

Experimental implementation of chaos-based video encryption using visible light communication technology

SOREL BAGIO NONO FOTSO,¹ JIMMI HERVÉ TALLA MBÉ,^{1,*}  JEAN ERNEST TAKOTUE,^{1,2} ARMAND CYRILLE NZEUKOU TAKOUGANG,^{1,3} AND ALAIN RICHARD NDJIONGUE⁴ 

¹Research Unit of Condensed Matter, Electronics and Signal Processing, Department of Physics, University of Dschang, P.O. Box 67, Dschang, Cameroon

²Research Unit of Automation and Applied Computer, Department of Electrical Engineering, IUT-FV Bandjoun, University of Dschang, P. O. Box: 134, Bandjoun, Cameroon

³Laboratory of Industrial Systems and Environmental Engineering, Fotso Victor University Institute of Technology, University of Dschang, P.O. Box 134, Bandjoun, Cameroon

⁴School of Electrical and Information Engineering, Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, South Africa

*jhtallam@yahoo.fr

Received 11 October 2024; accepted 18 December 2024; posted 23 December 2024; published 30 January 2025

Simulations are commonly used to develop and evaluate video encryption algorithms. Although these approaches are useful for demonstrating theoretical feasibility and algorithm performance, they neglect practical challenges and real-world communication conditions. This oversight creates a critical gap in evaluating their effectiveness and security in practical applications. In this paper, we describe a novel, to the best of our knowledge, video encryption scheme that employs a modified chaotic Colpitts oscillator, a customized Josephus problem, and ribonucleic acid (RNA) operations, and is implemented in visible light communication (VLC) environment. The modified Colpitts oscillator reveals significant phenomena, the most prominent for this study being chaos. The encryption scheme involves block scrambling and pixel-level permutation employing a customized Josephus problem and a column and row shifting technique. Diffusion is then achieved through a custom Josephus problem-oriented RNA operation combined with basic RNA operations and cipher block chaining (CBC). Based on the results, we found that the customized Josephus problem-based permutation performed more efficiently than traditional methods, whereas the column and row shifting permutation was more robust. By eliminating key sequence coding and item-to-item operations, the proposed RNA operation reduces computational overhead. This video encryption scheme is proven to be effective and resilient at securing video data transmitted over the VLC channel, and it constitutes a significant advancement in areas such as surveillance, medical imaging, and military communication requiring robust secured video data. © 2025 Optica Publishing Group. All rights, including for text and data mining (TDM), Artificial Intelligence (AI) training, and similar technologies, are reserved.

<https://doi.org/10.1364/AO.543978>

1. INTRODUCTION

The rapid advancement of digital technologies and communication networks has led to a substantial increase in the volume of digital data exchanged across diverse networks. Among these data types, video, characterized as a voluminous three-dimensional dataset, assumes pivotal roles in sensitive sectors such as medicine, national defense, legal affairs, and surveillance [1–3]. Securing video data during transmission and storage has become a critical challenge. Traditional encryption schemes such as the data encryption standard (DES) and advanced encryption standard (AES) often face difficulties in effectively

securing video data due to the inherent high correlation and redundancy within video datasets.

Recent advancements in video encryption have introduced various schemes employing diverse methodologies to enhance data security. Notable works include a modified AES scheme for digital video encryption, which demonstrated reduced encryption and decryption times alongside an expanded key space [4]. Aggarwal *et al.* presented a lossless encryption approach using column and row shifting, showcasing robust security in experimental evaluations [5]. Yasser *et al.* proposed a chaotic-based encryption scheme utilizing 2D alteration models, emphasizing encryption quality and sensitivity [6]. However, its performance needs to be further investigated against other

attacks. Furthermore, Valli *et al.* introduced two chaos-based video encryption schemes, one using the Ideka delay differential equation and the other employing a 12-D chaotic map [7]. The findings show that the schemes exhibit resilience against statistical, differential, and chosen/known plaintext attacks. Elkamchouchi *et al.* introduced two video encryption algorithms employing different chaotic maps for keystream generation, highlighting trade-offs between security and computational efficiency [8]. The proposed schemes are shown to be resistant to known attacks. However, the processing time needs improvement to meet real-time applications. Malladar *et al.* proposed a selective encryption scheme using 1-D logistic maps for securing H.264 video, employing a lookup table for efficient data conversion [9]. In the realm of quantum encryption, Song *et al.* proposed a scheme based on qubit-planes and an improved logistic map, enhancing encryption strength [10]. Malood *et al.* introduced a lightweight stream video encryption scheme combining chaotic maps and the ChaCha20 algorithm, achieving heightened security with reduced computation time [11]. Nevertheless, further analysis is needed regarding performance against noise and data loss attacks. Hosny *et al.* developed a video encryption scheme based on a chaotic logistic map, employing zigzag sweeps, block rotations, and random block arrangements to ensure high security and efficiency [12]. However, the proposed scheme appeared vulnerable to brute-force attacks due to the small key space. Li *et al.* proposed a chaotic encryption scheme tailored for the high-efficiency video coding standard, demonstrating effective protection of video data [13]. Yun *et al.* introduced a permutation-based encryption algorithm updating permutation lists for each frame using a crypto-secure pseudo-random number generator, minimizing memory overhead [14]. Cheng *et al.* proposed a selective encryption scheme combining video coding and encryption algorithms, affirming robustness through theoretical analysis [15]. El-den *et al.* presented a video scrambling technique employing two chaotic symmetric maps and a chaotic tent map, demonstrating resistance to various attacks [16]. Heshmat *et al.* proposed a video encryption method integrating Henson, elliptic curve, and logistic map advantages, achieving requisite security levels and resilience against diverse attacks [17]. Hosny *et al.* introduced an innovative approach to enhance privacy in surveillance videos by combining block scrambling-based encryption with object detection using a deep convolutional neural network (YOLOv3). The methodology involves face detection using YOLOv3, followed by scrambling the detected faces with fast block scrambling techniques. Subsequently, the scrambled faces are encrypted using a secret key derived from a chaotic logistic map [18]. Results demonstrated effective detection and protection of faces without leaks, with resilience against potential attacks, though real-time application execution time remains a concern. Afsa *et al.* proposed an improved AES, enhancing diffusion and confusion in ciphered videos [19]. Their findings indicate that the scheme overcame limitations of existing AES methods, offering a balance between speed and security for video transmission. Li *et al.* implemented a video encryption scheme based on hyperchaotic systems [20]. This scheme divided color images into Y, Cb, and Cr channels, encrypting Y with Arnold transforms and a DNA encoding algorithm, and Cb and Cr with a Lorenz hyperchaotic map. Experimental

results validated improved security and encryption efficiency compared to existing methods. Alarifi *et al.* proposed a novel hybrid cryptosystem combining DNA sequences, Arnold chaotic maps, and Mandelbrot sets for secure streaming of compressed high-efficiency video coding (HEVC) streams. Extensive simulations and security analyses substantiated the robustness and security of the proposed HEVC cryptosystem.

However, while theoretical advancements are crucial for advancing encryption technology, translating these theories into real-world applications is essential to validate the robustness and effectiveness of encryption schemes in protecting sensitive data across practical communication settings. The lack of implementation of previous cryptosystems in real communication systems raises doubts about their efficacy in real-life scenarios. In our recently published work [21], we introduced a novel approach for audio data encryption, with a particular focus on transmission through VLC communication systems. While our methodology has proven effective, it is crucial to emphasize that the specific case of video encryption, implemented and tested within a real-world VLC communication system, remains largely underexplored in the existing literature. Moreover, due to the predictable trajectories of 1-D chaotic systems, video encryption schemes based on them are generally less secure compared to those founded on higher-dimensional systems. Additionally, several schemes may struggle to effectively resist plaintext and ciphertext attacks because they do not adequately account for the inherent structure of plain video data.

This paper presents a novel video encryption scheme implemented and assessed in a VLC infrastructure. The proposed scheme is based on a modified chaotic Colpitts oscillator, a customized Josephus problem, and ribonucleic acid (RNA) operations. The encryption process begins with the conversion of an RGB video into YCbCr format, enabling distinct processing levels for each channel. Subsequently, pixel-level permutation is applied to the Y channel using a column and row shifting approach, while the Cb and Cr channels undergo block scrambling using a customized Josephus problem-based permutation. This is followed by a diffusion process that incorporates a customized Josephus problem-based RNA operation along with basic RNA operations and CBC mode. Furthermore, synchronization of the modified Colpitts oscillator to the plain video and the computer's timestamp is achieved using the SHA-3-512 hash function, enhancing result unpredictability. The key parameters of the proposed scheme are secured through the RSA public key cryptosystem, enabling robust asymmetric encryption. The VLC infrastructure consists of two main components: software, referred to as the VLC application; and hardware, identified as the VLC modem. In summary, this paper introduces several key contributions and innovations:

- Implementation and evaluation of the proposed video encryption scheme in a real-world communication system, demonstrating its practical applicability;
- Introduction of a modified chaotic Colpitts oscillator for robust encryption;
- Introduction of a versatile column and row shifting approach that obscures the original data structure and patterns, making it difficult for unauthorized parties to decipher or detect

meaningful information, thereby increasing overall encryption scheme complexity and security;

- Development of a customized Josephus problem-based permutation that enhances the diversity of generated sequences and introduces additional dislocation effects, thereby improving the permutation's effectiveness;
- Novel RNA operation based on a customized Josephus problem to reduce computational overhead, eliminating the typical need for key sequence coding and item-to-item operations;
- Enhancement of system unpredictability by synchronizing the chaotic system with the computer's timestamp using the SHA3-512 hash function;
- Design of a robust video encryption scheme;
- Achievement of superior performance metrics compared to various existing schemes in the literature.

The remainder of this paper is structured as follows. Section 2 describes the proposed system and model. Section 3 presents the experimental results and comparative analysis. Finally, concluding remarks are provided in Section 4.

2. SYSTEM AND MODEL

In this section, we delineate the system architecture and model framework that underpin our novel approach to video encryption, integrating a modified chaotic Colpitts oscillator, a customized Josephus problem, and RNA operations within a VLC infrastructure.

A. Background of the Proposed Encryption Scheme

1. RGB to YCbCr Image Transformation

The RGB color space commonly utilized in computer graphics is suboptimal for image processing due to mutual redundancy among its component signals. To address this inefficiency, numerous video standards, such as YCbCr, have been developed. The YCbCr color space separates luminance (Y) from chrominance components (Cr, red-difference; Cb, blue-difference), allowing for independent processing of these components. This separation enhances system efficiency by enabling high-resolution storage of the luminance component while independently processing the chroma components. The linear transformation from the RGB color model to the YCbCr color space is defined by Eq. (1) [22]:

$$\begin{cases} Y = \frac{16 + (65.738 \times R + 129.057 \times G + 25.064 \times B)}{256} \\ Cb = \frac{128 - (37.945 \times R + 74.494 \times G - 112.439 \times B)}{256} \\ Cr = \frac{128 + (112.439 \times R - 94.154 \times G - 18.285 \times B)}{256} \end{cases} \quad (1)$$

The reverse operation is given by

$$\begin{cases} R = \frac{(298.082 \times Y + 408.583 \times Cr)}{256} - 222.921 \\ G = \frac{(298.082 \times Y - 100.291 \times Cb - 208.120 \times Cr)}{256} + 135.576 \\ B = \frac{(298.082 \times Y + 516.412 \times Cb)}{256} - 276.836 \end{cases} \quad (2)$$

2. SHA-3-512

SHA-3-512 is a robust and extensively utilized cryptographic algorithm that transforms any given input into a fixed-length 512 bit output in hexadecimal format. Its strong avalanche effect ensures that even a slight change in the input produces a significantly different hash value, making it virtually impossible to deduce the original data from the hash output [21,23,24].

3. Computer Timestamping

Computer timestamping utilizes an internal clock powered by a real-time clock chip and a dedicated battery, enabling the operating system to regularly retrieve the current time and date. This facilitates event marking, process synchronization, and precise file management with accurate timestamps [25].

4. Modified Chaotic Colpitts Oscillator

A chaotic Colpitts oscillator is a type of electronic oscillator that utilizes a transistor as the active, nonlinear component, coupled with inductors and capacitors to generate oscillations at a specific frequency [26,27]. The electrical schematic of the modified chaotic Colpitts oscillator, which is derived from the model proposed in [28], is illustrated in Fig. 1.

Either:

- v_{C_1} , v_{C_2} , and v_{C_3} are the respective voltages across the capacitors C_1 , C_2 and C_3 ;
- i_{L_1} and i_{L_2} are the respective currents in the inductances L_1 and L_2 .

