image, encrypted output, as well as the final decrypted frame is shown, with the decrypted frame closely matching the original, demonstrating successful encryption and decryption processes.

Table-5: Visual Analysis showing original images, encrypted images and decrypted images

	Proposed Model		
Video Name	**Original Image**	**Encrypted Image**	**Decrypted Image**
Video 1			
Video 2			

Video 3			
Video 4			

5.2. Statistical Analysis

The analysis involves evaluating the relationship between original and encrypted images using metrics like histogram analysis and correlation coefficients to measure the scheme's robustness. Such evaluations help identify residual similarities between neighboring pixels, ensuring the encryption technique effectively conceals image details.

5.2.1. Correlation Coefficient

Results for frames from the 256 x 256 x 3 video set, detailed in Table-5, show the effectiveness of the encryption scheme, with correlation coefficient values indicating strong de-correlation between original and encrypted images. The correlation coefficient, *(r)*, for image encryption, is given by:

$$r = \frac{E\,[(X - \mu x)(Y - \mu y)\,]}{\sigma x\ \ \sigma y}$$

Where: E denotes the expectation operator,

Y as well as X are the gray scale value of 2 corresponding pixels in original and encrypted images, respectively,

μ_x and μ_y represent the mean values of x and y,

σx And σy are the standard deviations of x and y.

This expression measures correlation between pixel values in original as well as encrypted images, providing a quantitative evaluation of the encryption algorithm's effectiveness. Results of correlation coefficients of videos are compiled in Table-6.

Table-6: Correlation Coefficients of videos

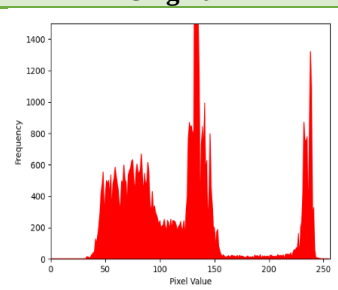
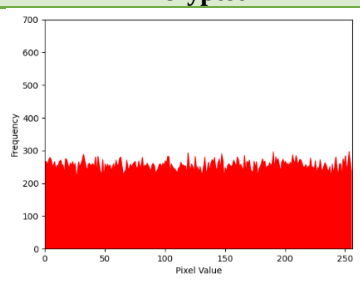
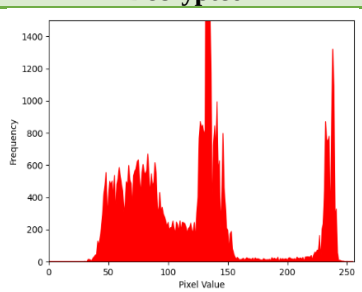
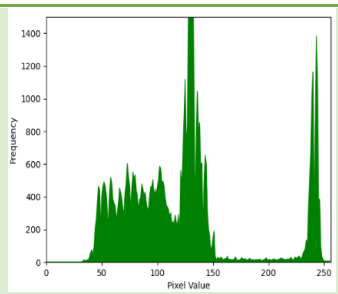
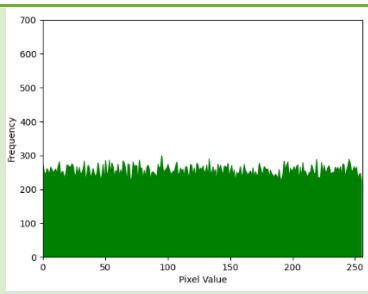
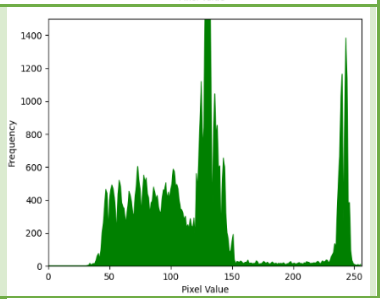
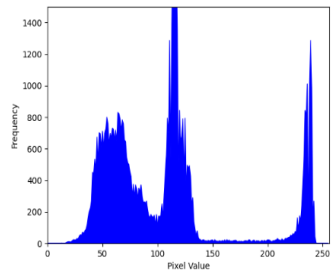
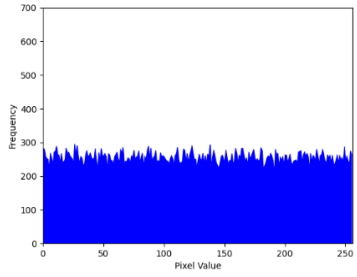
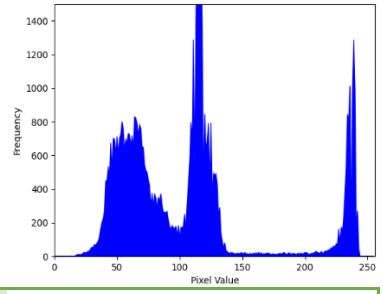
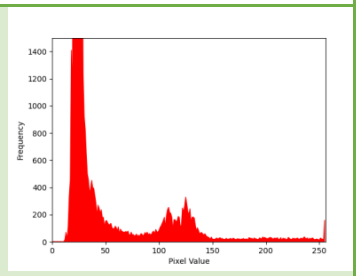
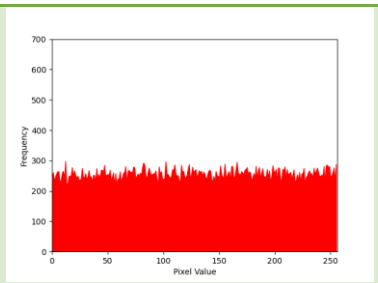
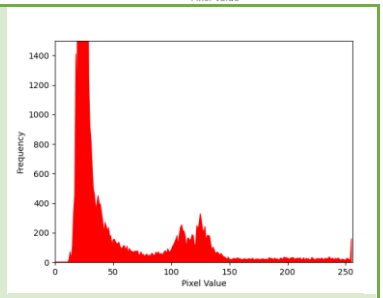
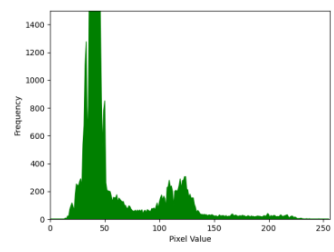
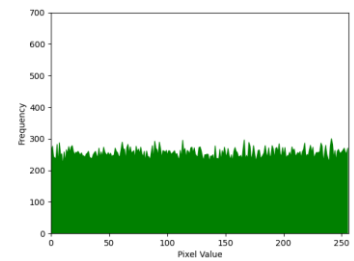
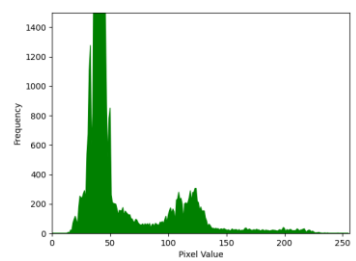
	Ref. [46]	**Proposed Model**
Video 1	0.0073778	0.0022667
Video 2	0.00800012	0.0023444
Video 3	0.0268000	0.0021222
Video 4	0.002718	0.0023111

