IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 58, No. 9, September, 2025, pp. 1899-1909
Journal homepage: http://iieta.org/journals/jesa

Selective Chaotic Video Encryption for Versatile Video Compression H.266/ VVC Standard

Athraa H. Hilal*, Maher K. Mahmood Al-Azawi

Department of Electrical Engineering, College of Engineering, Al-Mustansiriyah University, Baghdad 10047, Iraq

Corresponding Author Email: athraa148@uomustansiriyah.edu.iq

https://doi.org/10.18280/jesa.580912

Received: 20 June 2025
Revised: 28 July 2025
Accepted: 20 August 2025
Available online: 30 September 2025

Keywords:
Versatile Video Coding H.266/VVC, selective encryption, 2D-LSCM, video compression, chaotic systems, real-time security

ABSTRACT

The increased need to transfer High-Definition (HD) video quickly and securely led to an urgent necessity to protect these types of data. The Versatile Video Coding (H.266/VVC) standard provides significant improvements in compression efficiency, but full encryption of the compressed video is often computationally expensive and unsuitable for real-time applications. In this research, a hybrid method that combines video compression using (H.266/VVC) with Selective Encryption (SE) based on a highly secure two-dimensional chaotic map is proposed. The method relies on encrypting sensitive components of the bit stream, such as motion vectors and key frames (I-frames), ensuring an acceptable level of security without affecting compression efficiency. Simulation results showed that the proposed method maintains a high compression ratio and significantly increases visual distortion against intruders, in addition to excelling the quality metrics such as Structural Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR) and entropy. PSNR showed a decrease of 35% for the compressed and encoded video compared to the compressed video, and SSIM decreased by 59% for the compressed and encoded video compared to the compressed video. Simulation results also show a low computation time, which may enable a possible real-time video application. This confirms the effectiveness of combining compression and chaotic selective encryption to achieve secure and efficient visual communication.

1. INTRODUCTION

In the current era, digital means, including international conferences over the Internet, distance education, and online gaming, have become widespread. However, this internet proliferation faces security challenges.

Encryption is the major method of ensuring data confidentiality and content integrity during data transmission or storage, particularly in open environments or untrusted networks, such as the Internet or wireless communications. Security of multimedia data is of significant concern in the majority of applications to enable secure storage and transmission of images and videos. The most straightforward solution to perform secure transmission of a video is to encrypt the whole video bit stream using a secure encryption algorithm such as Advanced Encryption Standard (AES) [1]. Video compression is a process used to reduce data size by removing redundant data in a smart and efficient manner. This reduces storage consumption and facilitates transportation across networks with minimal impact on video quality.

The H.266/Versatile Video Coding (VVC) is the latest generation of video compression algorithms, with the objective of offering a significant reduction in data size by up to 50% over the High Efficiency Video Coding H.265/HEVC at the same video quality. The VVC has sophisticated techniques that make this method suitable for applications requiring high quality and high compression efficiency, including 4K/8K virtual reality, cloud gaming, and live streaming [2].

Video content accounts for 82% of internet data, requiring large storage, processing time, and high bandwidth consumption. Traditional full encryption methods are insufficient due to data size and the slow full encryption of compressed video. Selective encryption of compressed video is considered the best solution to this problem; however, it faces gaps in both security and compression efficiency.

In recent years, the confidentiality of video data has become a challenging research topic. In 2013, Van Wallendael et al. [3] proposed encrypting specific elements of HEVC video using AES. Encryption showed a significant impact on quality when applied to high-motion videos. Some elements did not affect the compression rate, while others increased it. Ouamri et al. [4] discussed SE of HEVC videos by using AES-CBC as the encryption algorithm before the Binary Arithmetic Coding (BAC) stage. Encrypting only the suffixes without affecting the rest of the data may leave a fixed pattern that can be exploited to extract unencrypted information. Abu Taha et al. [5] introduced SE to protect Regions of Interest (ROI) within the video, using a chaotic system in the HEVC standard. The difference elements in Motion Vectors Difference (MVD), Motion Vector signs (MV Signs), Transformation Coefficients (TC), Transaction signs (TC Signs), and Intra Prediction Modes (IPM) are all encoded. Chen et al. [6] proposed SE That relies on encrypting each slice using the RC4 algorithm. However, encrypting elements such as IPMs will increase the bit rate required for video transmission, which may affect

compression efficiency. Farajallah et al. [7] proposed using VVC with SE. The encryption was done by a simple XOR operation between a binary sequence as a key and some selected parameters. This makes this system vulnerable to small attacks. Yu and Kim [8] proposed a method using the HEVC standard and ROI to encode motion vectors (MV), transform coefficients (TC), and intra prediction mode (IPM). At the same time, the visual distortion was more efficient and accurate than traditional encryption, but still, the use of HEVC may cause less compression efficiency as compared to VVC. Chen et al. [9] discussed a robust encryption method for compressed video using the VVC standard to secure video content. They used a 3-Dimensional Dynamical Chaos Model (3D-DCM) for encryption to perform encryption at the slice level, where each slice is encrypted independently. Encrypted information includes texture-related elements such as QTC transformation coefficients, MVD motion vectors, and IPM intra prediction.

Ibraheem et al. [10] proposed an algorithm based on the genetic algorithm, where they encoded Intra Prediction Modes, Transform Units (TUs), Quantization Parameters (QPs), and Entropy Coding Modes. The goal is to reduce and optimize encoding without affecting video quality. Shah et al. [11] proposed a new encryption algorithm, pixJS, based on chaotic systems. The algorithm is built on three main components: jumping process, logistic map, and linear feedback shift register. The algorithm exhibited high entropy and resistance to differential attacks. Usmani et al. [12] proposed a solution to improve the efficiency of 360-degree interactive video streaming. They used SE as a trade-off between security and processing speed. Although security was improved, this was at the expense of increased computation time.

In this paper, an SE is used with H.266/VVC compression standard, utilizing a two-dimensional sine cosine logistic map. The algorithm, based on chaotic systems, ensures high confidentiality and security while maintaining compression efficiency and video quality. The algorithm's performance was evaluated through security analysis, quality assessment, execution time, and metrics. The suggested technique combines the powerful VVC video compression and SE based on a highly efficient two-dimensional sine-cosine logistic map (2D-SCLM). The proposed chaotic SE is done directly inside the VVC bitstream, ensuring complete compatibility with the standard format and bit rate stability. The sensitive parameters in the video bitstream, such as I-frames and motion vector signs, are used, resulting in robust SE without impacting the decryption process. Unlike previous approaches that employ previous compression standards like AVC or HEVC, the proposed method uses VVC as a compression algorithm. Additionally, the use of such a chaotic map to encrypt part of the video will give increased security, enhanced key sensitivity, and greater resistance to statistical and differential attacks.

The rest of the paper is organized as follows: Section II provides an overview of the VVC standard structure and SE algorithm; Section III explains the chaotic map; Section IV shows the proposed algorithm. Section V gives the simulation results of the proposed algorithm. Some concluding remarks are provided in Section VI.