The application of Kirchhoff's laws to the circuit of Fig. 1 leads to the following set of five first-order differential equations that characterize its dynamics:

$$\begin{cases} \frac{dv_{C_1}}{dt} = \frac{1}{C_1} (i_{L_1} - \alpha i_e) \\ \frac{dv_{C_2}}{dt} = \frac{1}{C_2} (i_{L_1} - I_0 + (1 - \alpha i_e)) \\ \frac{dv_{C_3}}{dt} = \frac{1}{C_3} (I_0 - i_{L_2} - i_e) \\ \frac{di_{L_1}}{dt} = \frac{1}{L_1} (V_{cc} - Ri_{L_1} - v_{C_1} - v_{C_2}) \\ \frac{di_{L_2}}{dt} = \frac{1}{L_2} v_{C_3} \end{cases} \quad (3)$$

In Eq. (3), i_e is a component introducing the nonlinearity as

$$i_e = I_s \left(-1 + e^{\left(\frac{v_{C_3} - v_{C_2}}{V_{ref}} \right)} \right). \quad (4)$$

By introducing Eq. (4) into Eq. (3), the system becomes

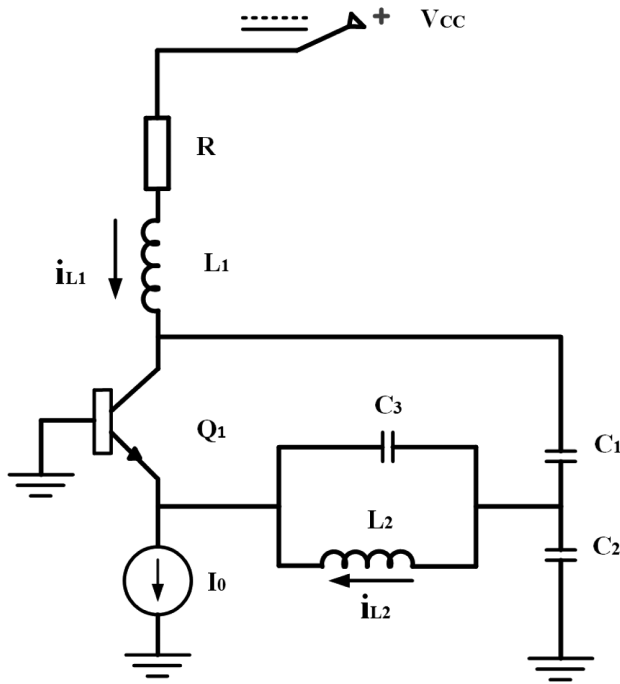


Fig. 1. Electrical diagram of the modified Colpitts oscillator.

$$\begin{cases} \frac{dv_{C1}}{dt} = \frac{1}{C_1} \left(i_{L1} - \alpha I_s \left(-1 + e^{\left(\frac{v_{C3} - v_{C2}}{V_{ref}} \right)} \right) \right) \\ \frac{dv_{C2}}{dt} = \frac{1}{C_2} \left(i_{L1} - I_0 + \left(1 - \alpha I_s \left(-1 + e^{\left(\frac{v_{C3} - v_{C2}}{V_{ref}} \right)} \right) \right) \right) \\ \frac{dv_{C3}}{dt} = \frac{1}{C_3} \left(I_0 - i_{L2} - I_s \left(-1 + e^{\left(\frac{v_{C3} - v_{C2}}{V_{ref}} \right)} \right) \right) \\ \frac{di_{L1}}{dt} = \frac{1}{L_1} (V_{cc} - Ri_{L1} - v_{C1} - v_{C2}) \\ \frac{di_{L2}}{dt} = \frac{1}{L_2} v_{C3} \end{cases} \quad (5)$$

Applying the following rescaling:

$$\begin{aligned} \tau &= \frac{t}{\sqrt{L_1 C_1}}, \quad \rho = \sqrt{\frac{L_1}{C_1}}, \quad \xi_1 = \frac{C_1}{C_2}, \quad \xi_2 = \frac{C_1}{C_3}, \quad \xi_3 = \frac{L_1}{L_2}, \\ \xi_4 &= \frac{\rho I_0}{V_T}, \quad \xi_5 = \frac{R}{\rho} x_i = \frac{v_{C_i} - v_{C_i}^*}{V_T}, \quad x_{j+3} = \frac{i_{L_j} - i_{L_j}^*}{V_T}. \end{aligned} \quad (6)$$

Equation (3) yields:

$$\begin{cases} \dot{x} = v - \alpha \xi_4 (-1 + e^{(y-u)}) \\ \dot{u} = \xi_1 (v + \xi_4 (1 - \alpha) (-1 + e^{(y-u)})) \\ \dot{y} = \xi_2 (-w - \xi_4 (-1 + e^{(y-u)})) \\ \dot{v} = -x - u - \xi_5 v \\ \dot{w} = \xi_3 v. \end{cases} \quad (7)$$

The bifurcation diagram depicting the state variable $w(\tau)$ as a function of the control parameter ξ_5 is shown in Fig. 2.

These findings are further substantiated by some phase portraits displayed in Fig. 3. Figures 3(a) and 3(b) depict chaotic orbits. Notably, the chaotic Colpitts oscillator exhibits chaos within a large range of the bifurcation parameter ($0.1 < \xi_5 < 0.6$). These chaotic characteristics underscore the suitability of the proposed modified Colpitts oscillator for applications in chaos-based data encryption [29–33].

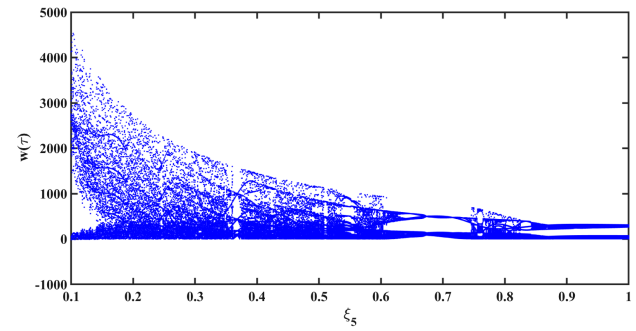


Fig. 2. Bifurcation diagram of the state variable $w(\tau)$ as a function of the control parameter ξ_5 .

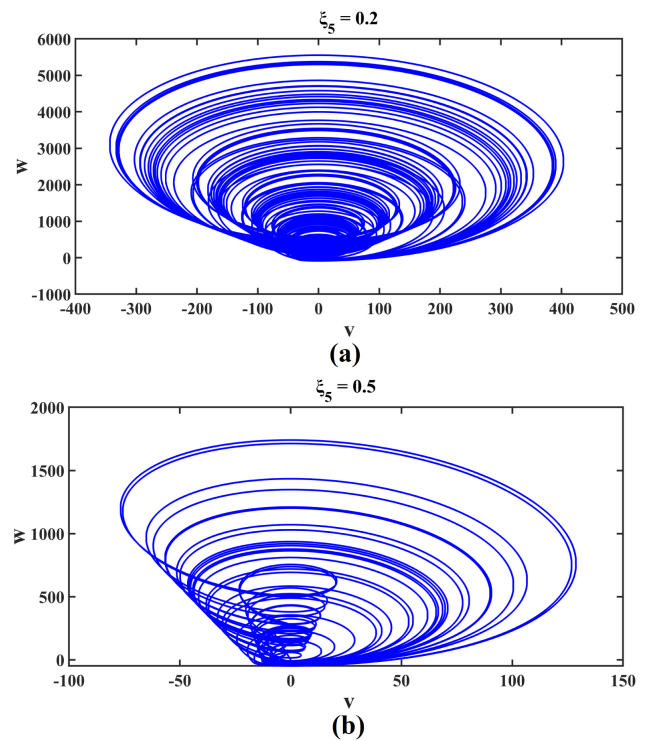


Fig. 3. Phase portraits on the (u, ξ_5) plane for various values: (a) chaotic behavior for $\xi_5 = 0.2$; (b) periodic pair behavior for $\xi_5 = 0.5$.

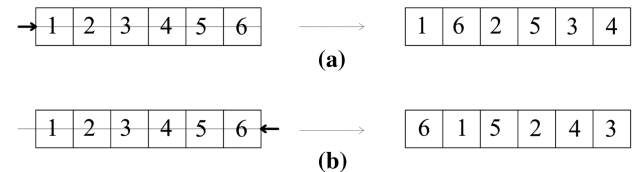


Fig. 4. Results obtained after scanning: (a) row scanned from left to right; (b) row scanned from right to left.

5. Proposed Scrambling Schemes

a. Column and row shifting-based scrambling scheme

This scrambling scheme dynamically scans the columns and rows of matrices. The scan involves successively

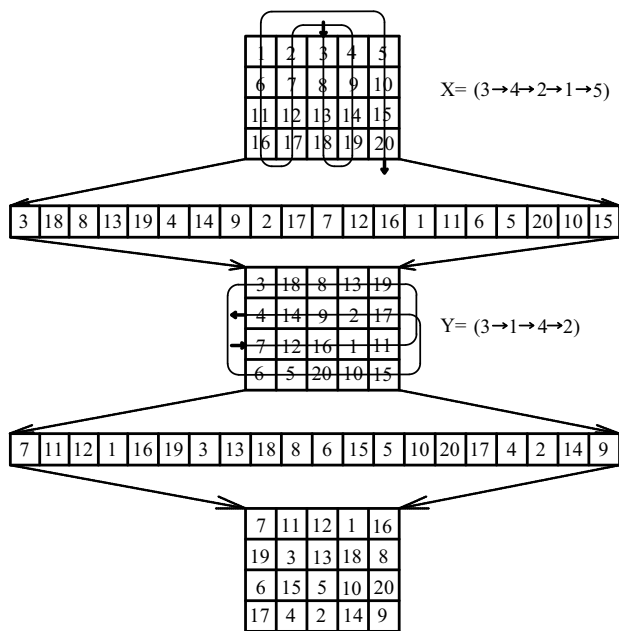


Fig. 5. Proposed scrambling scheme applied to a 5×4 matrix with X and Y representing the column and row selection order, respectively.

selecting the first element, then the last, followed by the second, then the penultimate, the third, and continuing in this manner until all elements in the row or column are exhausted. Figure 4 illustrates this process. Subsequently, the resulting vector is reshaped into another matrix of the same dimensions as the original, upon which a column/row scan is applied. This entire process is iterated k times. Figure 5 illustrates the application of the proposed scrambling process to a matrix with dimensions 5×4 for $k = 1$. After the scrambling process, it becomes evident that the positions of all elements have been altered. As a result, this scrambling scheme obscures the original data structure and patterns, making it difficult for unauthorized parties to decipher or detect meaningful information, thereby increasing overall encryption scheme complexity and security. Figure 6 shows the application of this encryption process to a grayscale image (image here just for illustration) for three values of k . It is clear that the proposed scheme gives better results and also scrambles the image perfectly for $k = 3$.

b. Customized Josephus problem-based permutation

The Josephus problem pertains to a scenario where a group of N individuals stands in a circle awaiting execution. Sequential counting occurs in a fixed direction, with every K^{th} person skipped until only one remains, the survivor [34]. Recently, this problem has found application in image encryption schemes [35]. In the traditional Josephus problem, the step size K remains constant, and the scan progresses either clockwise or counterclockwise. In contrast, the customized Josephus problem used in this study introduces variability by randomly changing the starting position, step size, and direction (see Fig. 7). This approach

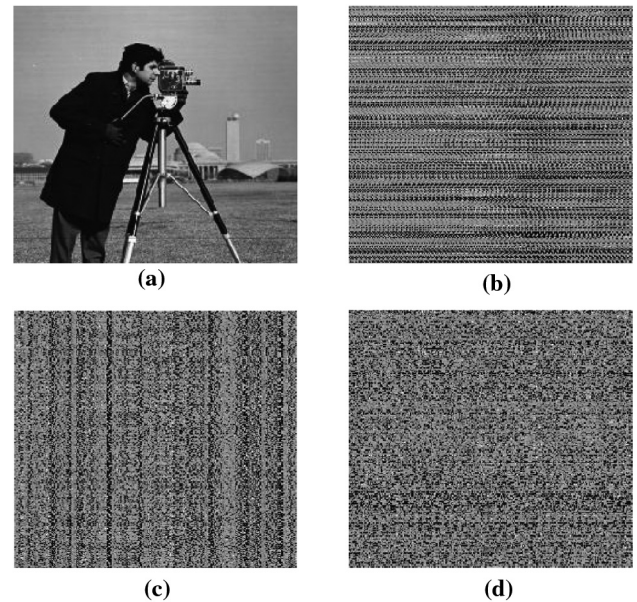


Fig. 6. Graphical output of the proposed scrambling scheme: (a) plain image, (b) scrambled image ($k = 1$), (c) scrambled image ($k = 2$), (d) scrambled image ($k = 3$).