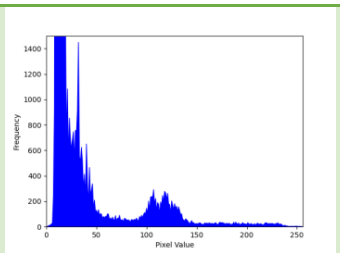
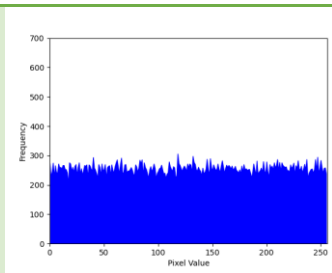
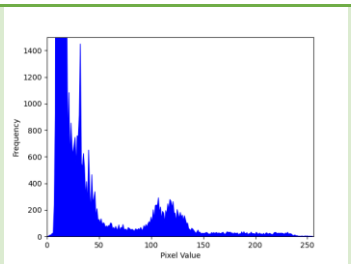
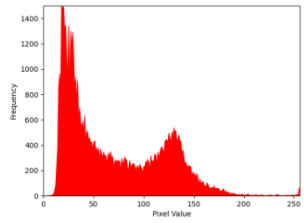
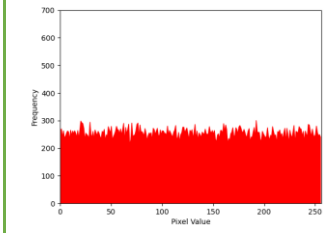
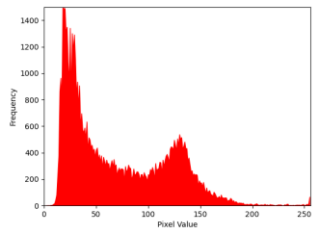
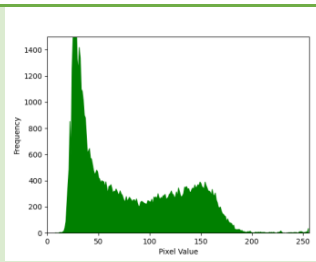
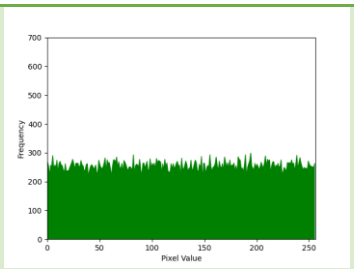
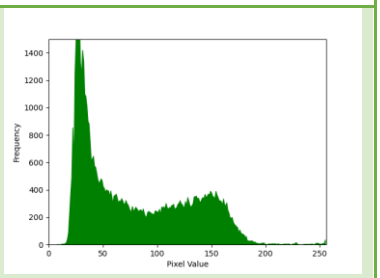
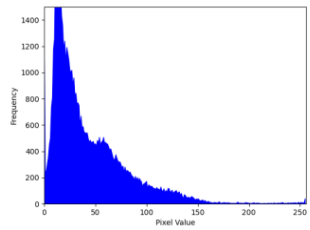
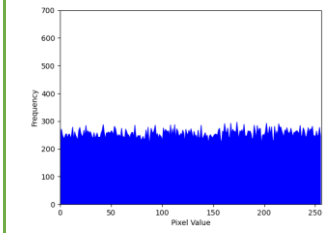
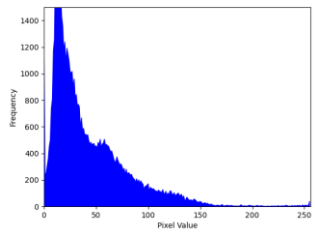
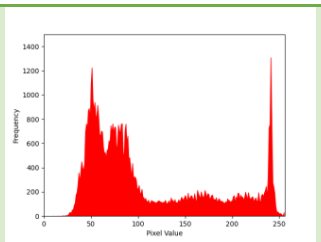
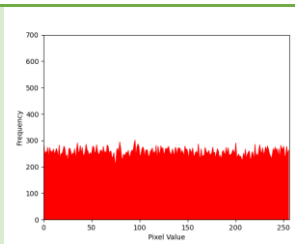
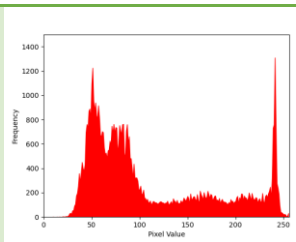
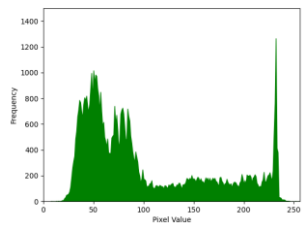
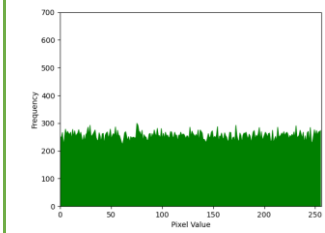
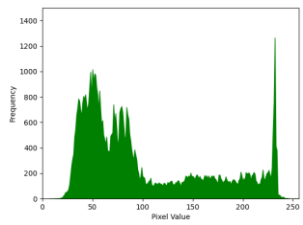
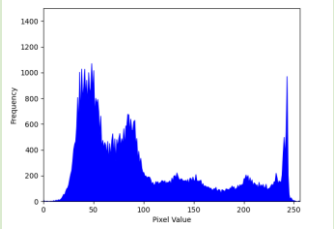
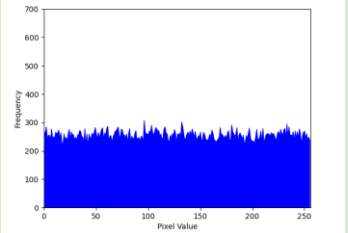
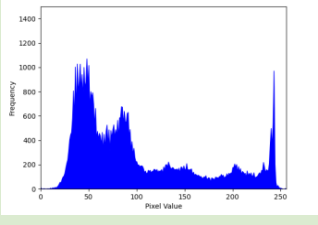
Table-6 demonstrates that the proposed model achieves near-zero correlation in horizontal, vertical, and diagonal directions, indicating a highly uncorrelated encrypted output. The correlation coefficients surpass those of existing image processing and deep learning techniques, enhancing the encryption scheme's resistance to attacks. Average value of correlation coefficient for the proposed model is 0.002261 which is nearly equal to 0 hence achieved value of correlation is good.

5.2.2. Histogram Analysis

Table-7 presents the histogram analysis for images of size (256, 256, 3), illustrating the distribution of pixel frequency before and after encryption. The results show that the encrypted image's histogram is uniform, indicating effective encryption by significantly altering the original pixel value distribution.