2. VERSATILE VIDEO CODING STANDARD (VVC)

The VVC is the latest standard from the ITU-T and ISO/IEC Standardization Organizations. This code was created by the Joint Video Expert Team (JVET), and VVC was completed in July 2020. The new VVC standard aims to substantially enhance compression efficiency relative to earlier techniques. The VVC architecture encompasses a wide array of video applications beyond traditional formats. The VVC has a high dynamic range and an extensive color gamut, accommodates computer-generated video and omnidirectional video, and enables adaptive streaming with resolution switching, scalable coding, and tile-based streaming for immersive applications [13].

Some of the uses of VVC are in Digital Video Broadcasting (DVB), Fifth generation networks (5G – 3GPP), and Smart television systems. The standard is designed to be flexible and serve current and future applications. It has been supported by several organizations to facilitate its adoption and spread in the market [14].

Figure 1 shows the block diagram of VVC. As compared to H.265/HEVC, which provides 35 intra prediction modes, H.266/VVC provides planar and 65 angle modes. This is one of the most significant changes that can have improved predictions. Although this improves encoding quality, it also enhances computational time and programming complexity. The H.266/VVC introduces many advanced coding tools, which significantly improve the coding effectiveness of the future video coding standard. Figure 2 shows the Intra prediction modes in VVC [15].

VVC, along with Advanced Video Coding AVC and the HEVC, has a DC intra prediction mode that generates predictions from the average of left and top reference samples. This mode is used for square blocks with an integral power of 2, but not non-square blocks. VVC uses reference samples along the longer edge of a rectangular block to calculate the mean value. It also implements the Planar intra-prediction mode of HEVC, which calculates predicted sample values as a weighted average of four reference samples [15].

Inter prediction is followed by intra prediction. Inter-coding relies on inter-prediction of texture and motion data from previously reconstructed images in the Decoded Picture Buffer (DPB). VVC utilizes inter-picture prediction motion, a technique first seen in HEVC, but incorporates key extensions that lead to improved coding performance, as shown in Figure 3. The motion from the position of the block in the motion vector (Δx, Δy) is represented by the position of the block in the current image, where (Δx) and (Δy) represent the horizontal and vertical motions compared to the previous reference image. The reference time index Δt refers to the previously coded pictures, also known as reference photographs. Motion data is the result of combining the motion vectors and the reference index. There are two forms of interpretation: uni-prediction and bi-prediction. Bi-prediction generates the final motion-compensated prediction by combining two pairs of motion data ($\Delta x0$, $\Delta y0$, $\Delta t0$) and ($\Delta x1$, $\Delta y1$, $\Delta t1$) [16].

The motion correction makes use of separate linear filters with 8 taps and 16 phases for luma and 4 taps and 32 phases for chroma. VVC further facilitates the adaptive modification of coded image resolution with the Reference Image Resampling (RPR) tool. Four distinct sets of interpolation filters are used based on the motion model, block size, and scaling ratio between the reference image and the current image. Image (maximum down-sampling ratio of 2 and up-sampling ratio of 8). RPR further provides innovative functionalities for bit-rate regulation to accommodate

fluctuations in network capacity, and it may be expanded for scalability and adaptive resolution encoding. VVC has various new settings for motion prediction blending compared to HEVC [14].

Post-compression SE is usually preferred where the video is first compressed, then selectively encrypted, as shown in Figure 4. This kind of algorithm is generally compression-friendly; minor overhead can be introduced to send the encryption key or encryption information. Encryption and decryption don't need to change at the encoder and decoder ends. Post-compression encoding maintains compression efficiency by applying encoding after compression, preserving video quality with the same file size as the compressed file. This reduces computational complexity, making it ideal for live streaming and real-time applications. It doesn't increase the encoded video size due to randomness, and can have varying levels of access [17].

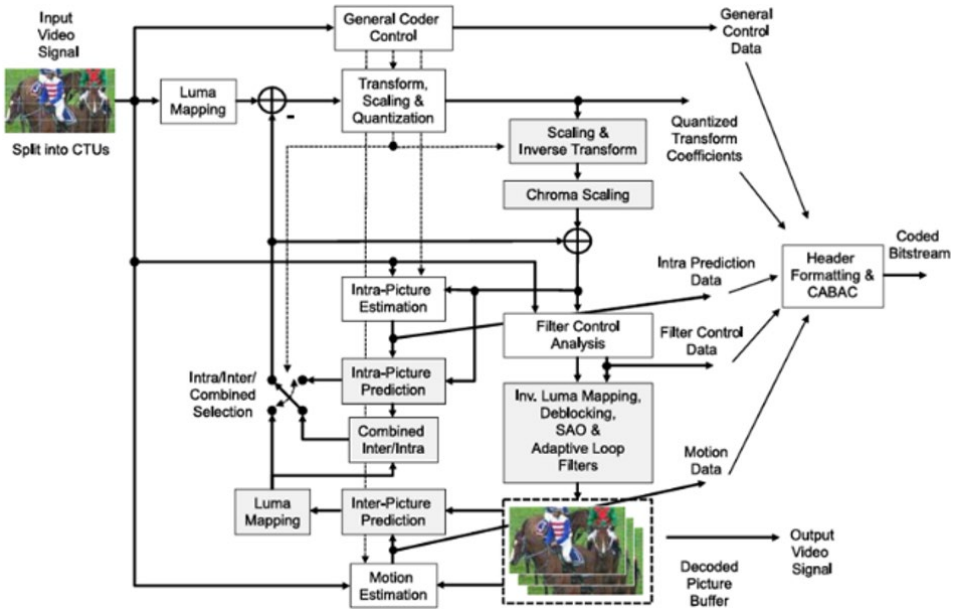


Figure 1. H.266 / VVC block diagram [18]

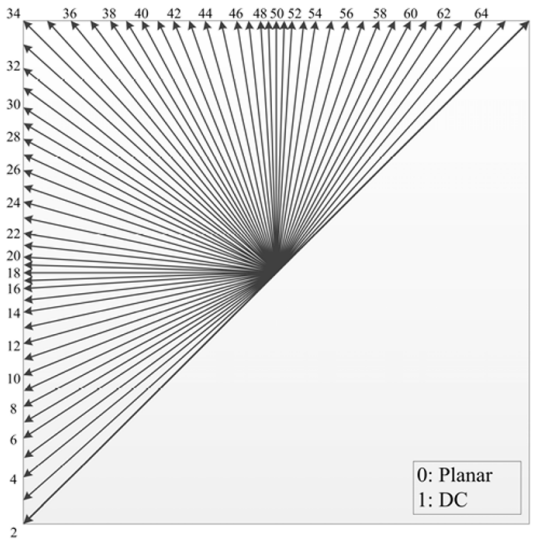


Figure 2. Intra prediction modes in VVC [19]