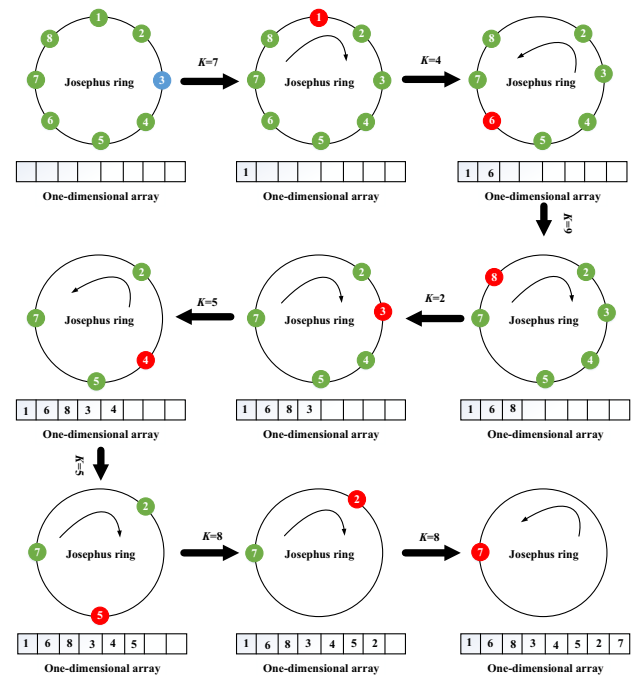


Fig. 7. An illustration of the proposed customized Josephus problem with $N = 8$.

enhances the diversity of generated sequences and introduces additional dislocation effects, thereby improving the permutation's effectiveness.

6. Customized Josephus Problem-based RNA Operation

RNA sequencing is a genomic approach used to detect and analyze messenger RNA molecules present in biological samples.

Table 1. RNA Encoding and Decoding Map Rules

Rule	1	2	3	4	5	6	7	8
00	A	A	U	U	C	C	G	G
01	C	G	C	G	A	U	A	U
10	G	C	G	C	U	A	U	A
11	U	U	A	A	G	G	C	C

Table 2. RNA Operations Results

Operation	Base	A	C	G	U
Addition	A	C	A	U	G
	C	A	C	G	U
	G	U	G	C	A
	U	G	U	A	C
Subtraction	A	C	G	U	A
	C	A	C	G	U
	G	U	A	C	G
	U	A	C	G	U
Multiplication	A	U	G	C	A
	C	G	U	A	C
	G	C	A	U	G
	U	A	C	G	U
XOR	A	A	C	G	U
	C	C	A	U	G
	G	G	U	A	C
	U	U	G	C	A
XNOR	A	U	G	C	A
	C	G	U	A	C
	G	C	A	U	G
	U	A	C	G	U
Complement	A	U			
	C	G			
	G	C			
	U	A			

Today, its applications include medicine, cryptography, and more [36]. The RNA sequence is composed of four nucleic acids including adenine (A), uracil (U), cytosine (C), and guanine (G). Table 1 presents the encoding and decoding map rules for the RNA sequence [37,38]. There are also RNA operations, such as addition, subtraction, and multiplication commonly used in RNA computing as shown in Table 2.

Furthermore, in protein synthesis, the three adjacent bases that identify amino acids in the messenger RNA chain are known as genetic codons [39]. Theoretically, there are 64 codons, as shown in Table 3.

The proposed RNA operation is grounded in the customized Josephus problem described in Section 2.A.5. This operation entails replacing each codon in an input array with another codon to produce an output array. To achieve this, 64 codons are systematically arranged in ascending order along the Josephus ring, each uniquely identified by its index. The substitution process initiates at the index of the codon being replaced, with the surviving codon representing the substituting codon. Matrices governing step sizes and directions are derived using two chaotic sequences. Figure 8 illustrates the application of this RNA operation to an RNA array. The direction matrix comprises 64 columns denoting direction (1 for clockwise and

0 for counterclockwise), while the step size matrix includes integers ranging from 1 to 64. Each row in these matrices denotes successive directions and step sizes used to determine the surviving codon for each input codon. This RNA operation reduces computational overhead, eliminating the need for key sequence coding and item-to-item operations, as usually required in several encryption schemes.

7. Cipher Block Chaining Mode

Block cipher modes involve a process of combining previous data blocks with the next one. This iterative process ensures that the encrypted plaintext is unique through a randomly generated initialization vector (IV). There are different block cipher modes of operation, but this paper focuses on the cipher block chaining (CBC) mode. This mode uses a single IV and XORs the previous cipher block with the current cipher block before the block cipher encryption [40].

B. Proposed Encryption Scheme

The steps involved in the proposed video encryption system are depicted in Fig. 9. Typically, video files consist of a sequence of frames, each represented by a 2D array of pixels. We denote the matrix set of plain videos as V_P and that of encrypted videos as V_E , both having the dimensions $[M, N, O, D]$, where M , N , O , and D are all positive integers.

1. Video Data Preprocessing

The proposed encryption scheme transforms an RGB video V_P to a YCbCr video V_3 as

$$\begin{cases} V_2 = \text{rgb2ycbcr}(V_P(:, :, j)) \\ V_3(:, :, 1, j) = V_2(:, :, 1) \\ V_3(:, :, 2, j) = V_2(:, :, 2) \\ V_3(:, :, 3, j) = V_2(:, :, 3), \end{cases} \quad (8)$$

where $j = 1, 2, 3, \dots, D$, with D the number of frame in V_P . Then, one gets the Y , Cb , Cr components of V_3 as

$$\begin{cases} Y = V_3(:, :, 1, :) \\ Cb = V_3(:, :, 2, :) \\ Cr = V_3(:, :, 3, :). \end{cases} \quad (9)$$

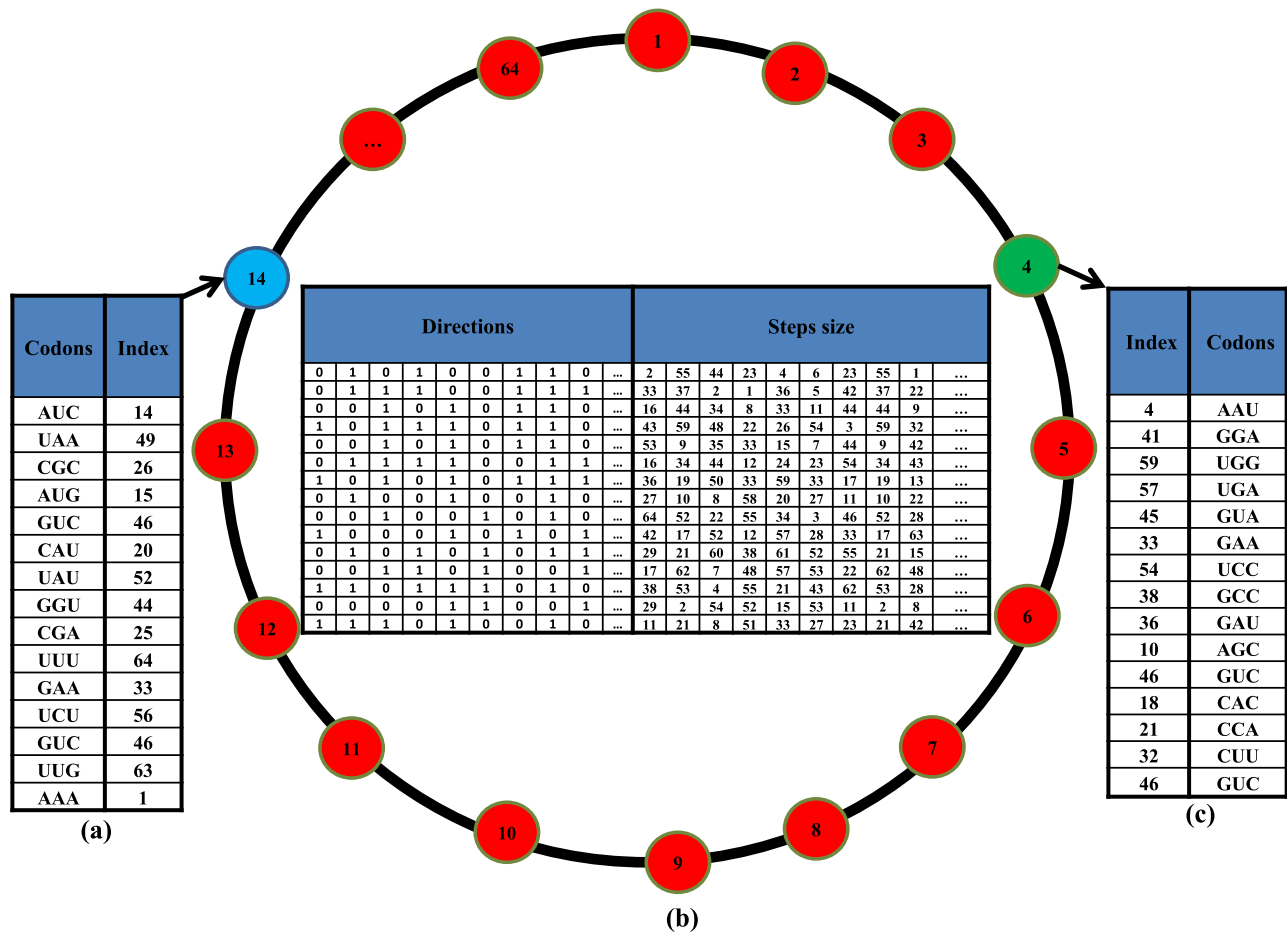
In this color space, the Y channel (luminance component) contains substantial information, while C_b (blue chromaticity component) and C_r (red chromaticity component) contain relatively less information. Consequently, our encryption scheme applies varying degrees of processing strength to each channel for enhanced efficiency. It is noteworthy that pixel values of Y range from 16 to 235, whereas those of C_b and C_r range from 16 to 240.

2. Key Parameters and System Initial Value Generation

The key parameters of the proposed encryption scheme include: (1) two 512-bit values, γ_1 and γ_2 , obtained by applying the SHA3-512 hash function to the plain video V_P and

Table 3. Codons Table

Index	Codons	Index	Codons	Index	Codons	Index	Codons
1	AAA	17	CAA	33	GAA	49	UAA
2	AAC	18	CAC	34	GAC	50	UAC
3	AAG	19	CAG	35	GAG	51	UAG
4	AAU	20	CAU	36	GAU	52	UAU
5	ACA	21	CCA	37	GCA	53	UCA
6	ACC	22	CCC	38	GCC	54	UCC
7	ACG	23	CCG	39	GCG	55	UCG
8	ACU	24	CCU	40	GCU	56	UCU
9	AGA	25	CGA	41	GGA	57	UGA
10	AGC	26	CGC	42	GGC	58	UGC
11	AGG	27	CGG	43	GGG	59	UGG
12	AGU	28	CGU	44	GGU	60	UGU
13	AUA	29	CUA	45	GUA	61	UUA
14	AUC	30	CUC	46	GUC	62	UUC
15	AUG	31	CUG	47	GUG	63	UUG
16	AUU	32	CUU	48	GUU	64	UUU

**Fig. 8.** Proposed RNA operation applied to an array: (a) input RNA array, (b) Josephus ring containing step sizes and directions matrix, (c) output RNA array.

the computer timestamp ST , respectively; and (2) the system's parameters. The ST format follows "yyyy/MM/dd HH:mm:ss.SSS". Subsequently, the initial values of the modified chaotic Colpitts oscillator are computed accordingly:

$$\begin{cases} u(1) = \text{mod}(a, 128^b) \\ v(1) = \text{mod}(b, 128^d) \\ w(1) = \text{mod}(c, 128^a) \\ x(1) = \text{mod}(d, 128^c) \\ y(1) = \text{mod}(e, 128^a) \end{cases} \quad (10)$$