Table-7: Histogram Graphical Analysis of videos

	Channel	Proposed Model		
		Original	Encrypted	Decrypted
Video 1	R			
	G			
	B			
Video 2	R			
	G			

	B			
Video 3	**R**			
	G			
	B			
Video 4	**R**			
	G			
	B			

The experimental results indicate that the proposed model achieves a uniform frequency distribution, producing an almost straight-line histogram—an ideal characteristic for effective image encryption. The original image was successfully recovered, demonstrating the scheme's reliability in maintaining data integrity.

5.3. Quantitative Analysis

The quantitative analysis demonstrated variations in algorithm performance across image data changes, with metrics including Bit Error Rate (BER), Mean Square Error (MSE), Entropy (H), Structural Similarity Index (SSIM), and Peak Signal to Noise Ratio (PSNR) revealing significant insights. These measures effectively captured the encryption algorithm's impact on image quality and data integrity.

5.3.1. Mean Square Error (MSE)

Table-8 presents the Mean Squared Error (MSE) results for 256 x 256 x 3 images, highlighting a comparative analysis of various techniques from the literature. The MSE between the original and encrypted images, MSE(O, E), for images with dimensions (i, j, k) is computed using the defined formula.:

$$MSE = \frac{1}{MNL}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{L}[I(i,j,k) - I'(i,j,k)]^2$$

In this context, M, N, and L denote the dimensions of the 3D image, while I(i, j, k) and I'(i, j, k) represent the pixel values at coordinates (i, j, k) in the original and encrypted 3D images, respectively. The formula computes the average squared difference between the corresponding pixels, quantifying the error introduced by the encryption process.

Table-8: Mean Square Error Analysis (MSE) of videos

		Proposed Model
Video 1	**R**	10551.29
	G	10551.33
	B	10547.66
Video 2	**R**	10549.01
	G	10546.91
	B	1055.15
Video 3	**R**	10549.27
	G	10552.84
	B	10547.72
Video 4	**R**	10554.13
	G	10554.21
	B	10546.97

The proposed scheme yields an average Mean Squared Error (MSE) of 9758.874, outperforming many traditional methods. This improvement highlights the effectiveness of our approach in enhancing performance metrics.

5.3.2. Peak Signal to Noise Ratio (PSNR)

Table-9 presents the Peak Signal to Noise Ratio (PSNR) results calculated for the image set, offering a detailed comparison of PSNR values across different methodologies in the literature. The mathematical expression for PSNR in the context of 3D image encryption is critical for assessing the performance and robustness of the proposed techniques:

$$PSNR = 10.\log_{10}\frac{(MAX)^2}{MSE}$$

MAX denotes the highest pixel value achievable in the image, while MSE represents the Mean Squared Error between the original and encrypted images. This PSNR formula quantifies the peak distortion between the original and encrypted 3D images, serving as a critical metric for evaluating the effectiveness of the encryption quality.

Table-9: Peak Signal to Noise Ratio Analysis (PSNR) of videos

	Channel	Ref. [46]	Proposed Model
Video 1	**R**	8.77	7.8978
	G	8.81	7.8977
	B	8.48	7.8993
Video 2	**R**	6.59	7.8987
	G	7.43	7.8996
	B	6.22	7.8978
Video 3	**R**	7.71	7.8986
	G	7.99	7.8971
	B	6.69	7.8992
Video 4	**R**	7.94	7.8966
	G	7.94	7.8966
	B	7.89	7.8995

The proposed model attains an average Peak Signal to Noise Ratio (PSNR) of 7.8982, indicating that performance is better as compared to many techniques reported in existing literature.

5.3.3. Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) serves as a robust metric for assessing image fidelity, particularly in terms of signal structure retention. As illustrated in Table-10, the SSIM results computed for the image set highlight the importance of evaluating structural, luminance, and contrast components between corresponding blocks of original and encrypted images.

$$SSIM\ (X,Y) = \frac{(2\mu x\mu y\ + C1)(2\sigma xy + C2)}{(\mu x^2 + \mu y^2 + C1) * (\sigma x^2 + \sigma y^2 + C2)}$$

Where: μx and μy are value of mean intensities of image blocks *Y* and *X*.

σx^2 and σy^2 are the variances of image blocks *Y* and *X*.

σxy is covariance between image blocks *Y* and *X*.

$C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are small constants to stabilize division with a weak denominator, where *L* is dynamic pixel range, K_1 and K_2 are scalar constants.

Table-10: Structural Similarity Index Analysis (SSIM) of videos

	Channel	Ref. [46]	Proposed Model
Video 1	**R**	0.0093	0.0098
	G	0.0097	0.0099
	B	0.0087	0.0097
Video 2	**R**	0.0039	0.0057
	G	0.0069	0.0071
	B	0.0049	0.0049
Video 3	**R**	0.0066	0.0069
	G	0.0080	0.0072
	B	0.0044	0.0051
Video 4	**R**	0.0097	0.0086
	G	0.0077	0.0084
	B	0.0095	0.0081

Table-10 demonstrates that the original input image and the encrypted image exhibit no structural similarities, indicating a successful encryption process. This lack of similarity is essential for validating the effectiveness of the model as a robust encryption solution. Average value of SSIM for the proposed model is 0.007617 which is nearly equal to 0.

5.3.4. Entropy (H) Values

Table-11 presents the entropy results calculated for the image set, offering a comprehensive comparison of entropy values across different methodologies documented in the literature. In this context, let p_i represent the probability of the occurrence of the i^{th} intensity value within the encrypted 3D image, with the entropy H quantifying the information content of the encrypted data.

$$H = \sum_{i=1}^{L} pi \ \log_2(pi)$$

In this formula, L represents the total number of intensity levels in the image, while p_i indicates the probability of the i^{th} intensity value occurring in the encrypted image, calculated as $pi = \frac{ni}{N}$, where n_i is the frequency of the i^{th} intensity value and N is the total pixel count. This approach assesses the randomness and unpredictability of intensity values in encrypted images, ensuring its relevance for high-level research discussions."

Table-11: Entropy Analysis (H) of videos

	Ref. [46]			Proposed Model		
	Original frame	**Encrypted Frame**	**Decrypted Frame**	**Original frame**	**Encrypted Frame**	**Decrypted Frame**
Video 1	6.8859	7.9976	6.8859	6.905667	7.997167	6.905667
Video 2	5.616433	7.997433	5.616433	5.9157	7.997233	5.9157
Video 3	6.9834	7.9973	6.9834	6.859767	7.9973	6.859767
Video 4	7.370733	7.997867	7.370733	7.3882	7.9972	7.3882

The proposed model attains an average entropy value of 7.997225, significantly exceeding those reported in existing literature. This high entropy demonstrates the model's effectiveness in ensuring data randomness and security.