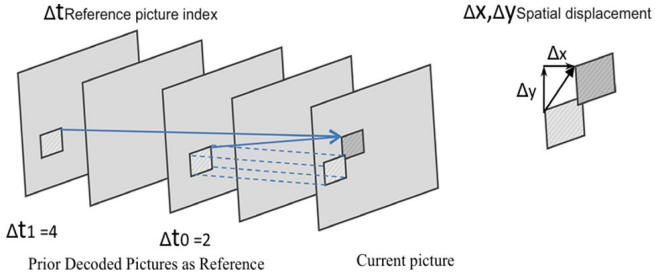


Figure 3. Inter-picture prediction concept and parameters [16]

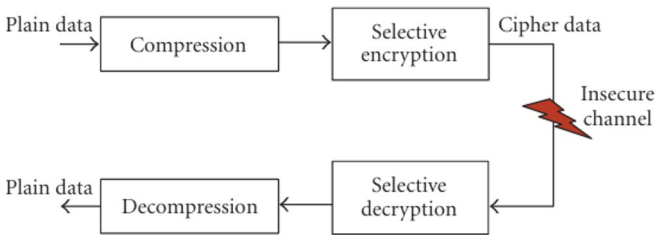


Figure 4. Post-compression approach [17]

The proposed approach secures compressed video content by encrypting only the most susceptible portions of the H.266/VVC compression standard's bit stream. This SE detects important visual or structural information, extracts these parts, and applies chaotic encryption with keys from a 2D Sine-Cosine-Logistic Map. This lightweight encryption is suitable for real-time use and ensures optimal efficiency.

3. TWO-DIMENSIONAL CHAOTIC SINE COSINE LOGISTIC MAP (2D-SCLM)

The proposed cryptographic key generation method uses a Two-Dimensional Sine Cosine Map (2D-SCLM). This combines sine, cosine, and logistic maps for (SE). This method has high randomness, high input sensitivity, and difficulty in making predictions about system behavior due to its non-linear nature and dependency on multiple dimensions.

The map possesses the feature of high potential for generating complex chaotic sequences and is therefore very suitable for generating random encryption keys in information

security, especially selective video encryption.

The 2D-SCLM was chosen for its durability and also has many advantages, like high sensitivity to initial conditions and parameters, increased complexity and security through 2D coupling, uniform and pseudo-random distribution, efficient implementation, suitability for SE, improved statistical properties, and compatibility with video compression standards.

The map is defined by the following Eqs. (1) and (2) [20].

$$x_{n+1} = \sin\left(\pi x_n\right) + a\left(sin\left(\pi x_n\right) - sin^2\left(\pi x_n\right) + \sin\left(\pi y_n\right)\right) \quad (1)$$

$$y_{n+1} = cos(\pi y_n) + b(cos(\pi y_n) - cos^2(\pi y_n) + cos\ (\pi x_n)) \quad (2)$$

The control parameters a &b (0, +∞), This increases the key size. The study used a Lyapunov exponent analysis to determine the ranges of a and b that lead to strong chaotic behavior. Results showed strong sensitivity to initial conditions at 1.7 and 0.67, and a deep chaotic regime at 0.67. Standard encryption metrics confirmed these parameters' excellent confusion and diffusion properties.

The 2D-SCLM has a wider chaotic range, larger Lyapunov exponent values, and complicated and non-linear chaotic dynamics. These make the generated keys highly random, highly sensitive to minimal changes in the initial keys (i.e., a trivial change resulted in completely different outputs). Complexity in the prediction of the system behavior, which strengthens the algorithm to withstand brute force and statistical attacks. Therefore, 2D-LSCM is employed as a chaotic key generator to selectively encrypt sensitive portions of the bit stream in the H.266/VVC compression standard, thereby achieving both security and efficiency. Figure 5 shows the attractors of 2D-SCLM.

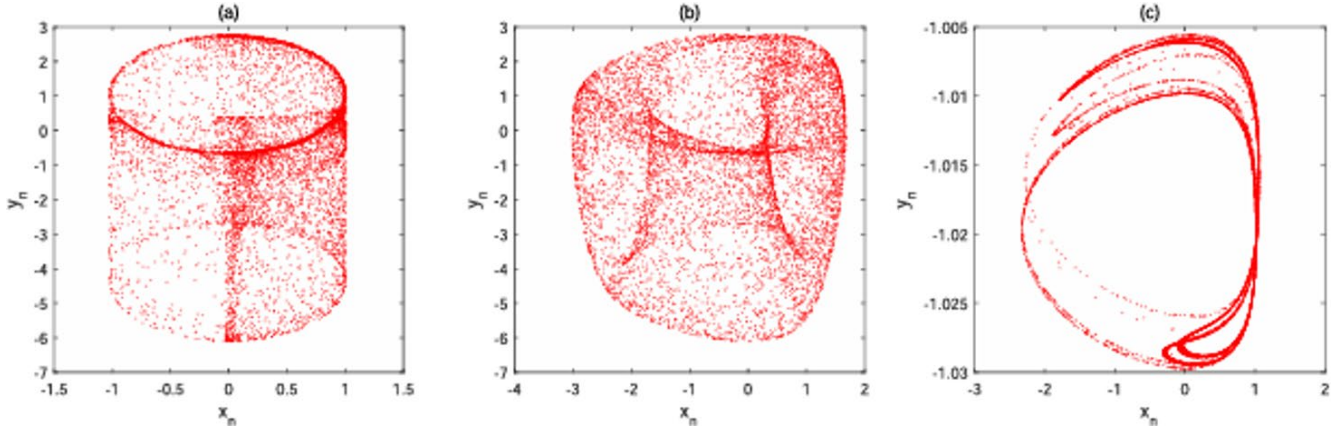


Figure 5. The attractors of the 2D-SCLM are as follows: (a) the attractor described by parameters a = 0.01 and b = 1.7; (b) the attractor with parameters a = 0.67 and b = 1.7; and (c) the attractor with parameters a = 0.67 and b = 0.01 [20]

4. PROPOSED SE ALGORITHM PROCESS

This research proposes a system for encoding compressed video using the VVC technique, based on chaos techniques (2D-SCLM), with the aim of enhancing the security of video transmission over digital networks while maintaining performance efficiency and reducing execution time. The proposed system focuses on encoding the motion vectors of I-frames resulting from video compression, considering them as a sensitive part that can reveal video content even when other frames are encrypted.

The system relies on using chaotic maps to generate random and unpredictable digital sequences used in the process of encrypting motion vectors, making it more difficult to analyze or retrieve the original content without knowledge of the encryption keys. This type of encryption was chosen due to its important characteristics, such as high sensitivity to inputs, simplicity in implementation, and speed of performance compared to traditional methods.

Components of the proposed system:

•Compressed video analysis: Motion vectors and I-frames from the compressed video are determined using the H.266/VVC algorithm, where motion is a key factor for movement between frames and is important for efficiently compressing the video. However, its encoding significantly deteriorates the visual quality of the video.

•Generating the chaotic sequence: A strong and long-range chaotic map was employed to generate the key using 2D-SCLM, consisting of 16 digits, which are used as keys for video encryption.