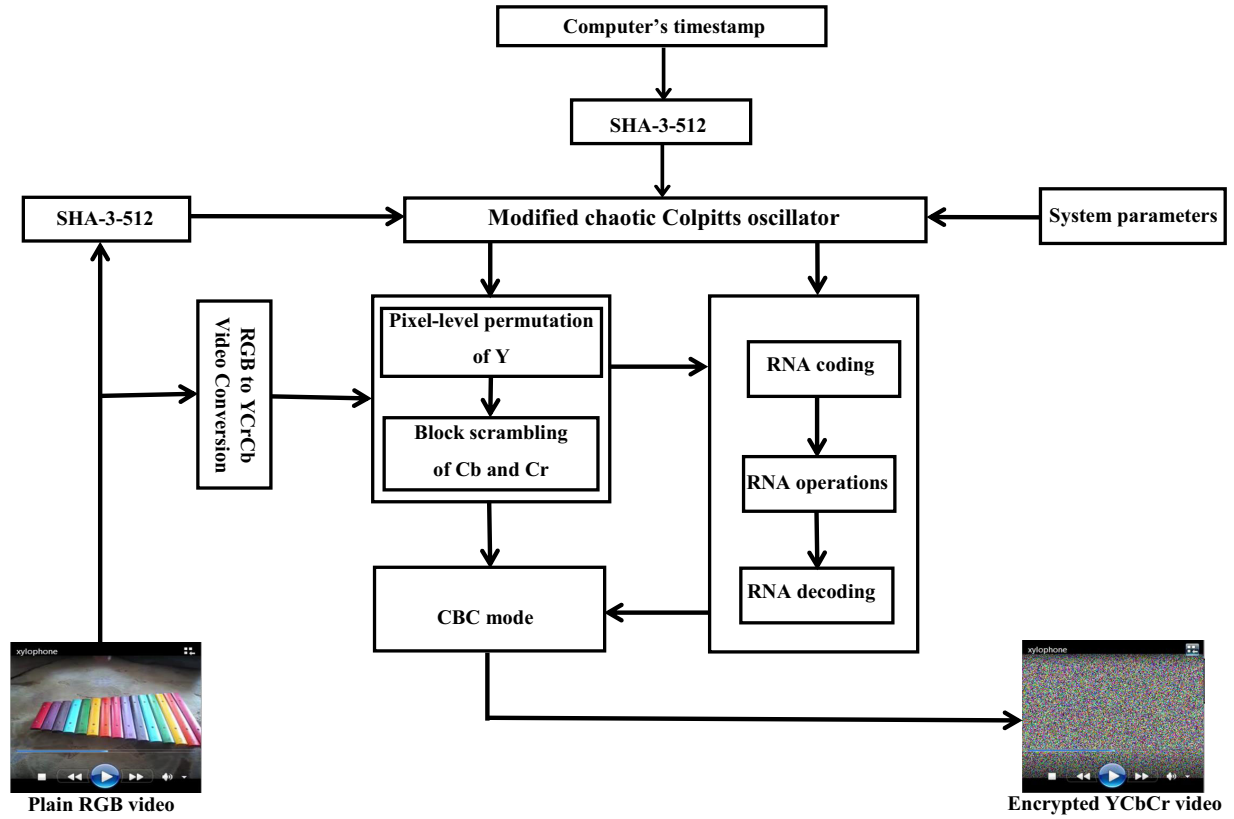


Fig. 9. Flowchart of the encryption process; the decryption process is the reverse.

where

$$\begin{cases} a = H2D(\gamma_1(1)) + H2D(\gamma_1(2)) + \dots + H2D(\gamma_1(64)) \\ b = H2D(\gamma_2(65)) + H2D(\gamma_2(66)) + \dots + H2D(\gamma_2(128)) \\ c = H2D(\gamma_2(1)) + H2D(\gamma_2(2)) + \dots + H2D(\gamma_2(64)) \\ d = H2D(\gamma_1(65)) + H2D(\gamma_1(66)) + \dots + H2D(\gamma_1(128)), \end{cases} \quad (11)$$

with “mod” denoting the modulo function and “ $H2D$ ” representing a function converting a hexadecimal number in textual form to its decimal equivalent.

3. Scrambling Process

The proposed encryption scheme implements pixel-level permutation on the Y channel and block scrambling on Cr and Cb channels as follows.

Step 1: Obtain the dimensions of the video file as $[M, N, O, D] = \text{size}(V_p)$;

Step 2: Iterate Eq. (7) for $M \times N \times D + 1000$ iterations and discard the first 1000 values;

Step 3: Compute:

$$\begin{cases} K_c1(i) = \text{round}(\text{rescale}(\text{mod}(\text{floor}(((\text{abs}([u(i) \times v(i)] - \text{floor}(\text{abs}([u(i) \times v(i)]))) \times 10^{15})/10^8), 256), 0, 255))) \\ K_c2(i) = \text{round}(\text{rescale}(\text{mod}(\text{floor}(((\text{abs}([x(i) \times y(i)] - \text{floor}(\text{abs}([x(i) \times y(i)]))) \times 10^{15})/10^8), 256), 0, 255))), \end{cases} \quad (12)$$

Step 4: Reshape Y , Cb , and Cr into 2D arrays Y_1 , Cb_1 , and Cr_1 , each of dimensions $[M, N \times D]$, as follows:

$$\begin{cases} Y_1 = \text{reshape}(Y, [M, N \times D]) \\ Cb_1 = \text{reshape}(Cb, [M, N \times D]) \\ Cr_1 = \text{reshape}(Cr, [M, N \times D]), \end{cases} \quad (13)$$

Step 5: Randomly alter the position of each pixel in Y_1 using the column and row shifting approach proposed in A.5 to obtain Y_2 . The orders of column and row selections V_c and V_r , along with the iteration count k , are specified as follows:

$$\begin{cases} V_c = \text{sort}(K_c1(1 : N \times D), \text{ascend}) \\ V_r = \text{sort}(K_c2(1 : M), \text{descend}) \\ k = \text{index}(\text{max}(w(1 : 10))). \end{cases} \quad (14)$$

The function “sort” arranges the elements in the specified segment of K_{c1} and K_{c2} in ascending or descending order, while also storing their respective indices. The function max determines the maximum value within a vector, and “index” identifies the position of a specific element within a vector.

Step 6: Divide the plain 2D arrays Cb_1 and Cr_1 into subarrays $Cb_2\{i\}$ and $Cr_2\{j\}$ of sizes $[M_r1, N_c1]$ and $[M_r2, N_c2]$, respectively:

$$\begin{cases} Cb_2\{i\} = Cb_1((r_1 - 1) \times M_r1 + 1 : M_r1 \times r_1, (c_1 - 1) \times N_c1 + 1 : N_c1 \times c_1, :) \\ Cr_2\{j\} = Cr_1((r_2 - 1) \times M_r2 + 1 : M_r2 \times r_2, (c_2 - 1) \times N_c2 + 1 : N_c2 \times c_2, :), \end{cases} \quad (15)$$

where $i = 1, 2, 3, \dots$, $j = 1, 2, 3, \dots$, $r_1 = 1, 2, \dots, M_{r1}$, $r_2 = 1, 2, \dots, M_{r2}$, $c_1 = 1, 2, \dots, N_{c1}$, and $c_2 = 1, 2, \dots$,

Algorithm 1. Algorithm for determining subarray sizes
Input: M, N, D
Output: $M_{r1}, M_{r2}, N_{c1}, N_{c2}$

 1: **for** $k = 1$ to $M - 1$ **do**

 2: Compute $a = \frac{M}{(M-k)}$

 3: **if** $is_integer(a)$ **then**

 4: $A(k) = a$

 5: **for** $l = 1$ to $N \times D - 1$ **do**

 6: Compute $b = \frac{N \times D}{(N \times D - l)}$

 7: **if** $is_integer(b)$ **then**

 8: $B(l) = b$

 9: Randomly select four integers: M_{r1} and M_{r2} from set A , and N_{c1} and N_{c2} from set B .

N_{c2} . The values of M_{r1} , N_{c1} , M_{r2} , and N_{c2} are randomly determined using Algorithm 1.

Step 7: Randomly permute the positions of each subarray in $Cb_2\{i\}$ and $Cr_2\{j\}$ using the customized Josephus problem (denoted as J_P) as described in Subsection 2.A.5, resulting in $Cb_3\{i\}$ and $Cr_3\{j\}$:

$$\begin{cases} Cb_3\{i\} = J_P(Cb_2\{i\}, [\text{start_pos_1}, \text{dir_1}, \text{step_size_1}]) \\ Cr_3\{j\} = J_P(Cr_2\{j\}, [\text{start_pos_2}, \text{dir_2}, \text{step_size_2}]) \end{cases} \quad (16)$$

where the initial positions ($\text{start_pos_1}, 2$), directions ($\text{dir_1}, 2(i)$), and step sizes ($\text{step_size_1}, 2(i)$) are specified as follows:

$$\begin{cases} \text{start_pos_1} = \text{index}(\max(u(1 : N_1))) - 1 \\ \text{start_pos_2} = \text{index}(\max(v(1 : N_2))) - 1 \end{cases} \quad (17)$$

$$\begin{cases} \text{dir_1}(i) : \text{clockwise if } x(i) \leq w(i), \text{ anti-clockwise, else} \\ \text{dir_2}(i) : \text{clockwise if } u(i) \leq y(i), \text{ anti-clockwise, else,} \end{cases} \quad (18)$$

$$\begin{cases} \text{step_size_1}(i) = \text{mod}(u(i) + v(i), N_1) + 1 \\ \text{step_size_2}(i) = \text{mod}(w(i) + x(i), N_2) + 1 \end{cases} \quad (19)$$

where N_1 and N_2 denote the number of subarrays in $Cb_3\{i\}$ and $Cr_3\{j\}$, respectively.

Step 8: Construct two 2D arrays, Cb_4 and Cr_4 , representing the scrambled arrays as follows:

$$\begin{cases} Cb_4((r_1 - 1) \times M_{r1} + 1 : M_{r1} \times r_1, (c_1 - 1) \times N_{c1} + 1 : N_{c1} \times c_1, :) = Cb_3\{i\} \\ Cr_4((r_2 - 1) \times M_{r2} + 1 : M_{r2} \times r_2, (c_2 - 1) \times N_{c2} + 1 : N_{c2} \times c_2, :) = Cr_3\{j\} \end{cases} \quad (20)$$

with $i = 1, 2, 3, \dots, M_{r1} \times N_{c1}$, $r_1 = 1, 2, \dots, M_{r1}$, $c_1 = 1, 2, \dots, N_{c1}$, and $j = 1, 2, 3, \dots, M_{r2} \times N_{c2}$, $r_2 = 1, 2, \dots, M_{r2}$, $c_2 = 1, 2, \dots, N_{c2}$.

4. Diffusion Process

The diffusion process is implemented through RNA encoding and CBC mode.

Step 1: Convert Y_2 and K_{c1} into sequences of bits, and concatenate each to form two 1-D binary vectors;

Step 2: RNA encode the binary vectors Y_2 and K_{c1} to generate $Y_{2\text{RNA}}$ and $K_{c1\text{RNA}}$, respectively, as

$$\begin{cases} Y_{2\text{RNA}}(i) = [Y_2(k : 2 \times i), \text{Enc_rule}(i)] \\ K_{c1\text{RNA}}(i) = [K_{c1}(k : 2 \times i), \text{Enc_rule}(i)] \end{cases} \quad (21)$$

where $i = 1, 2, 3, \dots, M \times N \times D$, and $k = 1, 3, 5, \dots, M \times N \times D$. The corresponding encoding rules $\text{Enc_rule}(i)$ are given as

$$\begin{cases} \text{Enc_rule}(i) = \text{mod}(u(i) + v(i), 7) + 1 \\ \text{Dec_rule}(i) = \text{mod}(x(i) - y(i), 7) + 1 \end{cases} \quad (22)$$

Step 3: Reshape the vector $Y_{2\text{RNA}}$ into a matrix of dimensions $L \times 4$ as follows:

$$Y_{2\text{RNA}} = \text{reshape}(Y_{2\text{RNA}}, L, 4), \quad (23)$$

with $L = \frac{M \times N \times D}{4}$.

Step 4: Split $Y_{2\text{RNA}}$ into two parts denoted FP and SP as follows:

$$\begin{aligned} FP &= X_{en4\text{RNA}}(:, 1 : 3) \\ SP &= X_{en4\text{RNA}}(:, 4), \end{aligned} \quad (24)$$

Step 5: Apply the RNA operation based on the customized Josephus problem, as detailed in A.6, to FP to produce FP_{RNA} . The matrices for step sizes (C_V) and directions (Dr) are generated using chaotic sequences, as depicted in Eq. (25):

$$\begin{cases} C_V(i : 64) = \text{mod}((i + j) \times v(i), 63) + 1 \\ Dr(i : 64) = \text{dec2bin}(w(i), 2, 64), \end{cases} \quad (25)$$

with $i = 1, 2, 3, \dots, L$, $j = 1, 2, 3, \dots, 64$, and dec2bin representing a function that converts decimal numbers into their binary equivalents.

Step 6: Depending on whether $x(i) \leq y(i)$ or not, the RNA operation ex_or or ex_nor is performed between the samples of $SP(i)$ and those of $K_{c1\text{RNA}}(i)$ to produce $SP_{\text{RNA}}(i)$;

Step 7: Concatenate FP_{RNA} with SP_{RNA} horizontally and convert it into a one-dimensional vector $X_{5\text{RNA}}$;

Step 8: Decode $X_{5\text{RNA}}$ according to the rule specified in sequence $\text{Dec_rule}(i)$, resulting in X_5 [refer to Eq. (26)], which is subsequently concatenated into a one-dimensional binary vector:

$$X_5(i) = [X_{5\text{RNA}}(i), \text{Dec_rule}(i)], \quad (26)$$

and then converted into decimal numbers as follows:

$$X_6(i) = \text{base2dec}(X_5(k : 8 \times i), 2), \quad (27)$$

with $k = 1, 9, 17, \dots, M \times N \times D/8$, $i = 1, 2, 3, \dots, M \times N \times D$.