5.4. Differential Analysis Attack

The results of the differential attack analysis are summarized in the tables below, indicating that the proposed model effectively withstands both the Uniform Average Change Intensity (UACI) and the Number of Pixels Change Rate (NPCR) tests. This successful performance underscores the robustness of the model against differential attacks.

5.4.1. Number of Pixel change rate (NPCR)

NPCR (Number of Pixel Change Rate) is an essential metric for evaluating a system's resistance to differential attacks, reflecting the rate at which pixel values in the encrypted image change following a single-pixel modification in the original image. As shown in Table-12, the NPCR test results underscore the randomness and robustness of the proposed model against potential security threats.

$$NPCR = \frac{1}{N}\sum_{i=1}^{N}\frac{|Xi \oplus Yi|}{L-1} * 100$$

Where: N is the total pixels in images X and Y.

Xi and Yi represent intensity values of the i^{th} pixel in images X and Y.

$\oplus$ denotes the bitwise XOR operation.

L is max possible intensity value (e.g., 256 of an 8-bit image).

This expression quantifies the percentage of pixels that differ between original as well as encrypted images, normalized by total number of pixels and intensity range.

Table-12: Number of Pixel Change Rate (NPCR) Analysis of videos

	Channel	Ref. [46]	Proposed Model
Video 1	**R**	99.6068	99.6077
	G	99.6237	99.6134
	B	99.6250	99.6073
Video 2	**R**	99.5827	99.6113
	G	99.6094	99.6072
	B	99.5605	99.6091
Video 3	**R**	99.5694	99.6099
	G	99.6094	99.6071
	B	99.6212	99.6110
Video 4	**R**	99.6286	99.6089
	G	99.5811	99.6129
	B	99.6445	99.6122

The average NPCR values consistently surpass the theoretical critical thresholds across all evaluated levels, demonstrating the robustness of the proposed schemes. This outcome confirms that the encryption methods effectively withstand cryptographic analysis. Average value of NPCR of proposed model is 99.60983 which tells that NPCR test is qualified

5.4.2. Unified Average Change Intensity (UACI)

The UACI metric, detailed in Table-12, is essential for assessing the resilience of the proposed model against differential attacks, indicating its robustness upon the modification of a single pixel in the original image. By comparing the original image X with the encrypted image Y, the UACI is calculated to quantify the average intensity variation, underscoring the encryption's effectiveness.

$$UACI = \frac{1}{N}\sum_{i=1}^{N}\frac{|Xi - Yi|}{L-1} * 100$$

Where: N is total pixels in images X and Y

Xi and Yi denote intensity values of i^{th} pixel in images X and Y.

L is max possible intensity value (e.g., 256 for 8-bit image).

This formula evaluates average intensity difference among corresponding pixels of original as well as encrypted images, normalized by intensity range.

Table-13: Unified Average Change Intensity Analysis (UACI) of videos

	Channel	Ref. [46]	Proposed Model
Video 1	**R**	33.5630	33.3840
	G	33.3876	33.3637
	B	33.3666	33.3782
Video 2	**R**	33.4939	33.7923
	G	33.5092	33.8241
	B	33.5853	33.8137
Video 3	**R**	33.3913	33.8821
	G	33.2731	34.0248
	B	33.5992	33.9635
Video 4	**R**	33.5842	33.6424
	G	33.4908	33.6938
	B	33.4451	33.7069

The average UACI values consistently fall within the theoretically acceptable range across all tested levels, demonstrating that the proposed schemes effectively meet the required security benchmarks. This success indicates a robust performance in achieving the desired level of unpredictability in the encryption process. From table-13 average value of UACI is 33.70579 which is above 33.33% this means that the proposed model qualifies the differential attack (UACI).

5.5. Robustness /Other Attack Analysis

To validate the robustness of the image encryption model in real-world scenarios, a series of tests were conducted to assess its resilience against various disturbances, including noise interference and malicious attacks. The outcomes of these evaluations, which encompass noise attacks, geometric attacks, key space analysis, and anti-occlusion attacks, are presented in the following tables.

5.5.1 Noise Attacks

When encrypted data is transmitted over an open channel, it is susceptible to various noise types, such as salt and pepper, which can hinder the decryption process. The results presented in Table-14 highlight the algorithm's robustness when subjected to noise attacks, specifically with 10%, 25%, and 50% of pixels altered across the three color planes prior to decryption. Table-15 and Table-16 consists of the values of PSNR and BER after the application of Gaussian noise and Salt and Pepper noise respectively.

Table-14: Visual Analysis after application of Noise Attacks on videos

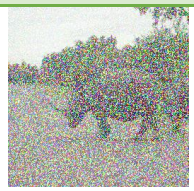
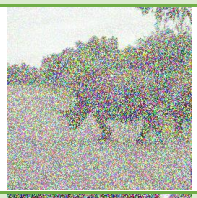
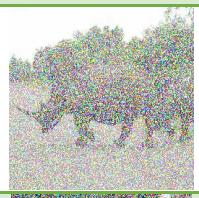
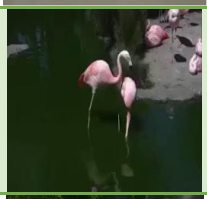
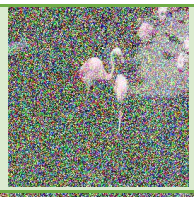
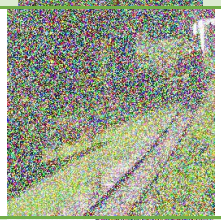
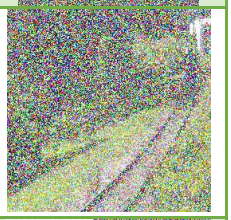
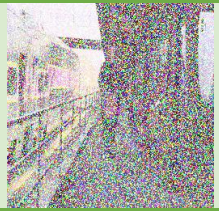
	Proposed Model			
	Original Image	**10% Noise**	**25% Noise**	**50% Noise**
Video 1				
Video 2				
Video 3				
Video 4				

Table-15: Values of PSNR and BER after Application of Gaussian Noise on videos

	Proposed Model (Gaussian Noise)					
	10% Noise		25% Noise		50% Noise	
	BER	PSNR	BER	PSNR	BER	PSNR
Video 1	75.19	32.0265	84.22	30.9844	88.89	30.1545
Video 2	75.12	31.2068	84.04	30.2692	88.64	29.5159
Video 3	74.95	31.5332	84.01	30.5625	88.68	29.7726
Video 4	75.00	31.4667	83.99	30.5098	88.66	29.7180

Table-16: Values of PSNR and BER after Application of Salt and Pepper Noise on videos

	Proposed Model (Salt and Pepper Noise)					
	10% Noise		25% Noise		50% Noise	
	BER	PSNR	BER	PSNR	BER	PSNR
Video 1	25.87	34.24045	52.49	31.1647	77.33	29.4885
Video 2	25.94	33.16424	52.56	30.0972	77.39	28.4183
Video 3	25.79	33.57753	52.40	30.5226	76.99	28.3649
Video 4	25.77	33.38073	52.49	30.3016	77.32	28.6036

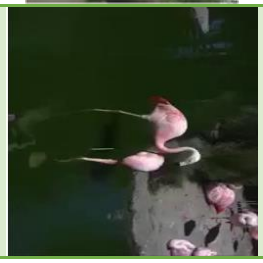
In Table-15 as well as in Table-16 the values of PSNR and BER are as per their desired values and are in range when noise attacks are applied on the encrypted image.