•The blocks containing motion are identified, then the horizontal and vertical displacements of the motion vector between the current frame and the previous frame are determined to find the fluidity of the motion. The two displacements are then multiplied by the key and added together to generate more randomness. The sum is then added to the actual pixel value in the frame within the video, and the result is taken modulo 256 to avoid overflow.

$$Enc = E(x,y) = ((v_x * key + v_y * key) + F(x,y))mod256 \quad (3)$$

where, $v_x\ and\ v_y$ Regarding the horizontal and vertical motion vector displacements, E (x, y) is the pixel value after encoding at the coordinates (x, y), F (x, y) is the original pixel value at the coordinates (x, y), and key represents the secret key that was generated from the chaotic map.

$$Dec = F(x,y) = E(x,y) - ((v_x * key + v_y * key))mod256 \quad (4)$$

F (x, y): The pixel value after decoding at the coordinates (x, y) few studies address selective encryption for VVC, and

most focus on HEVC Some studies report results on private video datasets, which prevents fair and reproducible comparisons therefore, we provided comparisons with HEVC-based methods as a baseline, which represent the most relevant reference available using publicly accessible datasets.

Eight video sequences were used to evaluate the proposed scheme. Figure 6 shows these eight video clips. These videos have different resolutions, movements, colors, CTU, and frame rates. The experiments were conducted in some environment functions like Visual Studio Code version 1.97.2 and Intel(R) Core (TM) i5-10210U CPU @ 1.60 GHz RAM on Windows 10 OS. The reference software used is the VVencapp library and the VVdecapp library for compressed. where vvencapp version 1.13.1, and VVdecapp version 3.0.0 were used [21].

Figure 7 shows the flow chart of the proposed (SE) process. Below is the proposed chaotic encryption and evaluation pipeline.

Proposed Chaotic Encryption and Evaluation Pipeline
Input: *video_file, QP_value*
Output: *encrypted video, entropy value, PSNR, SSIM, bitrate, encryption time*
CALL compress_video(video_file, QP_value) → output.vvc
CALL decode_vvc(output.vvc) → decoded.y4m
CALL convert_to_mp4(decoded.y4m) → output.mp4
key ← generate_chaotic_key (a, b, x0, y0, iterations)
FOR *each frame IN video:*
flow ← optical_flow (previous_frame, current_frame)
encrypted ← encrypt (frame, flow, key)
decrypted ← decrypt (encrypted, flow, key)
END FOR
entropy ← calculate_entropy(encrypted_video)
psnr, ssim ← calculate_psnr_ssim (original, encrypted)
bitrate ← calculate_bitrate(encrypted_video)
PRINT *entropy, psnr, ssim, bitrate*
END

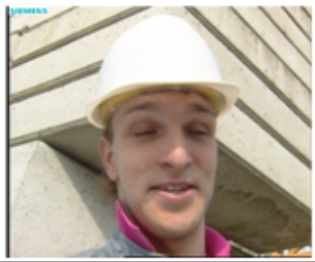
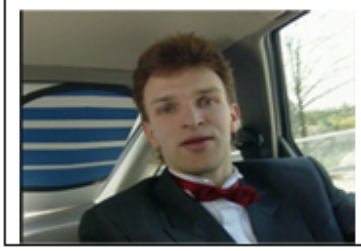
video sequence name	Resolution/ Number of tested frames	video sequence name	Resolution/ Number of tested frames
Akiyo 	Cif 352 x 288 300	Football 	Cif 352 x 288 260
bowing 	Cif 352 x 288 300	Foreman 	Cif 352 x 288 300
Bus 	Cif 352 x 288 150	Mobile 	Cif 352 x 288 300
Carephone 	Cif 352 x 288 260	News 	Cif 352 x 288 300

Figure 6. Video sequences

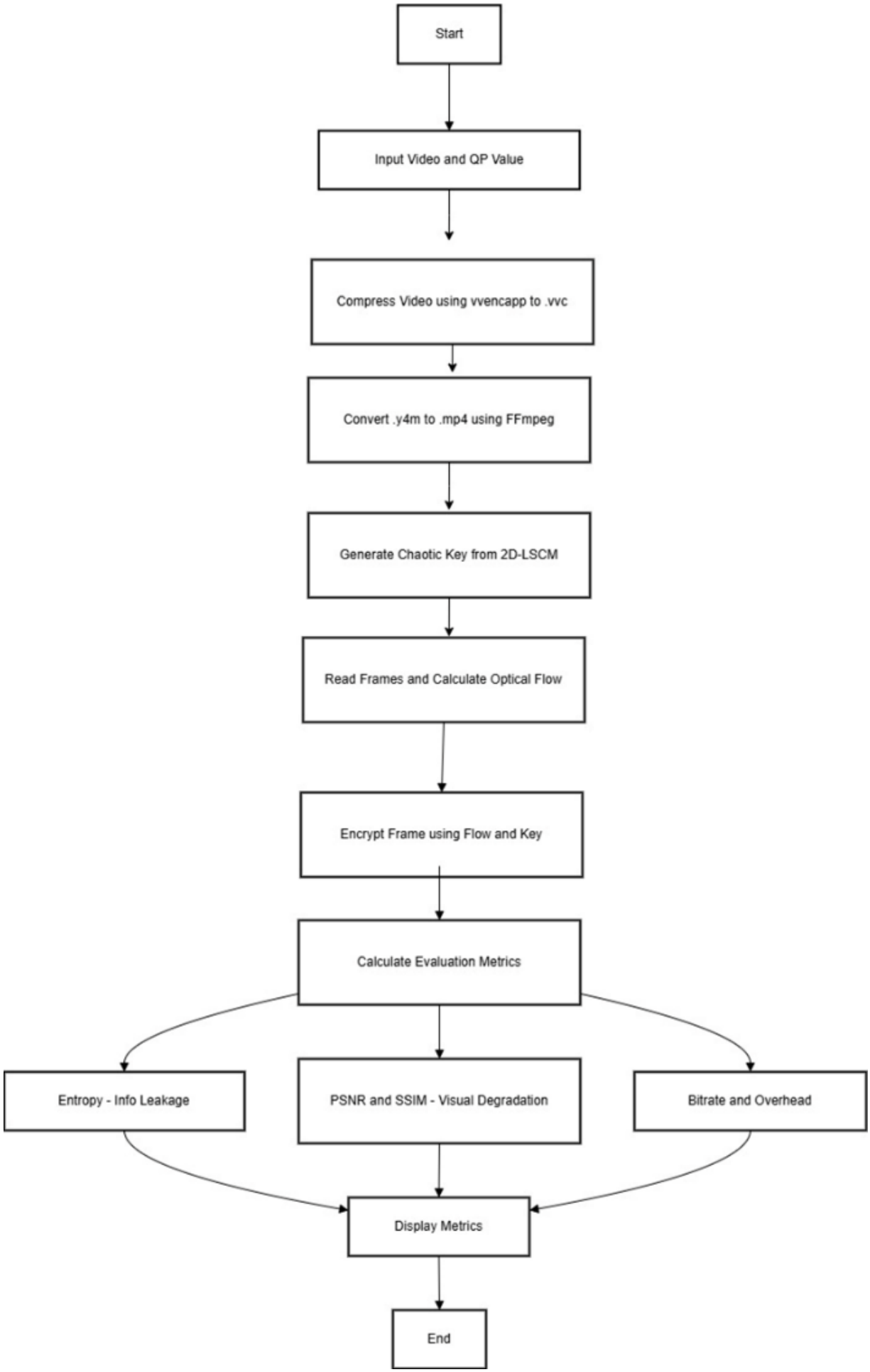


Figure 7. Flow chart of the proposed chaotic SE process

5. SIMULATION RESULTS

5.1 Video vision results

Here, the Quantization Parameter QP of the VVC encoder is set to 24, taking Akiyo, bus, and mobile videos as examples. The decoded results of some frames of the original video and the encrypted video are shown in Figure 7.