Step 9: Transform Cb_4 and Cr_4 into 1-D vectors, denoted as Cb_5 and Cr_5 , respectively. Apply the CBC mode to derive Cb_6 and Cr_6 as follows:

$$\begin{cases} Cb_6(i + 1) = [(Cb_5(i + 1) \oplus Cb_6(i)) \oplus K_{c1}(i + 1)] \\ Cr_6(i + 1) = [(Cr_5(i + 1) \oplus Cr_6(i)) \oplus K_{c2}(i + 1)] \\ i = 1, 2, 3, \dots, M \times N \times D - 1, \end{cases} \quad (28)$$

where the initialization vectors of the CBC modes are given as

$$\begin{cases} Cb_6(1) = \text{index}(\max(K_c 1(16:240))) \\ Cr_6(1) = \text{index}(\max(K_c 2(16:240))) \end{cases} \quad (29)$$

Step 10: Reshape X_6 , Cb_6 , and Cr_6 to generate Y_{en} , Cb_{en} , and Cr_{en} as follows:

$$\begin{cases} Y_{en} = \text{reshape}(X_6, [M, N, 1, D]) \\ Cb_{en} = \text{reshape}(Cb_6, [M, N, 1, D]) \\ Cr_{en} = \text{reshape}(Cr_6, [M, N, 1, D]) \end{cases} \quad (30)$$

Step 11: The final encrypted video V_E is given as

$$\begin{cases} V_E(:, :, 1, :) = Y_{en} \\ V_E(:, :, 2, :) = Cb_{en} \\ V_E(:, :, 3, :) = Cr_{en} \end{cases} \quad (31)$$

C. Proposed Two-way VLC System

Figure 10 depicts the free-space communication system where the proposed encryption scheme is implemented and evaluated [21].

The figure depicts two nodes (laptops) equipped with a VLC application for encryption and decryption. Each node connects to a microcontroller unit (MCU) via a universal serial bus (USB) cable as described in subsections 2.A and 2.B. Each MCU controls an LED and a photodiode (PD) for light signal transmission and reception, respectively. A pair consisting of an LED and a PD connected to an MCU forms a VLC modulator/demodulator (VLC modem). Encrypted data can be transmitted bidirectionally between node 1 and node 2, as indicated by the double arrow. For example, when encrypted data flows from node 1 to node 2, data from node 1 are sent to its VLC modem. The LED of this modem transmits light to the PD of the second VLC modem via an air-based VLC channel. Node 2 receives and processes these encrypted data upon reception.

1. Transmission Process

The VLC application supports video files with pixel values encoded within 8-bit levels, ranging from 0 to 255. Therefore, each encrypted pixel requires 8 bits (1 byte) for encoding. Post encoding, the pixels are framed using universal asynchronous receiver transmitter (UART) frames. An UART frame structure comprises a start bit for synchronization, followed by 8 data bits. Additionally, 1 or 2 parity bits are included to avoid transmission errors, and 1 to 2 stop bits mark the end of the frame. Each frame represents the transmission of a single encrypted pixel. At the VLC modem, the continuous bit sequence is conveyed through unipolar on-off keying non-return-to-zero (OOK-NRZ) modulated electrical pulses. In OOK-NRZ modulation, the carrier's amplitude changes between two states, corresponding to binary information. These modulated pulses drive the emitting antenna, a current-driven unidirectional conducting device, where their brightness correlates directly with the forward current.

2. Reception Process

The light pulses emitted by the LED cross the air-based VLC channel and are received by a PD operating in switching mode. Let $P_t(t)$ denote the instantaneous optical power transmitted by the LED. The modulated light emitted by the LED travels through the air-based VLC channel and is intercepted by the receiving antenna of the opposite node, which also operates in switching mode. Assuming the emission pattern of the LED is Lambertian, the instantaneous optical power received by the antenna is described by Eq. (32) [41]:

$$P_r(t) = \begin{cases} \frac{A_{phy}}{d^2} \frac{(m+1)}{2\pi} \cos^m \phi \times \cos \psi \times P_t(t) & 0 \leq \psi \leq \text{FOV} \\ 0 & \text{otherwise,} \end{cases} \quad (32)$$

where A_{phy} represents the physical area of the receiver, ψ denotes the angle of incidence of received radiation relative to the axis of the receiving antenna, m signifies Lambert's mode number of the emitting antenna, d denotes the distance between the

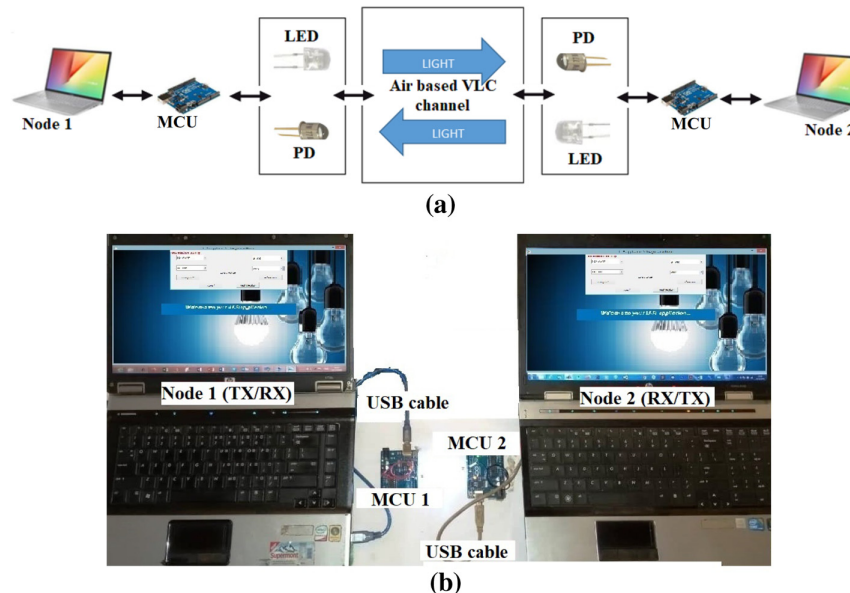


Fig. 10. Proposed two-way VLC system.

two transceivers, ϕ represents the angle between the axis of the emitter and the emitted light beam, and FOV denotes the field-of-view of the receiving antenna. Moreover, the receiving antenna functions as a square-law optoelectronic transducer, producing an electrical current ($I_P(t)$) that is directly proportional to the square of the instantaneous optical field incident on its surface [41]. Therefore, the photocurrent $I_P(t)$ generated is proportional to $P_r(t)$, expressed as follows [42–44]:

$$I_P(t) = \eta_{qe} \frac{q P_r(t)}{hf} (1 - e^{-\alpha l}). \quad (33)$$

In the equation, hf represents the photon energy, l denotes the junction width, q stands for the elementary charge, α represents the absorption coefficient, and η_{qe} denotes the quantum efficiency. In our system, the generated photocurrent $I_P(t)$ flows through a resistor R , resulting in a voltage drop ($V(t) = R \times I_P(t)$). This voltage $V(t)$ constitutes a unipolar OOK-NRZ signal that represents the received UART data frames. Subsequently, all UART data frames are transmitted from the VLC modem to the VLC application via the USB cable. Following this, the VLC application proceeds to process the received encrypted data frames to decrypt the data and reconstruct the original video file.

3. EXPERIMENTAL RESULTS AND COMPARATIVE ANALYSIS

This section presents the experimental results of our encryption algorithm, followed by a detailed comparative analysis to evaluate its performance and effectiveness against existing methods.

A. Experimental Results

This subsection presents the experimental results and performance analysis of the proposed encryption scheme. A robust video encryption method must withstand various attacks to ensure the security of videos during transmission or storage. We employ diverse security tests to assess the efficacy of the scheme. Importantly, the performance analysis utilizes the VLC infrastructure, with most evaluations conducted post-transmission and reception of video files. The test videos used include xylophone [240,320,3,141], traffic [120,160,3,120], rhinos [240,320,3,328], flamingo [192,352,3,328], and train [192,352,3,262].

1. Visual Inspection

Visual inspection serves as the initial metric for evaluating the proposed video encryption scheme. Figure 11 illustrates the representations of plain, encrypted, and decrypted videos. It is evident from the figure that the encryption scheme effectively conceals all visual details in the encrypted videos. Moreover, the decryption process at the receiver's end successfully restores the plain videos without observable degradation or artifacts.

2. Statistical Analysis

a. Information entropy measure

In video cryptography, assessing frame randomness involves measuring the entropy of both plain and encrypted

video frames. An optimal entropy value of 8 signifies robust security in encrypted frames. Entropy (E) is calculated as [21,45]:

$$E = - \sum_{k=0}^L P(k) \log_2 P(k). \quad (34)$$

We evaluated our proposed encryption scheme using every 15th frame of each video and recorded the results in Table 4. The table highlights that the entropy values of many encrypted video frames approximate 8, indicating the scheme's effectiveness in mitigating entropy attacks.

b. Correlation coefficient

The correlation coefficient analysis of a frame quantifies the degree of similarity between pixels at different spatial positions. This coefficient, ranging numerically from -1 to 1 , signifies the strength and direction of linear relationships between pixel intensities. A coefficient of 1 denotes a perfect positive correlation, 0 indicates no linear correlation, and -1 signifies a perfect negative correlation. The formula used to compute the correlation coefficient is as follows [21,45]:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad (35)$$

with $\text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2$, $D(y) = \frac{1}{N} \sum_{i=1}^N (y_i - E(y))^2$, $E(x) = \frac{1}{N} \sum_{i=1}^N x_i$, and $E(y) = \frac{1}{N} \sum_{i=1}^N y_i$ here. x and y denote two adjacent pixels, and N denotes the number of selected pixels. We chose $N = 3000$, meaning 3000 pairs of adjacent pixels from the 15th frame were randomly selected. The results are displayed in Table 5.

A decrease in the correlation coefficients of encrypted video frames demonstrates the efficacy of the proposed encryption scheme in minimizing correlations between neighboring pixels. Figure 12 illustrates the distribution diagram of adjacent pixel correlations in the 15th frame of both the plain and encrypted xylophone videos. The correlation distribution plots of adjacent pixels in plain-text frames exhibit strong correlation, whereas those in encrypted frames appear uniform and non-correlated. This observation underscores the effectiveness of our encryption scheme in significantly reducing pixel correlation within video frames, thereby enhancing the robustness and reliability of the encryption process.

c. Histogram analysis

Histogram analysis serves as a fundamental tool for evaluating the efficacy of an encryption scheme by depicting the frequency distribution of pixel values within a frame [46]. Figure 13 illustrates the histograms of the 15th plain and encrypted frames of the xylophone video.