5.5.2 Geometrical Attacks

This research investigates the resilience of the proposed algorithm against geometric distortions, including various rotational and flipping attacks, as detailed in Table-17 and Table-19. The results, illustrated in Tables-18 and Table-21, show that the proposed model effectively withstands rotational as well as flip attacks, outperforming existing approaches in terms of Peak Signal to Noise Ratio (PSNR) and Bit Error Rate (BER).

Rotational Attack: The encrypted image undergoes rotation prior to transmission, with various rotational attacks illustrated in Table-17. Tables-18 the PSNR and Bit Error Rate (BER) values for the dataset, demonstrating that the proposed model effectively withstands rotational attacks while surpassing existing models in both PSNR and BER performance.

Table-17: Visual Analysis after application of Rotational Attacks on videos

	Proposed Model		
	Original Image	Clockwise	Anti Clockwise
Video 1			
Video 2			

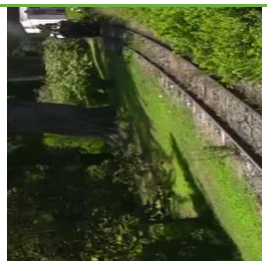
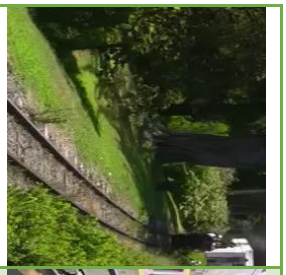
Video 3			
Video 4			

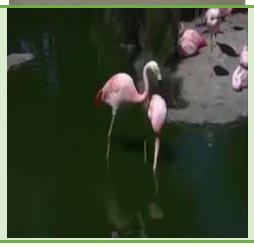
Table-18: Values of PSNR and BER after Application of Rotational Attack on videos

	Proposed Model			
	Clockwise		**Anti Clockwise**	
	BER	**PSNR**	**BER**	**PSNR**
Video 1	99.21	7.7752	99.59	7.7398
Video 2	99.22	7.7583	99.64	7.7641
Video 3	99.20	7.7589	99.60	7.7611
Video 4	99.23	7.7892	99.63	7.7901

In Table-18 the values of PSNR and BER are as per their desired values and are in range when rotational attacks are applied on the encrypted image.

Flip Attack: The encrypted image undergoes a flipping process prior to transmission, with different types of flip attacks outlined in Table-19. The proposed model demonstrates robust resistance against these attacks, outperforming existing models, as evidenced by the PSNR and Bit Error Rate (BER) values presented in Tables-20.

Table-19: Visual Analysis after application of Flip Attacks on videos

	Proposed Model			
	Original Image	**Vertical Direction**	**Horizontal Direction**	**Both Vertical & Horizontal Direction**
Video 1				
Video 2				

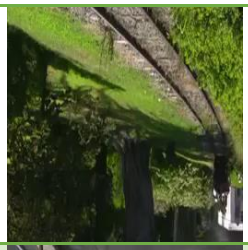
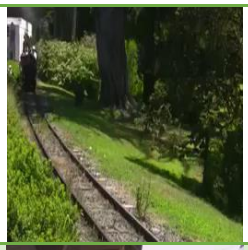
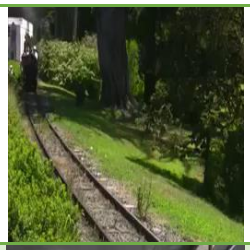
Video 3				
Video 4				

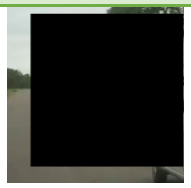
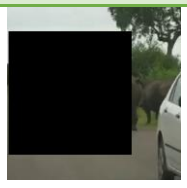
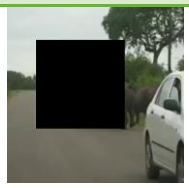
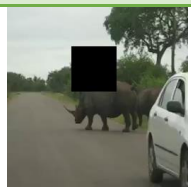
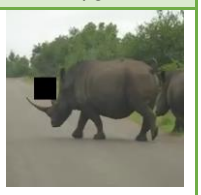
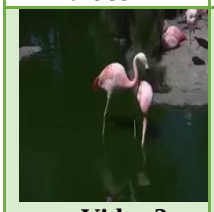
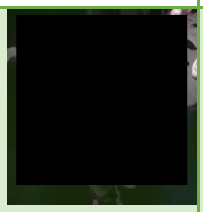
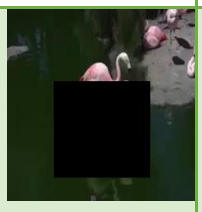
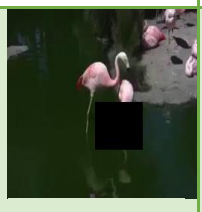
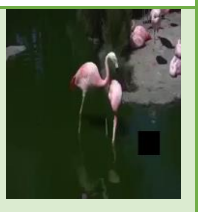
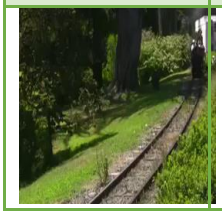
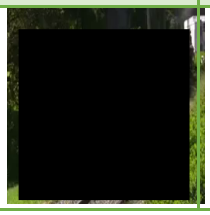
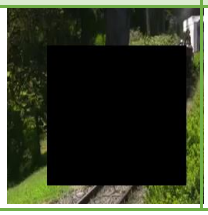
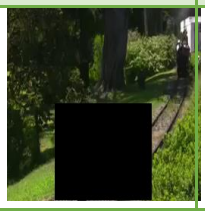
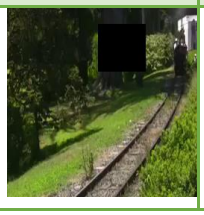
Table-20: Values of PSNR and BER after Application of flip Attack on videos

	Proposed Model					
	Vertical Direction		Horizontal Direction		Both Vertical & Horizontal Direction	
	BER	PSNR	BER	PSNR	BER	PSNR
Video 1	99.21	12.4957	99.59	11.9127	99.65	12.6386
Video 2	99.38	12.6019	99.33	11.8903	99.70	12.6664
Video 3	99.13	11.0110	99.13	10.5710	87.79	13.1937
Video 4	98.72	9.53624	98.77	7.90777	98.64	7.84200

In Table-20 the values of PSNR and BER are as per their desired values and are in range when flip attack is applied on the encrypted image.