Here, the Quantization Parameter QP of the VVC encoder is set to 24, taking Akiyo, bus, and mobile videos as examples. The decoded results of some frames of the original video and the encrypted video are shown in Figure 8, where vision analysis shows low residual intelligibility, noise, and loss of main information of the frames.

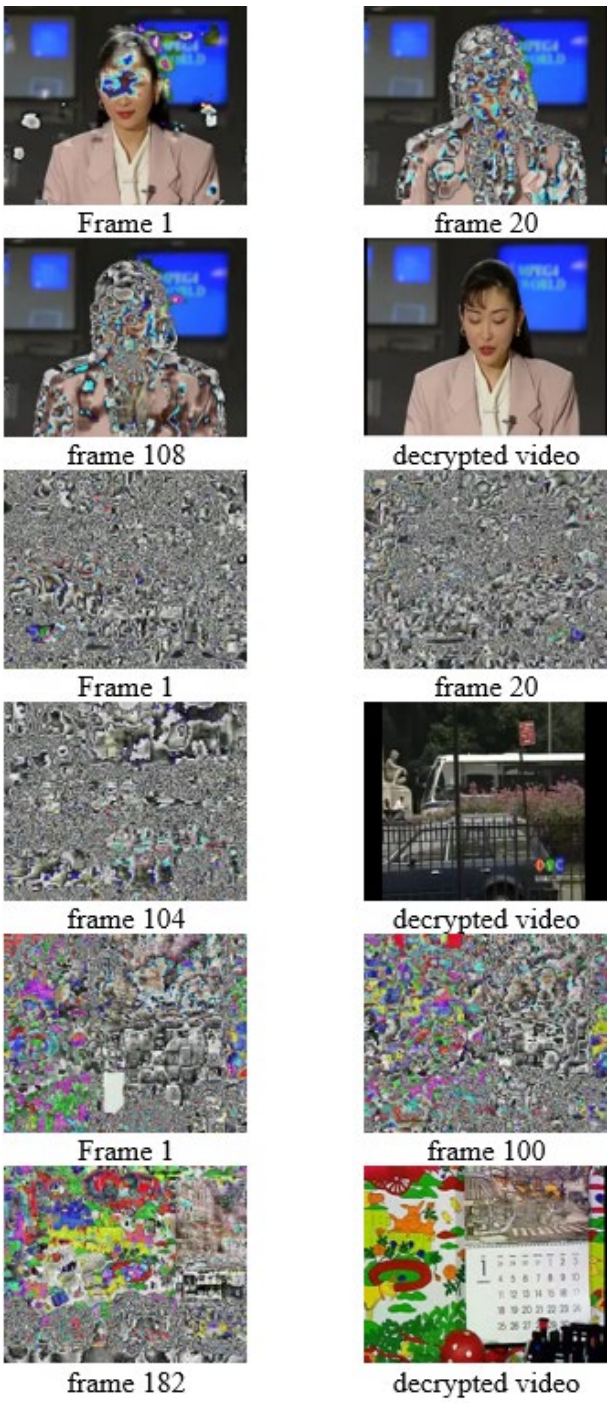


Figure 8. Encrypted and decrypted frames for Akiyo, bus and mobile videos

5.2 Objective indicator results

Peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are two objective indicators to further evaluate the quality of the encrypted frames. The PSNR is described as [7]

$$PSNR = 10\ log \frac{255\ \times\ 255}{\frac{1}{MH\ XMV}\ \sum_{i=1}^{MH}\sum_{j=1}^{mv}(X(i,j)-Y(i,j))^2} \quad (5)$$

where, MHxNV is the size of the video frame. X and Y represent the original frame and the encrypted frame, respectively. A smaller value of PSNR implies a higher level of encryption (less residual intelligibility). The SSIM is defined as [7]:

$$SSIM = \frac{(2\mu_X\ \mu_Y + (LK_1)^2)(2\sigma_{XY}\ + (LK_2)^2)}{(\mu_X^2 + \mu_Y^2 + (LK_1)^2)(\sigma_X^2 + \sigma_Y^2 + (LK_2)^2)} \quad (6)$$

where, μ_x, μ_y are the averages of X and Y $and\ \sigma_x^2,\ \sigma_y^2$ are variances of X and Y. L = 255, k_1 = 0.1, k_2 = 0.03, and SSIM $\in$(0,1). The smallest values for SSIM refer to less residual intelligibility.

Table 1 shows PSNR and SSIM values for the eight selectively encrypted videos. This shows a significant decrease in both PSNR and SSIM values following the SE of video sequences. The PSNR metric is decreased due to distortion in the output image. The SSIM norm, quantifying structural similarity between original images and encrypted ones, also shows a drastic loss. The effect is weak for more stationary (less motion) videos like "akiyo" and "news". The low-motion videos like "akiyo" maintain quality better than high-motion videos. The results suggest a correlation between numerical loss of PSNR and structural similarity loss (SSIM), indicating that the degree of motion in the video type directly affects video quality after encoding. Of course, this is not related to SE, but it is related to the compression process.

In this part of the study, the performance of the proposed system based on the VVC (H.266) standard, with other algorithms based on HEVC by Jovanović [21] and RSVE [6] is compared. This is due to the unavailability of papers that have tested VVC and contain results. The comparison includes measuring video quality using indicators such as PSNR and SSIM under various conditions, including different QP values. Tables 2 and 3 show the comparison between Jovanović and the RSVE HEVC SE technique and the proposed algorithm for PSNR and SSIM, respectively.

5.3 Bit rate overhead results

The bitrate line shows the short-term impact of the proposed encryption algorithm on the size of the resulting information. The bit rate overhead is defined as in Eq. (7).

Bitrate

$$Overhead = \left(\frac{Bitrate\ after\ encryption - Original\ Bitrate}{Original\ Bitrate}\right) \times 100 \quad (7)$$

Table 4 shows the results of the bitrate overhead for compressed video, encrypted compressed video and bitrate overhead.