One can observe that the distribution of pixels in the encrypted frame is notably uniform and flat.

d. Chi-square test analysis and deviation from ideality

The flatness of the histogram can be assessed using the chi-square test [47]. Typically, histogram uniformity is confirmed if the computed chi-square value of the video is below 293.2478. The chi-square value is calculated using the formula

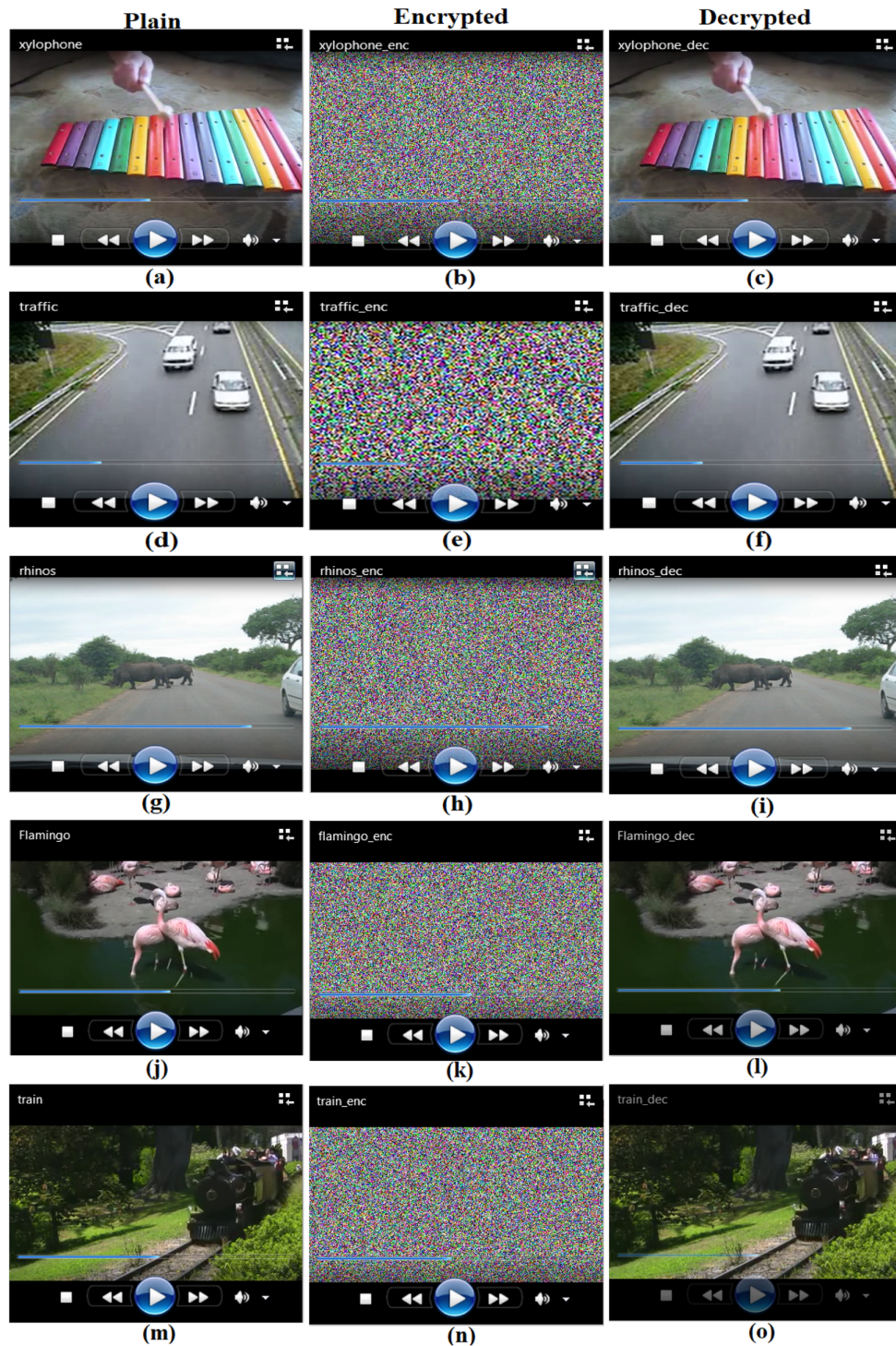


Fig. 11. Visual representation of video encryption and decryption process: (a, d, g, j, m) show plain videos; (b, e, h, k, n) show corresponding encrypted videos; (c, f, i, l, o) show corresponding decrypted videos.

$$\chi^2 = \sum_{L=0}^{255} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}, \quad (36)$$

where $\text{expected} = \frac{M \times N}{256}$. Also, L and $M \times N$ denote the gray level values and the frame size, respectively. Regarding Table 6, the chi-square values are below 293.2478, indicating that the histograms of all encrypted

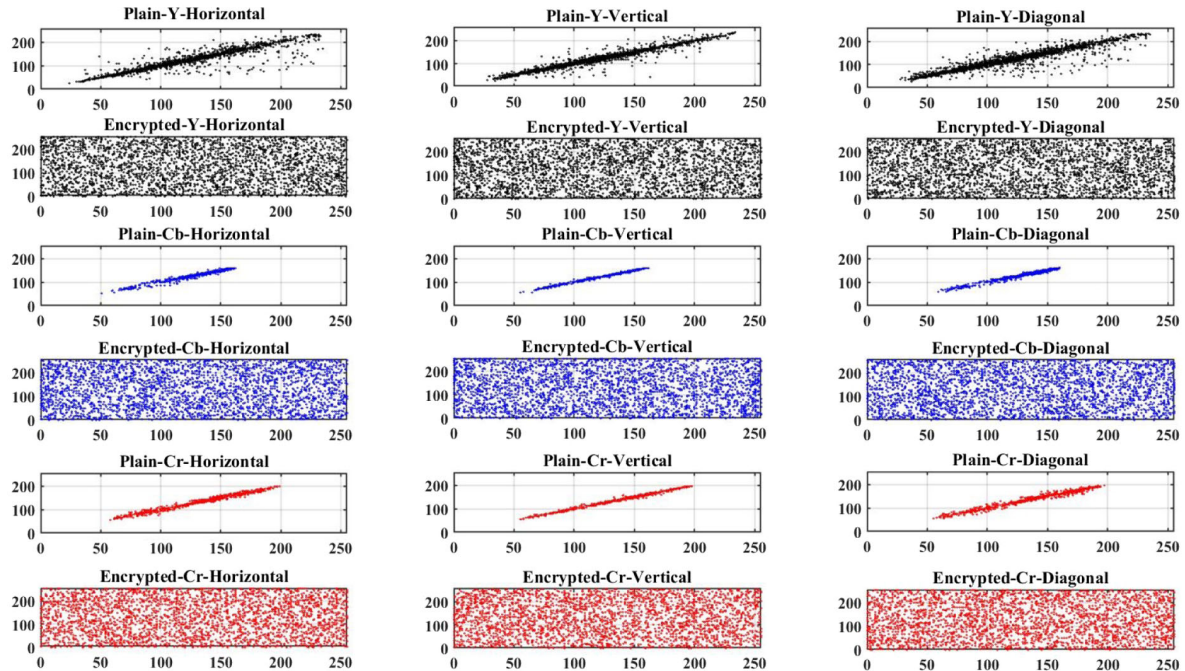
frames exhibit uniform distribution. Additionally, the small deviations between the obtained encryption histograms and the ideal encryption case serve as metrics for evaluating resistance against statistical attacks. Table 6 demonstrates minimal deviations, affirming the nearly ideal performance of the proposed encryption scheme.

Table 4. Entropy Values of Plain and Encrypted Video Frames

Videos	Plain Video			Encrypted Video		
	Y	Cb	Cr	Y	Cb	Cr
Xylophone	7.3649	4.8281	5.1552	7.9972	7.9975	7.9977
Traffic	6.8188	3.9244	3.0077	7.9899	7.9893	7.9909
Rhinos	6.6764	3.7569	3.3006	7.9976	7.9974	7.9976
Flamingo	5.3131	3.6595	3.2885	7.9975	7.9971	7.9976
Train	6.8318	5.3640	3.7274	7.9971	7.9969	7.9972

Table 5. Correlation Coefficients of Plain and Encrypted Video Frame

Videos	Directions	Plain Video Frame			Encrypted Video Frame		
		Y	Cb	Cr	Y	Cb	Cr
Xylophone	H	0.9576	0.9865	0.9890	−0.0134	0.0302	0.0207
	V	0.9774	0.9946	0.9945	−0.0114	−0.0624	0.0114
	D	0.9437	0.9869	0.9877	0.0011	−0.0095	−0.0038
Traffic	H	0.8135	0.9442	0.8975	0.0081	0.0189	−0.0273
	V	0.8398	0.9608	0.9438	−0.0151	−0.0093	0.0198
	D	0.7934	0.9064	0.8519	−0.0403	−0.0042	−0.0083
Rhinos	H	0.9826	0.9760	0.9849	0.0087	−0.1377	−0.0099
	V	0.9776	0.9680	0.9715	0.0286	−0.0026	0.0242
	D	0.9635	0.9456	0.9599	0.0022	−0.0079	0.0042
Flamingo	H	0.9730	0.9948	0.9925	0.0068	−0.0018	0.0028
	V	0.9647	0.9892	0.9798	−0.0186	−0.0047	0.0139
	D	0.9515	0.9856	0.9701	−0.0061	0.0218	−0.0133
Train	H	0.9378	0.9892	0.9899	0.0132	−0.0538	−0.0002
	V	0.9233	0.9728	0.9735	−0.0063	−0.0023	0.0020
	D	0.8854	0.9644	0.9627	0.0181	−0.0319	−0.0086

**Fig. 12.** Correlation distribution diagrams of adjacent pixels of plain and encrypted 15th frame of xylophone.

3. Differential Analysis

Cryptographic attackers leverage differential attacks to potentially recover secret keys by analyzing the impact of slight differences between input pairs on corresponding output pairs.

To mitigate this, video encryption schemes must exhibit sensitivity to even minimal changes in plaintext. Evaluating the scheme's resilience against differential attacks involves assessing key metrics such as the number of pixel change rate (NPCR)

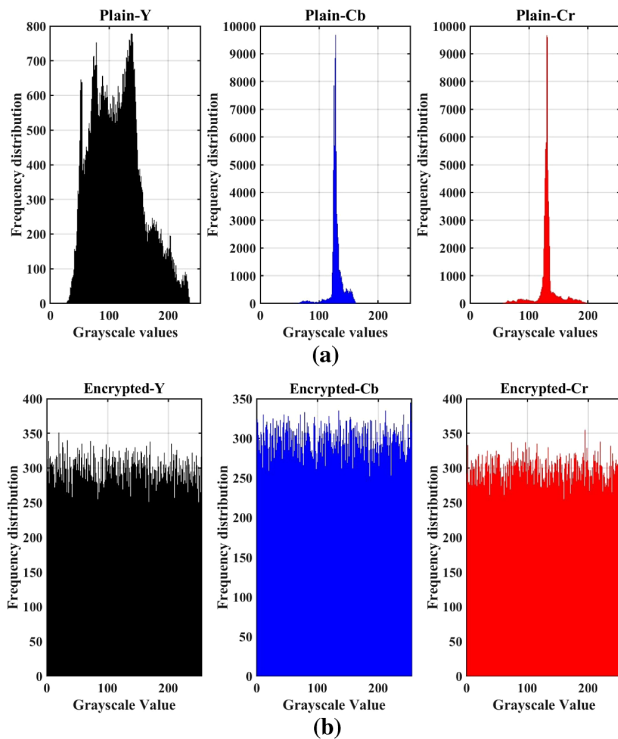


Fig. 13. (a) Histogram of the plain video frame and (b) histogram of the corresponding encrypted frame.

and unified average changing intensity (UACI), calculated as follows [45]:

$$\text{NPCR} = \sum_{i=1}^N D_i \times 100\% \quad (37)$$

$$\text{UACI} = \frac{1}{N} \left[\sum_{i=1}^N \frac{|Y1_i - Y2_i|}{2^M} \right] \times 100\%,$$

where

$$D_i = \begin{cases} 1, & Y1_i \neq Y2_i \\ 0, & \text{otherwise} \end{cases} \quad (38)$$

In Eqs. (37) and (38), $Y1_i$ and $Y2_i$ represent the pixel values of the two encrypted video frames, where NN denotes the total number of pixels. The expected NPCR is 99.6094%, and the expected UACI is 33.4635% [45]. In our experiment, we assessed NPCR and UACI by randomly modifying a single pixel in the plain video. Table 7 summarizes the test outcomes.

The results indicate that the proposed scheme successfully passes the NPCR and UACI tests for the majority of the videos, demonstrating its robust resistance against differential attacks.

4. Peak Signal-to-Noise Ratio

The peak signal-to-noise ratio (PSNR) serves as a metric to quantify the disparity between plain and encrypted frames. A diminished PSNR value correlates with increased noise levels, consequently reflecting heightened degradation in the encrypted frame. The PSNR is computed using the formula [45]:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right), \quad (39)$$

Table 6. Chi-square Test Values and Deviation of Histogram from Ideality

Videos		Chi-square Value	Deviation
Xylophone	Y	292.7400	0.0503
	Cb	263.8400	0.0488
	Cr	253.2610	0.0454
Traffic	Y	267.7067	0.0961
	Cb	281.1467	0.0968
	Cr	242.6400	0.0905
Rhinos	Y	252.6867	0.0461
	Cb	276.7533	0.0498
	Cr	250.9533	0.0439
Flamingo	Y	232.9091	0.0484
	Cb	271.6364	0.0511
	Cr	226.2273	0.0458
Train	Y	267.0379	0.0520
	Cb	291.5758	0.0522
	Cr	260.7652	0.0507

where MAX denotes the maximum possible pixel value, and MSE is calculated as the mean square error according to the formula $\text{MSE} = \frac{1}{N} \sum_{i=1}^N (X_i - Y_i)^2$. As observed in Table 8, the PSNR values indicate significant degradation, indicating that the encrypted videos exhibit substantial noise levels.

5. Structural Similarity Index

The structural similarity index (SSIM) serves as a metric quantifying the degradation level observed in images post-encryption. SSIM is employed here to evaluate degradation between plain and encrypted videos, as well as between plain and decrypted frames. Low SSIM values are desirable between plain and encrypted frames, ideally approaching zero, while SSIM values close to 1 are expected between plain and decrypted frames to validate the efficacy of encryption and decryption operations. SSIM is computed using the formula given in Eq. (40), and the outcomes are documented in Table 9:

$$\text{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (40)$$

where μ_x denotes the pixel sample mean of x , μ_y denotes the pixel sample mean of y , σ_x^2 denotes the variance of x , σ_y^2 denotes the variance of y , and σ_{xy} denotes the covariance between x and y . Additionally, we introduce c_1 and c_2 as variables employed to stabilize division operations involving potentially weak denominators. The table demonstrates that the SSIM values between the plaintext and encrypted frames exhibit a significant reduction, whereas the SSIM values between the plaintext and decrypted frames consistently reach 1 across all tests. This unequivocally validates the efficacy of both the encryption and decryption procedures.