Anti Occlusion Attack: To assess the proposed algorithm's robustness against data loss due to occlusion, systematic tests were conducted at intervals of 1/64, 1/16, 1/4, and 1/2, revealing effective data recovery from all occluded encrypted images. The comparative analysis presented in Tables-21 and Table-22 highlights the algorithm's superior performance in terms of Bit Error Rate (BER) and Peak Signal to Noise Ratio (PSNR) against existing methodologies in the literature.

Table-21: Visual Analysis after application of Anti occlusion attack on videos

	Proposed Model				
	3/4	1/2	1/4	1/16	1/64
Video 1						
Video 2						
						

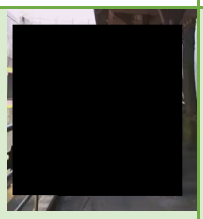
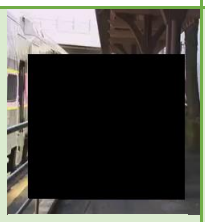
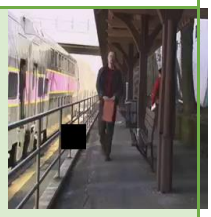
Video 3					
					
Video 4					

Table-22: Values of PSNR and BER after Application of Anti occlusion Attack on videos

	Proposed Model									
	3/4		1/2		1/4		1/16		1/64	
	BER	PSNR	BER	PSNR	BER	PSNR	BER	PSNR	BER	PSNR
Video 1	74.52	28.9525	49.99	30.6594	25.00	33.6491	6.25	39.3952	1.56	45.6644
Video 2	74.51	28.4979	49.98	30.3113	24.99	33.3256	6.25	39.1939	1.56	44.7594
Video 3	74.07	29.0242	49.68	30.7994	24.90	33.5333	6.22	39.75219	1.54	46.1866
Video 4	74.52	28.7773	49.98	30.5405	25.00	33.4413	6.25	39.3837	1.56	45.3099

In Table-22 the values of PSNR and BER are as per their desired values and are in range when anti occlusion attack is applied on the encrypted image.

6. Comparison with State of Art Techniques

Table-23 illustrates that the proposed model outperforms established models across multiple dimensions, encompassing quantitative analysis through metrics such as Bit Error Rate (BER), Structural Similarity Index (SSIM), and Entropy (H). Additionally, it demonstrates superior performance in visual analysis, differential attack assessments, statistical evaluations, randomness tests, and various other attack analyses.

Table-23: Comparative Study with referenced Techniques

S.No.	Reference Name & number (Year)	VA	SA		QA				DAA	
			CC	HGA	MSE	PSNR	H	SSIM	NPCR	UACI
1	[66]	✓	0.0083	✓	----	----	7.99	----	99.62	33.49
2	[67]	✓	----	✓	----	----	7.99	----	99.16	33.57
3	[68]	✓	0.00490	✓	----	----	7.99	----	99.61	33.46
4	[69]	✓	----	✓	----	----	7.99	----	----	----
5	[70]	✓	0.00506	✓	----	----	7.99	----	99.86	33.56
6	[46]	✓	0.011224	✓	----	7.705	7.9975	0.007442	99.60519	33.47411
7	Proposed Model	✓	0.002261	✓	9758.874	7.8995	7.99722	0.007617	99.60983	33.70579

7. Conclusion

In this study, we present a secure cryptographic model that integrates uniquely designed diffusion; permutation as well as substitution boxes utilizing keys generated using quantum

logistic maps (QLMs). The model employs QLMs for encryption key generation. The results indicate that our proposed model consistently outperforms existing methodologies across all performance metrics. The keys generated using QLMs successfully passes NIST tests and demonstrates resilience against prevalent attacks documented in the literature. Moreover, the algorithm achieves optimal values in UACI, VCC, NPCR, DCC, HCC, BER, MSE, H, SSIM, and PSNR. The model ensures an exceptional balance of key sensitivity and BFST values approaching 1, facilitated by the integration of diffusion; permutation as well as substitution. This work meets all critical criteria for a robust image encryption algorithm.

Declarations

Conflict of interest: The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication. Also the data compiled is part of original work and is not generated using any AI Tool.

Data availability: The data used to support the findings of this study are available from the corresponding author upon request.

References

1. Verizon DBIR 2024: https://www.verizon.com/business/resources/reports/dbir/
2. Cisco 2024 Cybersecurity Report: https://www.cisco.com/c/en/us/products/security/security-reports.html
3. ENISA Threat Landscape 2023: https://www.enisa.europa.eu/publications
4. M. Jain and A. Sharma, "Review and performance analysis of modern cryptographic techniques for secure data communication," *IEEE Access*, vol. 8, pp. 194379-194397, 2020.
5. K. S. Kumar et al., "Improved elliptic curve cryptography for data encryption in cloud computing," *IEEE Access*, vol. 9, pp. 1781-1790, 2021.
6. H. R. Nemati, M. Bahrami, and E. Alizadeh, "An efficient implementation of RSA cryptosystem using residue number system for secure communication," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2555-2564, Jun. 2021.
7. R. Basnet, D. L. Khosla, and N. R. Gawande, "Comparison of lightweight encryption algorithms in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9188-9197, Oct. 2020.
8. A. Kumar, S. Jain, and P. Sharma, "A new hybrid encryption algorithm based on DES and RSA for secure data transmission in IoT," *IEEE Access*, vol. 8, pp. 23058-23067, 2020.
9. M. Banik et al., "Efficient and secure identity-based encryption with equality test for industrial IoT environments," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3521-3532, Mar. 2022.
10. S. Kundu and S. Bandyopadhyay, "Secure image transmission using advanced encryption standard (AES) for wireless body area networks," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 311-319, Nov. 2020.
11. B. Li et al., "Efficient lightweight encryption with optimization strategies in IoT devices," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2055-2067, Feb. 2022.
12. J. Liu, X. Zhou, and Y. Feng, "Implementation of a hybrid encryption scheme based on the optimization of traditional algorithms in smart environments," *IEEE Access*, vol. 10, pp. 1225-1235, 2022.
13. M. Wang, X. Zhang, and J. Liu, "A novel color image encryption algorithm based on a fractional-order chaotic system and S-box scrambling," *IEEE Access*, vol. 8, pp. 21863-21876, 2020.
14. S. Ullah et al., "An efficient image encryption scheme using chaotic maps and fractional Fourier transform," *IEEE Access*, vol. 8, pp. 184382-184396, 2020.
15. W. Hu, L. Liu, and J. Ren, "A chaos-based encryption algorithm for enhancing the security of medical images in healthcare applications," *IEEE Access*, vol. 9, pp. 112907-112918, 2021.