Table 1. PSNR and SSIM values for the encrypted video

Video Sequence Name	Original PSNR Compressed Without Encryption (dB)	PSNR Compressed and Encrypted Video (dB)	Original SSIM Compressed Without Encryption (dB)	SSIM Compressed and Encrypted Video (dB)
Akiyo	22.75352	13.62451	0.78943	0.53181
Bowing	26.73008	16.89302	0.85692	0.61220
Bus	10.62862	8.31778	0.10657	0.00905
Carphone	16.22445	8.72333	0.53010	0.03012
Football	12.70281	8.31778	0.17399	0.00905
Foreman	12.31483	8.58213	0.28806	0.01802
Mobile	8.90851	9.60240	0.06094	0.01392
News	18.54412	9.45860	0.69608	0.18505

Table 2. PSNR comparison of the proposed algorithm with Jovanović and RSVE HEVC (SE) techniques

Video Sequence	QP	Original (dB)	Jovanović [21] (dB)	RSVE [6] (dB)	Proposed (dB)
Mobile	12	51.2080	10.8199	9.9067	9.20227
	24	40.9997	11.3320	9.9786	9.19263
	36	32.9772	10.2106	9.5058	9.27812
foreman	12	51.5915	12.2185	11.1853	8.78254
	24	44.4308	13.8540	12.4377	8.78254
	36	38.4626	13.0729	12.7978	8.78256
Johnny	12	51.8393	14.0890	13.2773	12.25012
	24	46.6292	14.2316	12.9926	12.43497
	36	41.9026	14.3882	13.5148	12.61361
KristenAndSAR	12	51.8475	15.7376	14.0696	11.87881
	24	46.4661	15.7631	14.7831	11.97953
	36	41.4027	15.1599	13.9391	12.13750

Table 3. SSIM comparison of proposed algorithm with Jovanović and RSVE HEVC (SE) techniques

Video Sequence	QP	Original	Jovanović [21]	RSVE [6]	Proposed
Mobile	12	0.9984	0.0948	0.0592	0.01554
	24	0.9889	0.0957	0.0663	0.01563
	36	0.9313	0.1098	0.0738	0.01542
Foreman	12	0.9961	0.3556	0.2381	0.01740
	24	0.9712	0.3005	0.2279	0.01740
	36	0.9015	0.1450	0.3437	0.01740
Johnny	12	0.9944	0.4925	0.4359	0.55079
	24	0.9725	0.4706	0.3682	0.56636
	36	0.9428	0.5706	0.3905	0.57257
KristenAndSAR	12	0.9944	0.4474	0.4022	0.52317
	24	0.9765	0.4253	0.3939	0.53057
	36	0.9517	0.4779	0.4221	0.54064

Table 4. The bitrate overhead for the original and encrypted video

Video Sequences1	Original Bitrate	Bitrate Encrypted Video	Bitrate Overhead
Akiyo	88.70	88.77	0.000789
Bowing	129.06	133.07	0.031070
Bus	665.11	690.01	0.037437
Carphone	119.31	122.09	0.023300
Football	991.50	999.8	0.008371
Foreman	382.44	393.22	0.028187
Mobile	733.78	745.99	0.016639
News	199.11	212.23	0.065893

5.4 Encryption time results

The proposed encryption algorithm's performance in terms of processing time for videos was evaluated. Table 5 shows that the average encoding time per frame ranges from 0.00505 seconds for "Carphone" to 0.01884 seconds for "bus". Videos with low complexity or fewer frames have higher efficiency, while more detailed or motion-heavy ones consume more time per frame. The algorithm operates within a fair timescale for real-world applications.

5.5 Information entropy results

Entropy measures randomness and data distribution in the video, and it is an excellent measure of encryption strength. The entropy must be higher after encryption, which means that there is an improvement in data randomness, and thus, predicting is more difficult [7].

Table 5. Encryption time per frame for test videos

Video Sequences	Original Time Video (Sec)	Total Encrypted Time	Encryption Time per Frame
Akiyo	00.10	5.67877	0.01594
Bowing	00.10	4.83611	0.01537
Bus	00.05	2.83111	0.01884
Carphone	00.12	2.46477	0.00505
Football	00.08	4.47277	0.01803
Foreman	00.10	9.47928	0.01855
Mobile	00.10	10.98191	0.01709
News	00.10	5.37847	0.01761

$$H(m) = \sum_{i=1}^{2^{Lb}} P(m_i) log \frac{1}{P(m_i)} \quad (8)$$

where, Lb is the number of bits per pixel and $p(m_i)$ is the probability of pixel's level m_i.

Table 6. Entropy values for video sequences

Video Sequences	Original Entropy (bits)	Encrypted Video Entropy (bits)
Akiyo	7.23569	7.32102
Bowing	6.77865	7.38630
Bus	7.24721	7.88028
Carphone	7.30409	7.84210
Football	6.74452	7.88028
Foreman	7.36493	7.86216
Mobile	7.77445	7.85538
News	7.06541	7.40659

Table 6 shows that all video sequences gained an increase in their entropy value after being selectively encrypted, indicating improved randomness and enhanced system reliability. This increase in entropy enhances the algorithm's ability to conceal the original.

5.6 Analysis of key space

The key space specifies the number of available keys that may be utilised to create various key streams. A larger key space makes the system more resistant to brute-force (exhaustive) attacks. The suggested approach uses the chaotic system's initial values (x_0, y_0) as the secret key.

Assuming both (x_0, y_0) are represented as 16-digit double-precision floating-point integers inside the region [0,1] [0,1], the number of potential values for each variable is about 10^{16}. Therefore, the total key space may be estimated as follows, with 1000 iterations, as explained in Eq. (9).

$$s_k = Cardx_0 . Cardy_0 . Cardt = 10^{16} \times 10^{16} \times 10^3 = 10^{35} \approx 2^{116} \quad (9)$$

where Card {} · represents the cardinality of a set.

Since the key space is approximately 2^{116}, the system resists the Brute force attack.

5.7 Analysis of computation complexity

The encryption relied mainly on simple multiplication and addition operations on the encryption side, with multiplication and subtraction operations on the decryption side, which means the computational complexity is extremely low. Because of that, the proposed algorithm is high-speed and has minimal impact on the overall coding efficiency.

5.8. Key sensitivity analysis

A key sensitivity experiment was performed to evaluate the proposed chaotic encryption scheme's sensitivity to minor changes in the initial key settings. Two chaotic keys were produced with a 10-10 difference.

Key 1: initialized with x_0=0.1731234987

Key 2: initialized with x_0=0.1731234988

Key 1 is used to encrypt the bus video. The resulting encrypted video is decrypted with key 2. Figure 9 shows a frame of the encrypted video with key 1 and the decrypted video with key 2. Table 7 shows the key sensitivity result.