6. Occlusion Attacks Analysis

The examination of occlusion attacks holds significant importance in assessing the resilience of a cryptosystem. These attacks manifest when segments of encrypted video data are inadvertently lost during transmission [48]. A robust encryption

Table 7. NPCR and UACI Values

Videos	NPCR			UACI		
	Y	Cb	Cr	Y	Cb	Cr
Xylophone	99.5768%	99.6120%	99.6393%	33.4052%	33.5872%	33.5515%
Traffic	99.5677%	99.6406%	99.6094%	33.4781%	33.3446%	33.5737%
Rhinos	99.6211%	99.6029%	99.5829%	33.4693%	33.5643%	33.3643%
Flamingo	99.6168%	99.6375%	99.6138%	33.5454%	33.3969%	33.4575%
Train	99.5931%	99.6345%	99.5650%	33.5230%	33.4461%	33.5529%

Table 8. PSNR Values

Filenames	PSNR (dB)
Xylophone	10.1294
Traffic	10.416
Rhinos	10.1582
Flamingo	9.52
Train	9.52

scheme must demonstrate efficacy against such adversities. Figure 14 depicts the experimental outcomes of our proposed scheme under occlusion attacks. Despite data loss, the decrypted videos remain discernible, underscoring the algorithm's robust resistance to occlusion attacks.

7. Noise Attacks Analysis

Another critical criterion for evaluating the efficacy of a video encryption scheme is its resilience against noise attacks [49]. To assess the robustness of our algorithm, we introduced varying levels of salt-and-pepper and Gaussian noise to several encrypted videos. As illustrated in Fig. 15, the results demonstrate that our scheme successfully reconstructs legible videos from corrupted encrypted counterparts.

Table 9. SSIM Values

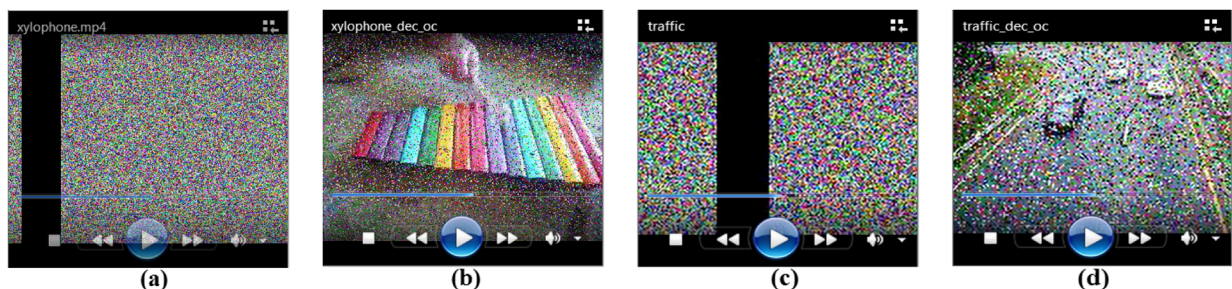
Videos	Plain and Encrypted			Plain and Decrypted		
	Y	Cb	Cr	Y	Cb	Cr
Xylophone	0.0095	0.0116	0.0113	1	1	1
Traffic	0.0113	0.0117	0.0113	1	1	1
Rhinos	0.0116	0.0112	0.0113	1	1	1
Flamingo	0.0081	0.0113	0.0113	1	1	1
Train	0.0098	0.0113	0.0113	1	1	1

8. Key Space Analysis

The security of an encryption scheme hinges on several critical factors, among which the key space plays a pivotal role. This parameter denotes the product of the lengths of all key parameters involved. To resist brute-force attacks effectively, a robust encryption scheme necessitates a key space exceeding 2^{100} [50]. The encryption scheme under consideration employs key parameters comprising two 512-bit values, γ_1 and γ_2 , derived from the SHA3-512 function, each possessing a length of 2^{512} . Additionally, the system parameters collectively span a length of 10^{144} . Consequently, the proposed encryption scheme boasts a total key space of $2^{1024} \times 10^{144}$, significantly surpassing the specified threshold. This expansive key space ensures formidable resilience against brute-force attacks.

9. Key Sensitivity Analysis

The key sensitivity analysis pertains to the distinction between two encrypted videos derived from the same source video but encrypted with marginally different cryptographic keys. An effective encryption system is characterized by heightened key sensitivity. Employing the "xylophone" video as a case study illustrates that even a minor alteration in the initial encryption key results in erroneous decryption, as depicted in Fig. 16.

**Fig. 14.** Experimental results under occlusion attacks: attacked encrypted videos (a), (c) and corresponding recovered videos (b), (d).

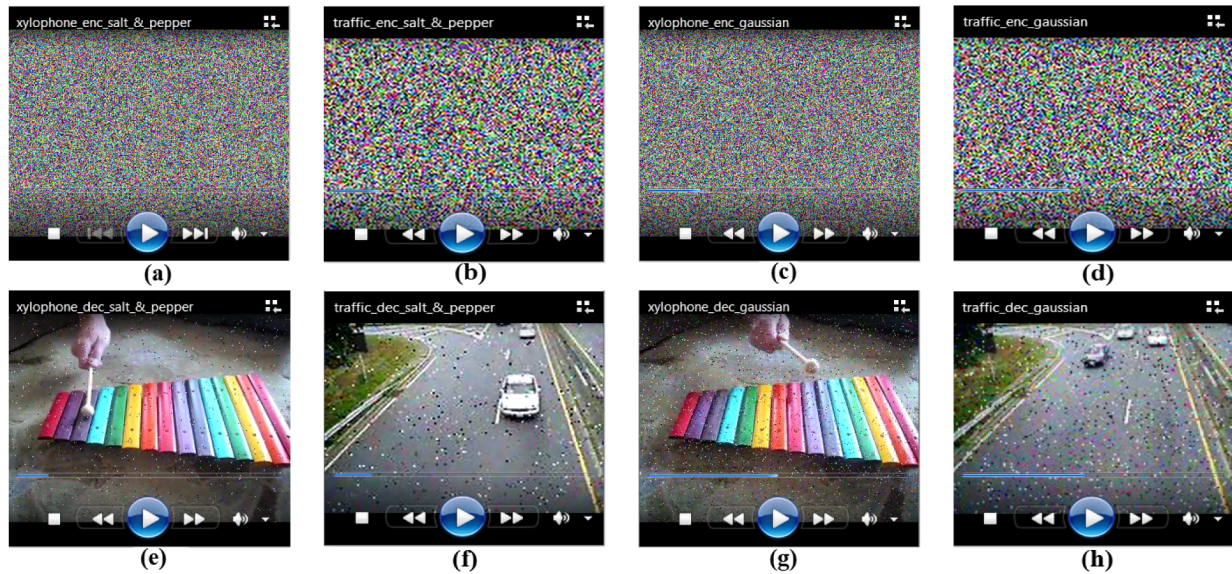


Fig. 15. Effects of noise attacks on output results. (a) and (b) depict videos attacked with salt-and-pepper noise at intensities of 0.01 and 0.05, respectively, with corresponding decrypted versions shown in (e) and (f). (c) and (d) display videos attacked with Gaussian noise at intensities of 0.001 and 0.005, respectively, with corresponding decrypted versions shown in (g) and (h).

Table 10. Speed for Encryption Process

Videos	Size	Time (s)
Xylophone	$240 \times 320 \times 3 \times 141$	82.021
Traffic	$120 \times 160 \times 3 \times 140$	21.250
Rhinos	$240 \times 320 \times 3 \times 328$	205.523
Flamingo	$192 \times 352 \times 3 \times 328$	195.580
Train	$292 \times 352 \times 3 \times 262$	157.825

10. Known-plaintext and Chosen-plaintext Attacks

Cryptanalysis endeavors to uncover the obscured key or its components to decrypt encoded messages. The literature categorizes attacks into four types: ciphertext-only, known-plaintext, chosen-plaintext, and chosen-ciphertext attacks. Known-plaintext and chosen-plaintext attacks are particularly formidable. The secret key of the proposed encryption scheme intricately synchronizes with both the input video signal and computer timestamp using the SHA3-512 hash function. This synchronization significantly bolsters the system's resilience against known-plaintext and chosen-plaintext attacks [51].

11. Time Complexity Analysis

The temporal complexity of an encryption scheme stands as pivotal, particularly in applications involving real-time communication [52]. Accordingly, the proposed video encryption scheme exhibits a linear complexity denoted by $O(M \times N \times D)$. This substantiates that our encryption algorithm maintains a commendable temporal efficiency, rendering it well-suited for computer-based real-time communication applications.

12. Speed for Encryption Process

To measure the time required to encrypt, we used videos of different sizes with hardware configuration 3.4 GHz, Intel Core i76600U, 16 GB RAM, and Windows 11. Table 10 shows that the proposed scheme has a satisfying encryption speed.

B. Comparative Analysis

Numerous video encryption schemes have been documented in the scientific literature. To substantiate the efficacy of our scheme, we conducted a comparative performance analysis

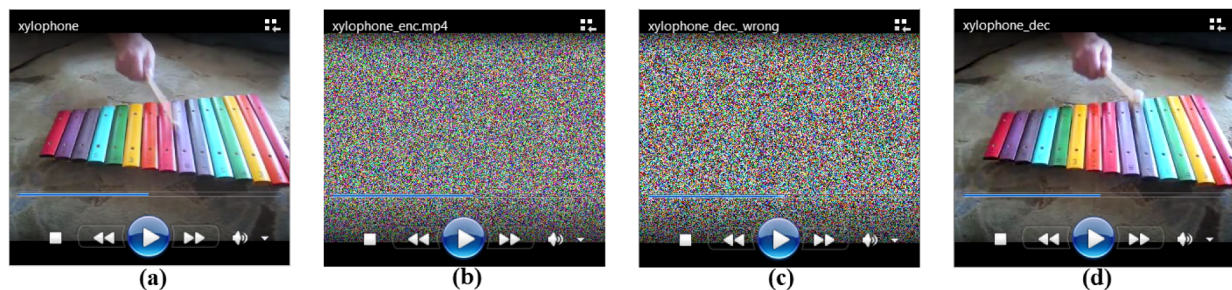


Fig. 16. Encryption and decryption key sensitivity assessment: (a) plaintext xylophone; (b) encrypted xylophone using the initial key; (c) decrypted xylophone with an incorrect key; and (d) decrypted xylophone with the correct key.

Table 11. Outcome of Comparative Analysis^a

References	Entropy	Correlation Coefficients			NPCR	UACI	Key Space	Time Complexity	Experimental Implementation
		H	V	D					
[4]	7.9977	—	—	—	99.6544	33.4134	2 ³⁸⁴	—	no
[5]	7.8729	—	—	—	—	—	—	—	no
[6]	—	—	—	—	99.61	33.61	—	—	no
[7]	—	0.0176	0.0147	0.0158	99.95	33.5990	49152 × 2 ¹²⁸	—	no
[8]	—	—	—	—	98.9897	33.9845	230 × 10 ⁶³	—	no
[9]	7.598	0.4529	0.5928	0.4040	—	—	—	—	no
[10]	7.8708	-0.0245	-0.0135	-0.143	—	—	32738 ⁶	$O(2^{m+2n+2}n^2)$	no
[11]	7.9998	—	—	—	99.73	33.68	—	—	no
[12]	7.99	-0.0099	-0.0043	0.0116	99.6	33.47	—	$O(M \times N \times K)$	no
[15]	7.9899	—	—	—	—	—	—	—	no
[16]	7.9978	—	—	—	99.60	33.462	—	$O(M \times N)$	no
[17]	7.9981	0.001548	-0.005947	0.01148	99.62	30.68	—	—	no
This work	7.9974	-0.0134	-0.0624	-0.0403	99.6406	33.5872	2¹⁰²⁴ × 10¹⁴⁴	$O(M \times N \times D)$	yes

^aBold fonts denote the best results.

against established methodologies. Detailed findings from this comparative study are documented in Table 11.