16. H. Zhu, Z. Chen, and L. Li, "Logistic chaotic map-based cryptographic scheme for lightweight encryption of multimedia content," *IEEE Transactions on Multimedia*, vol. 24, pp. 2028-2037, May 2022.
17. Y. Song, X. Hu, and Y. Zhao, "New chaos-based dynamic encryption algorithm for secure image communication," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 412-424, 2022.
18. S. Yuan and Z. Liu, "A chaotic map image encryption scheme based on fractional calculus," *IEEE Access*, vol. 9, pp. 190643-190654, 2021.
19. C. Li, J. Liu, and Y. Song, "Logistic map-enhanced cryptographic framework for real-time multimedia data protection," *IEEE Transactions on Multimedia*, vol. 26, pp. 3295-3308, Sept. 2024.
20. R. Raj and S. Satapathy, "A hybrid image encryption scheme using a chaotic map and DNA computing," *IEEE Access*, vol. 10, pp. 110344-110355, 2022.
21. X. Wang and X. He, "Enhanced multimedia encryption algorithm based on logistic and tent maps," *IEEE Transactions on Multimedia*, vol. 26, pp. 3503-3515, 2024.
22. Y. Cao et al., "Experimental quantum key distribution using weak coherent states with enhanced security," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 6, pp. 1-8, Nov.-Dec. 2021.
23. F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," *Reviews of Modern Physics*, vol. 92, no. 2, pp. 025002-025078, Apr. 2020.
24. L. Banchi, S. Pirandola, and P. Perinotti, "Quantum algorithms for cryptography and cybersecurity," *IEEE Transactions on Information Theory*, vol. 68, no. 3, pp. 1896-1915, Mar. 2022.
25. H. Y. Qi, J. Xu, and Y. Wang, "Quantum secure direct communication using high-dimensional entanglement," *IEEE Access*, vol. 9, pp. 87545-87553, 2021.
26. T. Lee, C. Cui, and W.-Y. Hwang, "Efficient quantum key distribution with double-checking of eavesdropping," *IEEE Transactions on Information Theory*, vol. 67, no. 12, pp. 7824-7833, Dec. 2021.
27. S. Pirandola et al., "Quantum cryptography: Advances and practical challenges," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 2, pp. 1-18, 2021.
28. X. Li et al., "Continuous-variable quantum key distribution with coherent detection," *IEEE Transactions on Information Theory*, vol. 68, no. 4, pp. 2345-2360, Apr. 2022.
29. J. Yin et al., "Realizing satellite-based quantum key distribution with entanglement," *Nature*, vol. 582, pp. 501-505, 2020.
30. A. Pathak, "Recent advances in quantum cryptography," *IEEE Transactions on Quantum Engineering*, vol. 4, no. 1, pp. 1-16, Feb. 2023.

31. Wang, H., Zhang, L., & Li, X. "An improved image encryption algorithm based on chaotic maps and DNA sequence operations." *IEEE Access*, vol. 11, pp. 4983-4995, 2023.
32. Chen, Q., & Gao, Y. "A novel video encryption method combining chaotic system and block-level processing for real-time applications." *Multimedia Tools and Applications, Springer*, vol. 82, pp. 12345-12368, 2024.
33. Liu, J., Zhao, L., & Huang, W. "Enhanced encryption algorithm for secure multimedia transmission based on hyper-chaotic maps." *IEEE Transactions on Multimedia*, vol. 26, pp. 3298-3309, 2024.
34. Patel, A., & Singh, K. "Performance analysis of steganography techniques for secure data communication." *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1223-1234, 2023.
35. Zhu, Y., Zhou, M., & Yang, R. "High-capacity image hiding algorithm based on chaotic encryption and optimized embedding." *Signal Processing: Image Communication, Elsevier*, vol. 118, pp. 102875, 2024.
36. Li, X., Zhou, Y., & Chen, G. "A color image encryption algorithm based on hyperchaotic system and DNA coding." *IEEE Transactions on Multimedia*, vol. 26, pp. 3375-3386, 2024.
37. Wang, L., Liu, J., & Li, Z. "Chaotic-based encryption scheme for secure image transmission over IoT networks." *Multimedia Tools and Applications, Springer*, vol. 83, pp. 1578-1601, 2024.
38. Chen, M., & Yang, H. "Efficient image encryption using logistic-sine system and cellular automata." *Signal Processing: Image Communication, Elsevier*, vol. 118, pp. 102889, 2024.
39. Zhu, Q., & Zhang, P. "Secure medical image transmission using hybrid chaotic maps and DNA encoding." *IEEE Access*, vol. 11, pp. 18245-18258, 2023.