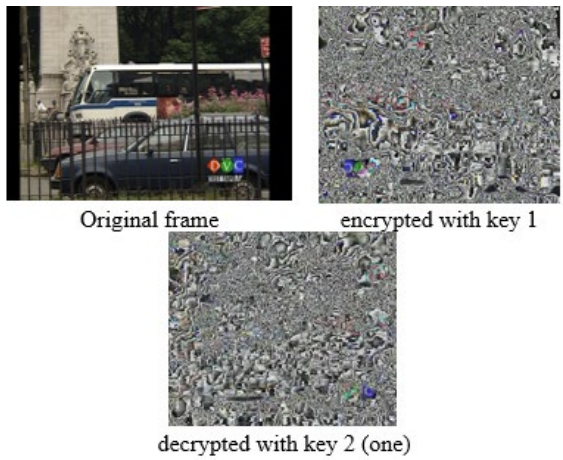


Figure 9. Key sensitivity result

Table 7. Key sensitivity results

	Original	Encrypted with Key 1	Decrypted with Key 2
PSNR	19.62860	8.31778	8.31652
SSIM	0.10657	0.00905	0.00910

This experiment highlights the significant key sensitivity of the chaotic encryption method. A slight modification in the initial condition produces significantly varied Decrypted outputs, an important feature against brute-force and differential assaults.

5.9 Objective indicator results for all frames

Figure 10 shows the objective indicator results (PSNR, SSIM, entropy of encrypted video, and encryption time per frame) for all frames of the Akiyo video. A sharp decline in PSNR and SSIM values is observed after the first frame, reflecting the impact of (SE) on image quality. Furthermore, the entropy remains consistently high (approximately 7.32), indicating strong randomness and effective encryption. Additionally, the encoding time stabilizes rapidly at a low value (around 0.01 to 0.02 seconds per frame), demonstrating the efficiency of the proposed method and its potential for real-time applications.

5.10 Statistical analysis

A test was conducted to confirm the effectiveness of the

proposed method against statistical analysis attacks by examining the histogram of one of the videos. The histogram compares the distribution of pixels between the compressed video represented by the blue line and the compressed and encoded video represented by the red line, for each channel of the YUV color space.

The histogram in Figure 11 shows the strong statistical effect of the algorithm, where the blue distribution representing the original unencrypted compressed video displays clear and irregular patterns, with pixels concentrated in specific areas. These characteristics constitute a weakness, allowing an intruder to infer information or statistically attack the system. As for the compressed and encrypted video, represented by the blue histogram, it appeared almost flat and uniform across all color channels, which completely proves the randomness of the data.

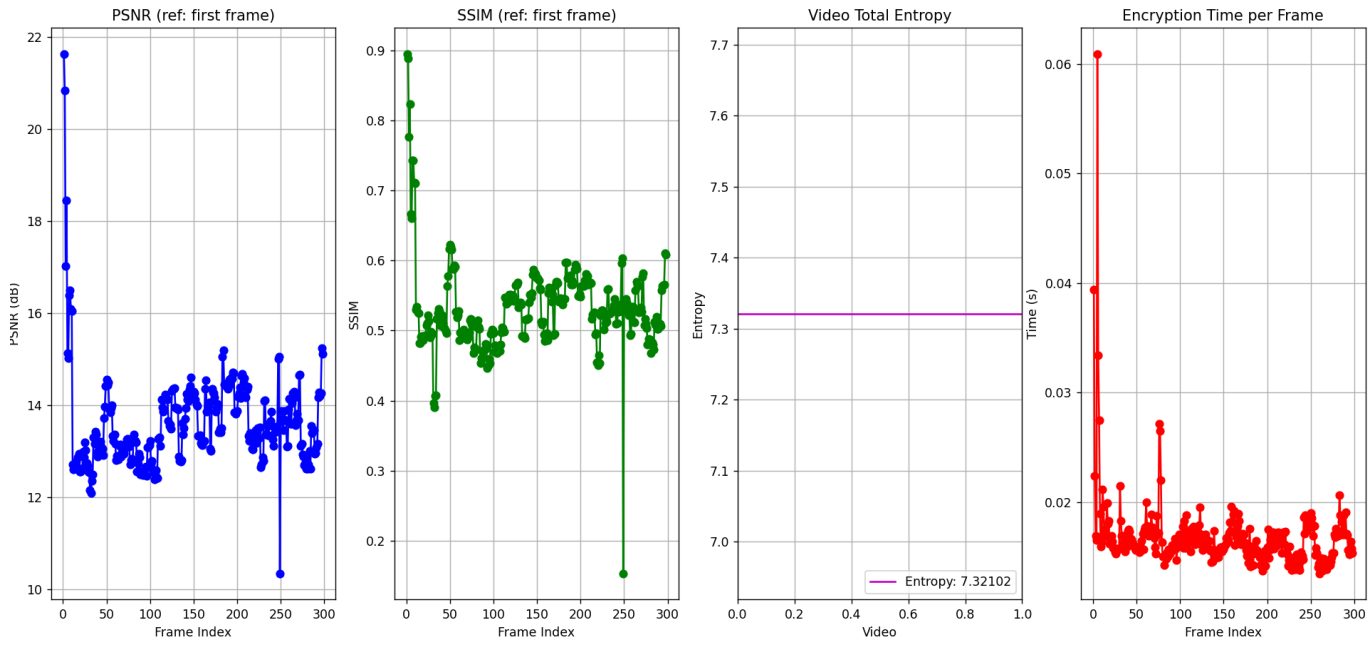


Figure 10. Objective indicator results for all frames of the Akiyo video

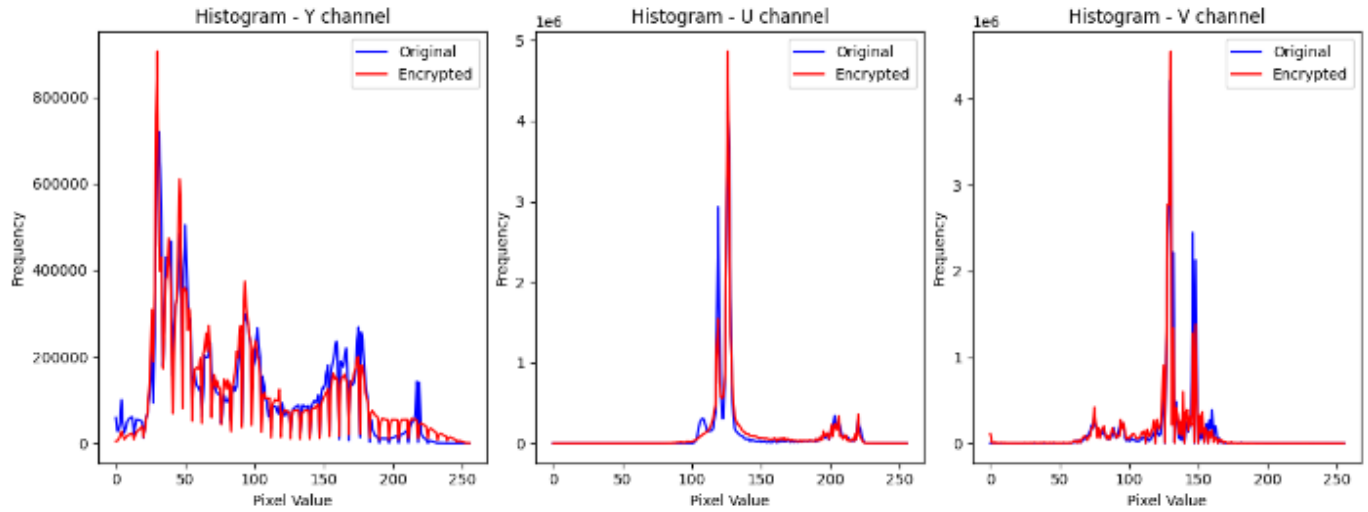


Figure 11. Histogram analysis

6. CONCLUSIONS

The proposed encryption algorithm can be used in Cloud Video Streaming, where video content can be protected during transmission and storage on cloud servers. Another application is Teleconferencing, where faces or certain ROIs can be encrypted without significantly affecting streaming speed. It can also be used in Video-on-Demand for television programs and movies, or for security and surveillance. As for the constraints and practical challenges, the encryption algorithm relies on video motion. Therefore, in high-definition videos or when the frame rate increases, the algorithm requires significant computational power, which will increase the encryption time. Additionally, encryption may also increase the bit rate.