Our encryption scheme exhibits high efficiency in terms of information entropy, surpassing those detailed in Refs. [5,9,10,12,15]. It marginally deviates from the results reported in Refs. [4,11,16,17]. A primary advantage of our approach is its capability to generate encrypted videos with exceptionally low correlation coefficients. Evaluation against metrics such as NPCR and UACI consistently demonstrates superior performance compared to recent advancements. However, our achieved NPCR and UACI values slightly trail those reported in Refs. [4,7] for NPCR and Refs. [6,8,11] for UACI. Significantly, our encryption scheme boasts a larger key space and heightened key sensitivity relative to existing literature. Furthermore, our system effectively recovers encrypted videos post-decryption. The time complexity analysis highlights that our proposed encryption scheme exhibits linear time complexity $O(M \times N \times D)$, which is competitive compared to other works in the table. Notably, unlike previous approaches lacking practical implementation, our encryption scheme has been successfully integrated into a VLC system, showcasing its efficacy in securing video transmissions over optical wireless channels.

4. CONCLUSION

This paper has introduced an advanced video encryption scheme that leverages a modified chaotic Colpitts oscillator, a bespoke Josephus problem, and RNA operations, implemented and evaluated within a VLC framework. The modified Colpitts oscillator was meticulously tuned to optimize its performance, emphasizing the exponential model of the transistor's base-emitter junction. The encryption process commenced with RGB to YCbCr conversion, followed by pixel-level permutation of the Y channel and block scrambling of the Cb and Cr channels using two distinct schemes. The initial scheme utilized a customized Josephus problem, while the second integrated column- and row-shifting techniques. For diffusion, a custom Josephus problem-based RNA operation, in conjunction with fundamental RNA operations and CBC mode, was

employed. Synchronization of the modified Colpitts oscillator with plaintext video and computer timestamps was achieved using the SHA-3-512 hash function, ensuring heightened unpredictability. Additionally, the RSA public key encryption method safeguarded the key parameters of the proposed encryption scheme, ensuring robust asymmetric encryption. Our results have demonstrated that the column- and row-shifting permutation scheme enhanced efficiency and augmented permutation complexity, which are crucial for robust video encryption. Moreover, the customized Josephus problem-based permutation improves granularity, which is essential for robust encryption schemes. The customized Josephus-based RNA operation streamlines the diffusion process, enhancing computational efficiency. Theoretical analysis and experimental validation within a VLC framework affirm the efficacy and practicality of the proposed video encryption scheme for securing video data transmitted via optical wireless channels. This scheme shows promise for applications requiring robust video data security such as surveillance, medical imaging, and defense communications. Future research aims to explore further optimizations.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results of this paper may be obtained from the authors upon reasonable request.

REFERENCES

1. L. D. Singh and K. M. Singh, "Image encryption using elliptic curve cryptography," *Proc. Comput. Sci.* **54**, 472–481 (2015).
2. L. D. Singh and K. M. Singh, "Implementation of text encryption using elliptic curve cryptography," *Proc. Comput. Sci.* **54**, 73–82 (2015).
3. J. Ahmad and F. Ahmed, "Efficiency analysis and security evaluation of image encryption schemes," *Computing* **23**, 25 (2010).
4. S. A. Abaas and A. K. Shibebe, "A new approach for video encryption based on modified AES algorithm," *IOSR J. Comput. Eng.* **17**, 44–51 (2015).
5. P. Aggarwal and S. Vishwanath, "Design and implementation of video encryption for multimedia applications," *J. Eng. Res. Appl.* **4**, 29–34 (2014).

6. I. Yasser, M. A. Mohamed, A. S. Samra, *et al.*, "A chaotic-based encryption/decryption framework for secure multimedia communications," *Entropy* **22**, 1253 (2020).
7. D. Valli and K. Ganesan, "Chaos based video encryption using maps and Ikeda time delay system," *Eur. Phys. J. Plus* **132**, 1–18 (2017).
8. H. Elkamchouchi, W. M. Salama, and Y. Abouelseoud, "New video encryption schemes based on chaotic maps," *IET Image Process.* **14**, 397–406 (2020).
9. R. S. Malladar and S. R. Kunte, "Selective video encryption using the cross coupling of one-dimensional logistic maps," *Int. J. Comput. Netw. Inf. Secur.* **13**, 40–54 (2021).
10. X. H. Song, H. Q. Wang, S. E. Venegas-Andraca, *et al.*, "Quantum video encryption based on qubit-planes controlled-XOR operations and improved logistic map," *Physica A* **537**, 122660 (2020).
11. A. T. Maolood, E. K. Gbashi, and E. S. Mahmood, "Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map," *Int. J. Electr. Comput. Eng.* **12**, 4988–5000 (2022).
12. K. M. Hosny, M. A. Zaki, N. A. Lashin, *et al.*, "Fast colored video encryption using block scrambling and multi-key generation," *Vis. Comput.* **39**, 6041–6072 (2022).
13. J. Li, C. Wang, X. Chen, *et al.*, "A selective encryption scheme of CABAC based on video context in high efficiency video coding," *Multimedia Tools Appl.* **77**, 12837–12851 (2018).
14. J. Yun and M. Kim, "JLVEA: lightweight real-time video stream encryption algorithm for internet of things," *Sensors* **20**, 3627 (2020).
15. S. Cheng, L. Wang, N. Ao, *et al.*, "A selective video encryption scheme based on coding characteristics," *Symmetry* **12**, 332 (2020).
16. B. M. El-den, W. A. Raslan, and A. A. Abdullah, "Even symmetric chaotic and skewed maps as a technique in video encryption," *EURASIP J. Adv. Signal Process.* **2023**, 40 (2023).
17. M. Heshmat, "Video encryption technique based on hybrid chaotic maps and multi-operation keys," *Inf. Sci. Lett.* **11** 2197–2207 (2022).
18. K. M. Hosny, M. A. Zaki, H. M. Hamza, *et al.*, "Privacy protection in surveillance videos using block scrambling-based encryption and DCNN-based face detection," *IEEE Access* **10**, 106750 (2022).
19. A. Hafsa, M. Fradi, A. Sghaier, *et al.*, "Real-time video security system using chaos-improved advanced encryption standard (IAES)," *Multimedia Tools Appl.* **81**, 2275–2298 (2022).
20. X. Li, H. Yu, H. Zhang, *et al.*, "Video encryption based on hyperchaotic system," *Multimedia Tools Appl.* **79**, 23995–24011 (2020).
21. S. B. N. Fotso, W. N. Atchoffo, A. C. Nzeukou, *et al.*, "Enhanced security in lossless audio encryption using zigzag scrambling, DNA coding, SHA-256, and Hopfield networks: a practical VLC system implementation," *Multimedia Tools Appl.* 1–35 (2024).
22. M. A. Khan and M. A. AlGhamdi, "An intelligent and fast system for detection of grape diseases in RGB, grayscale, YCbCr, HSV, and L*a*b* color spaces," *Multimedia Tools Appl.* **83**, 50381–50399 (2024).
23. H. Yoshida and A. Biryukov, "Analysis of a SHA-256 variant," in *International Workshop on Selected Areas in Cryptography*, Springer, (2005), pp. 245–260.
24. B. S. Rawal, L. S. Kumar, S. Maganti, *et al.*, "Comparative study of Sha-256 optimization techniques," in *IEEE World AI IoT Congress (AllIoT)* (2022), 387–392.
25. S. Haber and H. Massias, "Time-stamping," in *Encyclopedia of Cryptography, Security and Privacy* (Springer, 2024), pp. 1–6.
26. L. O. Chua, C. W. Wu, A. Huang, *et al.*, "A universal circuit for studying and generating chaos. I. Routes to chaos," *IEEE Trans. Circuits Syst. I* **40**, 732–744 (1993).
27. V. Kamdoun Tamba, H. B. Fotsin, J. Kengne, *et al.*, "Emergence of complex dynamical behaviors in improved Colpitts oscillators: anti-monotonicity, coexisting attractors, and metastable chaos," *Int. J. Dyn. Control* **5**, 395–406 (2017).
28. J. Kengne, J. C. Chedjou, V. A. Fono, *et al.*, "On the analysis of bipolar transistor based chaotic circuits: case of a two-stage Colpitts/Colpitts/Colpitts oscillator," *Nonlinear Dyn.* **67**, 1247–1260 (2012).
29. S. T. Kingni, J. H. T. Mbé, and P. Wofo, "Semiconductor lasers driven by self-sustained chaotic electronic oscillators and applications to optical chaos cryptography," *Chaos* **22**, 033108 (2012).
30. R. M. Kouayep, A. F. Talla, J. H. T. Mbé, *et al.*, "Bursting oscillations in Colpitts oscillator and application in optoelectronics for the generation of complex optical signals," *Opt. Quantum Electron.* **52**, 1–13 (2020).
31. R. M. M. Kouayep, J. H. T. Mbé, and P. Wofo, "Power spectrum analysis of time-delayed optoelectronic oscillators with wide and narrow band nonlinear filters," *Opt. Quantum Electron.* **55**, 62 (2023).
32. W. N. Atchoffo, J. H. T. Mbé, P. Wofo, *et al.*, "Time-delay signature suppression in single-loop optoelectronic oscillators based on sine hyperbolic nonlinear amplifier," *Opt. Quantum Electron.* **55**, 355 (2023).
33. J. H. T. Mbé, M. C. Njijou, A. F. Talla, *et al.*, "Multistability, relaxation oscillations, and chaos in time-delayed optoelectronic oscillators with direct laser modulation," *Opt. Lett.* **49**, 1277–1280 (2024).
34. L. Wang, Y. Cao, H. Jahanshahi, *et al.*, "Color image encryption algorithm based on double-layer Josephus scramble and laser chaotic system," *Optik* **275**, 170590 (2023).
35. Z. Guan, J. Li, L. Huang, *et al.*, "A novel and fast encryption system based on improved Josephus scrambling and chaotic mapping," *Entropy* **24**, 384 (2022).
36. Q. Zhang, J. Han, and Y. Ye, "Image encryption algorithm based on image hashing, improved chaotic mapping and DNA coding," *IET Image Process.* **13**, 2905–2915 (2019).
37. S. Wang, Q. Peng, and B. Du, "Chaotic color image encryption based on 4D chaotic maps and DNA sequence," *Opt. Laser Technol.* **148**, 107753 (2022).
38. J. Yu, W. Xie, Z. Zhong, *et al.*, "Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation," *Chaos Solitons Fractals* **162**, 112456 (2022).
39. Q. Cun, X. Tong, Z. Wang, *et al.*, "A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing," *Vis. Comput.* **39**, 6589–6608 (2023).
40. R. K. Meenakshi and A. Arivazhagan, "RTL modelling for the cipher block chaining mode (CBC) for data security," *Indones. J. Electr. Eng. Comput. Sci.* **8**, 709–711 (2017).
41. A. R. Ndjongue, H. C. Ferreira, and T. M. Ngatched, "Visible light communications (VLC) technology," in *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999), pp. 1–15.
42. Z. Ghassemloo, W. Popoola, and S. Rajbhandari, *Optical wireless communications: System and channel modelling with Matlab* (CRC Press, 2019).
43. J. H. T. Mbé and P. Wofo, "Modulation of distributed feedback (DFB) laser diode with the autonomous Chua's circuit: theory and experiment," *Opt. Laser Technol.* **100**, 145–152 (2018).
44. J. H. T. Mbé, P. Wofo, and H. B. Fotsin, "Experimental direct modulation of a laser diode with a van der Pol circuit and applications," *Opt. Eng.* **58**, 066114 (2019).
45. L. Zhou, X. Li, F. Tan, *et al.*, "A two-layer networks-based audio encryption/decryption scheme via fixed-time cluster synchronization," *Soft Comput.* **26**, 9761–9774 (2022).
46. C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Opt. Commun.* **285**, 29–37 (2012).
47. H. Khanzadi, M. Eshghi, and S. E. Borujeni, "Image encryption using random bit sequence based on chaotic maps," *Arab. J. Sci. Eng.* **39**, 1039–1047 (2014).
48. P. Singh, A. K. Yadav, and K. Singh, "Phase image encryption in the fractional Hartley domain using Arnold transform and singular value decomposition," *Opt. Lasers Eng.* **91**, 187–195 (2017).
49. A. Shafique, J. Ahmed, M. U. Rehman, *et al.*, "Noise-resistant image encryption scheme for medical images in the chaos and wavelet domain," *IEEE Access* **9**, 59108–59130 (2021).
50. Z. W. Huang and N. R. Zhou, "Image encryption scheme based on discrete cosine Stockwell transform and DNA-level modulus diffusion," *Opt. Laser Technol.* **149**, 107879 (2022).
51. A. A. A. El-Latif, B. Abd-El-Atty, A. Belazi, *et al.*, "Efficient chaos-based substitution-box and its application to image encryption," *Electronics* **10**, 1392 (2021).
52. S. B. N. Fotso, J. H. T. Mbé, W. N. Atchoffo, *et al.*, "Chaotic image encryption scheme based on improved Z-order curve, modified Josephus problem, and RNA operations: an experimental Li-Fi approach," *Complexity* **2024**, 2910833 (2024).