40. Sun, W., & Zhao, X. "Enhanced video encryption technique based on 3D chaotic maps for real-time applications." *Journal of Real-Time Image Processing*, *Springer*, vol. 20, pp. 145-158, 2023.
41. Li, Y., Zhang, X., & Wang, H. "Robust image encryption algorithm against noise and geometrical attacks using chaotic maps." *IEEE Transactions on Multimedia*, vol. 26, pp. 3355-3367, 2024.
42. Chen, W., & Liu, Y. "Efficient video encryption method using chaotic systems for secure multimedia streaming." *Signal Processing: Image Communication*, *Elsevier*, vol. 119, pp. 102900, 2024.
43. Sun, R., & Zhao, J. "A secure image transmission scheme based on hybrid chaotic maps and wavelet transform." *Multimedia Tools and Applications*, *Springer*, vol. 83, pp. 12267-12285, 2023.
44. Wang, T., & Xu, Z. "Robustness analysis of chaotic encryption techniques for real-time IoT applications." *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1823-1835, 2023.
45. Zhang, Q., & Huang, M. "Advanced encryption algorithm based on fractional chaotic maps for secure image communication." *Journal of Real-Time Image Processing*, *Springer*, vol. 20, pp. 145-162, 2023.
46. Khalid M. Hosny, Mohamed A. Zaki, Nabil A. Lashin, Hanaa M. Hamza, Fast colored video encryption using block scrambling and multi-key generation, Springer The Visual Computer (2023) 39:6041–6072 https://doi.org/10.1007/s00371-022-02711-y
47. Heping Wen, Yiting Lin, Zhiyu Xie & Tengyu Liu, Chaos-based block permutation and dynamic sequence multiplexing for video encryption, www.nature.com/scientificreports , Scientific Reports | (2023) 13:14721 | https://doi.org/10.1038/s41598-023-41082-9
48. Deepti Dhingra, Mohit Dua, A chaos-based novel approach to video encryption using dynamic S-box, Multimedia Tools and Applications (2024) 83:1693–1723 https://doi.org/10.1007/s11042-023-15593-6
49. Yang Yang, Ming Cheng, Yingqiu Ding, and Weiming Zhang, A Visually Meaningful Image Encryption Scheme Based on Lossless Compression SPIHT Coding, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 16, NO. 4, JULY/AUGUST 2023, Digital Object Identifier 10.1109/TSC.2023.3258144
50. BHARTI AHUJA, RAJESH DORIYA, SHARAD SALUNKE, MOHAMMAD FARUKH HASHMI, (Senior Member, IEEE), AND ADITYA GUPTA, IoT-Based Multi-Dimensional Chaos Mapping System for Secure and Fast Transmission of Visual Data in Smart Cities, Digital Object Identifier 10.1109/ACCESS.2023.3318014
51. MOHAMED GABR, (Member, IEEE), RIMON ELIAS, (Senior Member, IEEE), KHALID M. HOSNY, (Senior Member, IEEE), GEORGE A. PAPAKOSTAS, AND WASSIM ALEXAN, (Senior Member, IEEE), Image Encryption via Base-n PRNGs and Parallel Base-n S-Boxes, Digital Object Identifier 10.1109/ACCESS.2023.3301460
52. Wang Jin, Liu Jiandong, Xu Haoqiang, H.264/AVC video encryption algorithm based on integer dynamic cross-coupling tent mapping mode, Multimedia Tools and Applications (2024) 83:13369–13393 https://doi.org/10.1007/s11042-023-15448-0
53. Dong Jiang, Tao Chen, Zhen Yuan, Wen-xin Li, Hai-tao Wang, Liang-liang Lu, Real-time chaotic video encryption based on multi-threaded parallel confusion and diffusion, Information Sciences 666 (2024) 120420, https://doi.org/10.1016/j.ins.2024.120420
54. Suo Gao , Herbert Ho-Ching Iu , Mengjiao Wang , Donghua Jiang , Ahmed A. Abd El-Latif , Rui Wu and Xianglong Tang, Design, Hardware Implementation, and Application in Video Encryption of the 2-D Memristive Cubic Map, IEEE INTERNET OF THINGS JOURNAL, VOL. 11, NO. 12, 15 JUNE 2024, Digital Object Identifier 10.1109/JIOT.2024.3376572
55. Monu Singh , Naman Baranwal , Kedar Nath Singh , and Amit Kumar Singh, Using GAN-Based Encryption to Secure Digital Images With Reconstruction Through Customized Super Resolution Network, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, VOL. 70, NO. 1, FEBRUARY 2024, Digital Object Identifier 10.1109/TCE.2023.3285626
56. TANVEER QAYYUM, TARIQ SHAH, ALI YAHYA HUMMDI, AMER ALJAEDI, AND ZAID BASSFAR, An Innovative Feasible Approach for Multi-Media Security Using Both Chaotic and Elliptic Curve Structures, Digital Object Identifier 10.1109/ACCESS.2024.3354170
57. Himanshu Kumar Singh and Amit Kumar Singh, Using Deep Learning to Embed Dual Marks With Encryption Through 3-D Chaotic Map, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, VOL. 70, NO. 1, FEBRUARY 2024, Digital Object Identifier 10.1109/TCE.2023.3286487

58. Sweta Kumari, Mohit Dua, Shelza Dua and Deepti Dhingra, A novel Cosine-Cosine chaotic map-based video encryption scheme, Journal of Engineering and Applied Science (2024) 71:36 https://doi.org/10.1186/s44147-024-00376-z
59. Parul Saini, Krishan Berwal, Shamal Kashid, Alok Negi, STKVS: secure technique for keyframes-based video summarizationmodel, Multimedia Tools and Applications https://doi.org/10.1007/s11042-024-18909-2
60. Deepti Dhingra, Mohit Dua, Medical video encryption using novel 2D Cosine-Sine map and dynamic DNA coding, Medical & Biological Engineering & Computing (2024) 62:237–255 https://doi.org/10.1007/s11517-023-02925-9
61. N. Ramesh Babu, P. Balasubramaniam, Er. Meng Joo, Video encryption via synchronization of a fractional order T-S fuzzy memristive hyperchaotic system, Multimedia Tools and Applications (2024) 83:26055–26088 https://doi.org/10.1007/s11042-023-16483-7
62. R. Roselinkiruba, T. Sree Sharmila, J. K. Josephine Julina, An efficient Moving object, Encryption, Compression and Interpolation technique for video steganography, Multimedia Tools and Applications (2024) 83:62741–62771, https://doi.org/10.1007/s11042-023-17930-1
63. Deyang Wu, Xinpeng Zhang, Jiayan Wang, Li Li, Guorui Feng, Novel robust video watermarking scheme based on concentric ring subband and visual cryptography with piecewise linear chaotic mapping, JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015, DOI 10.1109/TCSVT.2024.3405558
64. Liu X, Xiao D, Xiang Y (2019) Quantum image encryption using intra and inter bit permutation based on logistic map. IEEE Access 7:6937–6946. https://doi.org/10.1109/ACCESS.2018.2889896
65. Zhou N, Chen W, Yan X, Wang Y (Apr. 2018) Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system. *Quantum Inf Process* 17(6):137. https://doi.org/10.1007/s11128-018-1902-1
66. Sethi, J., Bhaumik, J., Chowdhury, A.S.: Chaos-Based Uncompressed Frame Level Video Encryption. Springer (2022). Accessed 15 Jun 2022 [Online]. https://doi.org/10.1007/978-981-16-6890- 6_15
67. Elkamchouchi, H., Salama, W.M., Abouelseoud, Y.: New video encryption schemes based on chaotic maps. Wiley Online Libr. 14(2), 397–406 (2019). https://doi.org/10.1049/iet-ipr.2018.5250
68. Kotel, S., Zeghid, M., Baganne, A., Saidani, T., Daradkeh, Y.I., Rached, T.: Fpga-based real-time implementation of aes algorithm for video encryption. Recent Adv. Telecommun Informatics Edu Technol 27–36 (2014)
69. Cheng, S.,Wang, L.,Ao, N.,Han,Q.:ASelectiveVideo Encryption Scheme Based on Coding Characteristics. Symmetry 12(3), 332 (2020). https://doi.org/10.3390/SYM12030332
70. Hafsa, A., Fradi, M., Sghaier, A., Malek, J., Machhout, M.: Real-time video security system using chaos- improved advanced encryption standard (IAES). Multimed. ToolsAppl. (2021). https:// doi.org/10.1007/S11042-021-11668-4/TABLES/14