A compressed video selective encryption approach using a Two-Dimensional Sine Cosine Logistic Map (2D-SCLM) is presented in this study. Its prime focus is to increase the security of the video and minimize encryption time without

actually affecting overall performance. Outcomes show that the suggested approach has an apparent visual quality loss, as reflected by decreased PSNR and SSIM metrics following the first frame. This is a sign of the encryption process being successful in concealing visually significant information. Conversely, the comparatively high values of entropy in all cases indicate enhanced randomness and resistance to predictability, which suggests maximum security. Apart from this, the encryption time for each frame was also observed to decrease to a consistently lower level, showing its feasibility in real-time systems. The proposed method is found to realize an acceptable balance among security, computation cost, and the visual degradation level needed.

REFERENCES

[1] Abdullah, A.M. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptography and Network Security, 16(1): 11. https://www.researchgate.net/publication/317615794.

[2] Zhu, B., Liu, S., Liu, Y., Luo, Y., et al. (2020). A software decoder implementation for H. 266/VVC video coding standard. arXiv preprint arXiv:2012.02832. https://doi.org/10.48550/arXiv.2012.02832

[3] Van Wallendael, G., Boho, A., De Cock, J., Munteanu, A., Van de Walle, R. (2013). Encryption for high efficiency video coding with video adaptation capabilities. IEEE Transactions on Consumer Electronics, 59(3): 634-642. https://doi.org/10.1109/TCE.2013.6626250

[4] Ouamri, M., Faraoun, K.M. (2014). Robust and fast selective encryption for HEVC videos. Journal of Communications Software and Systems, 10(4): 221-229. https://doi.org/10.24138/jcomss.v10i4.118

[5] Abu Taha, M., Hamidouche, W., Sidaty, N., Viitanen, M., et al. (2020). Privacy protection in real time HEVC standard using chaotic system. Cryptography, 4(2): 18. https://doi.org/10.3390/cryptography4020018

[6] Chen, C., Wang, X., Liu, G., Huang, G. (2022). A robust selective encryption scheme for H. 265/HEVC video. IEEE Access, 11: 17252-17264. https://doi.org/10.1109/ACCESS.2022.3210132

[7] Farajallah, M., Gautier, G., Hamidouche, W., Déforges, O., El Assad, S. (2022). Selective encryption of the versatile video coding standard. IEEE Access, 10: 21821-21835. https://doi.org/10.1109/ACCESS.2022.3149599

[8] Yu, J.Y., Kim, Y.G. (2023). Coding unit-based region of interest encryption in HEVC/h. 265 video. IEEE Access, 11: 47967-47978. https://doi.org/10.1109/ACCESS.2023.3276243

[9] Chen, C., Wang, X., Xu, J. (2023). A robust VVC video encryption scheme based on the dynamical chaotification model. Journal of King Saud University-Computer and Information Sciences, 35(9): 101752. https://doi.org/10.1016/j.jksuci.2023.101752

[10] Ibraheem, M.K.I., Abdalameer, A.K.I.M., Naji, A.A.Z.H. (2024). A genetic approach-based intra coding algorithm for H. 266/VVC. Informatics and Automation 23(3): 801-830. https://doi.org/10.15622/ia.23.3.6

[11] Shah, N., Nayak, A., Anklesaria, S., Maheshwari, U., et al. (2025). Enhancing video encryption with PIXJS algorithm: A chaos-based approach. Available at SSRN 5266379.

[12] Usmani, M.W., Shannigrahi, S., Zink, M. (2025). Securing immersive 360 video streams through attribute-based selective encryption. arXiv preprint arXiv:2505.04466. https://doi.org/10.48550/arXiv.2505.04466

[13] Hamidouche, W., Biatek, T., Abdoli, M., François, E., et al. (2022). Versatile video coding standard: A review from coding tools to consumers deployment. IEEE Consumer Electronics Magazine, 11(5): 10-24. https://doi.org/10.1109/MCE.2022.3144545

[14] Pfaff, J., Filippov, A., Liu, S., Zhao, X., et al. (2021). Intra prediction and mode coding in VVC. IEEE Transactions on Circuits and Systems for Video Technology, 31(10): 3834-3847. https://doi.org/10.1109/TCSVT.2021.3072430

[15] Sze, V., Budagavi, M., Sullivan, G.J.V. Sullivan, Integrated Circuits and Systems. http://www.springer.com/series/7236.

[16] Massoudi, A., Lefebvre, F., De Vleeschouwer, C., Macq, B., Quisquater, J.J. (2008). Overview on selective encryption of image and video: Challenges and perspectives. Eurasip Journal on Information Security, 2008(1): 179290. https://doi.org/10.1155/2008/179290

[17] Bross, B., Wang, Y.K., Ye, Y., Liu, S., et al. (2021). Overview of the versatile video coding (VVC) standard and its applications. IEEE Transactions on Circuits and Systems for Video Technology, 31(10): 3736-3764. https://doi.org/10.1109/TCSVT.2021.3101953

[18] Zhao, L., Zhao, X., Liu, S., Li, X., et al. (2019). Wide angular intra prediction for versatile video coding. In 2019 Data Compression Conference (DCC), Snowbird, UT, USA, pp. 53-62. https://doi.org/10.1109/DCC.2019.00013

[19] Huang, H. (2019). Novel scheme for image encryption combining 2d logistic-sine-cosine map and double random-phase encoding. IEEE Access, 7: 177988-177996. https://doi.org/10.1109/ACCESS.2019.2958319

[20] Fraunhofer HHI, VVenC - the Fraunhofer Versatile Video Encoder. https://www.hhi.fraunhofer.de/en/departments/vca/technologies-and-solutions/h266-vvc/fraunhofer-versatile-video-encoder-vvenc.html.

[21] Jovanović, B., Gajin, S. (2018). An efficient mechanism of cryptographic synchronization within selectively encrypted H. 265/HEVC video stream. Multimedia Tools and Applications, 77(2): 1537-1553. https://doi.org/10.1007/s11042-017-4389-3