



## Securing Transmitted Color Images Using Zero Watermarking and Advanced Encryption Standard on Raspberry Pi

Doaa Sami Khafaga<sup>1</sup>, Sarah M. Alhammad<sup>1,\*</sup>, Amal Magdi<sup>2</sup>, Osama ElKomy<sup>2</sup>, Nabil A. Lashin<sup>2</sup> and Khalid M. Hosny<sup>2</sup>

<sup>1</sup>Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia

<sup>2</sup>Department of Information Technology, Zagazig University, Zagazig, 44519, Egypt

\*Corresponding Author: Sarah M. Alhammad. Email: smalhammad@pnu.edu.sa

Received: 15 March 2023; Accepted: 16 May 2023; Published: 28 July 2023

**Abstract:** Image authentication techniques have recently received a lot of attention for protecting images against unauthorized access. Due to the wide use of the Internet nowadays, the need to ensure data integrity and authentication increases. Many techniques, such as watermarking and encryption, are used for securing images transmitted via the Internet. The majority of watermarking systems are PC-based, but they are not very portable. Hardware-based watermarking methods need to be developed to accommodate real-time applications and provide portability. This paper presents hybrid data security techniques using a zero watermarking method to provide copyright protection for the transmitted color images using multi-channel orthogonal Legendre Fourier moments of fractional orders (MFrLFMs) and the advanced encryption standard (AES) algorithm on a low-cost Raspberry Pi. In order to increase embedding robustness, the watermark picture is scrambled using the Arnold method. Zero watermarking is implemented on the Raspberry Pi to produce a real-time ownership verification key. Before sending the ownership verification key and the original image to the monitoring station, we can encrypt the transmitted data with AES for additional security and hide any viewable information. The receiver next verifies the received image's integrity to confirm its authenticity and that it has not been tampered with. We assessed the suggested algorithm's resistance to many attacks. The suggested algorithm provides a reasonable degree of robustness while still being perceptible. The proposed method provides improved bit error rate (BER) and normalized correlation (NC) values compared to previous zero watermarking approaches. AES performance analysis is performed to demonstrate its effectiveness. Using a  $256 \times 256$  image size, it takes only 2 s to apply the zero-watermark algorithm on the Raspberry Pi.

**Keywords:** Zero watermarking; Raspberry Pi; advanced encryption standard



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

Nowadays, everything is visible in digital communication via the Internet, so securing multimedia data such as images, videos, and text is challenging [1]. Many techniques, such as watermarking and encryption, are used to protect digital images transmitted via the Internet to achieve confidentiality, integrity, and authentication (CIA) [2,3]. The term “confidentiality” means the image is unavailable or revealed to unauthorized individuals; integrity ensures the information’s authenticity, accuracy and that unauthorized persons are not modifying it [4].

Powerful assaults on multimedia data are becoming more common as internet technologies evolve, so securing digital image data important [5]. Data encryption is needed before transmitting data over a network [6]. Encryption is a process where a message is encoded in a format that an unauthorized person cannot understand. Decryption is the inverse process, unlocking what is encoded in an unreadable format to recover the original message [7].

There are two primary encryption kinds: symmetric and asymmetric. Data is encrypted and decrypted using the same key in symmetric encryption. Asymmetric encryption, however, employs two distinct keys [8]. The advantage of symmetric encryption over asymmetric encryption is that it consumes fewer CPU cycles and is therefore faster and easier to implement [9].

In image processing, watermarking is crucial to ensure the transmitted images’ copyright [10]. Because of the wide use of the Internet, securing the transmitted images becomes essential [11]. Images are thought to be the most understandable multimedia tool that has viewable information that must be protected [12]. Image watermarking aims to hide information in the host image to provide copyright protection [13]. Traditional watermarking techniques safeguard image copyright by inserting a signal in the host image that is invisible but detectable [10]. As a result, the watermarked image created using these approaches is invariably warped. Zero watermarking doesn’t put any data into the host image itself. Instead, it creates an ownership authentication key based on a watermark signal and the host image’s essential features [14].

Feature extraction is an essential step in image processing applications; it keeps the image’s necessary information, and the goal is to select the most relevant features. Feature computation aims to extract unique values from the image that differentiate it from other images; a pixel sequence will no longer represent it. Still, it is now a vector of each selected feature, known as a feature vector [15]. Features can be classified into structural, statistical, and global transformations [16]. Among the global transformations used are image moments. These features are invariant to global deformations like translation, scaling, and rotations [17]. Grayscale images have less detail than color images; extraction of color image features is vital in many image-processing applications, including color image watermarking.

Most watermarking systems are PC-based but not very portable because of their size and weight. They cannot be easily used in many smart city applications or demanding situations, such as military usage. To get beyond the mobility limits of the PC, we employ embedded computers like the Raspberry Pi. It’s a portable platform with low cost compared to ordinary PCs and doesn’t use a lot of power [18]. Raspberry Pi operates in the open-source ecosystem and has many models that can be used in many projects [19]. The portability of the Raspberry Pi platform has allowed for extensive research in various domains, including image processing. The results reveal that the Raspberry Pi is a viable option for real-time applications [20–23]. It is affordable and controllable via the Internet [24]. In real-time applications, the Raspberry Pi can help make your system easier [25].

For the purpose of encrypting images, several common encryption techniques have been suggested. The best method for safeguarding images among encryption techniques is a hybrid approach. In this article we present hybrid data security techniques using a zero watermarking method and the AES. We want such methods that require less calculation and are quick to implement on a raspberry pi as time is a critical metric in image processing applications.

Nevertheless, there are several modern approaches that mix many encryption methods for images. Recent years have seen academics pay special attention to the similarities between chaotic systems and encryption [26]. Also deoxyribonucleic acid (DNA) technology is recently used in the image encryption field [27]. As technology advances quickly, the need for better encryption techniques grows as data is sent from one location to another. To safeguard the transmitted multimedia, a variety of encryption approaches can be utilized.

The zero watermarking technique (MFrLFMs) is used in our work on the Raspberry Pi to provide security to transmitted images. The AES encryption technique can encrypt the zero watermarking ownership verification key for security. Most watermarking methods are PC-based, so employing embedded portable devices like the Raspberry Pi is critical to overcoming the PC's limitations on portability. This article aims to secure the transmitted images using portable embedded systems such as the Raspberry Pi.

Zero watermarking techniques protect images from copyright violations without altering the original image. Therefore, there is absolutely no degradation in the visual image quality and the original image quality is maintained. Zero watermarking is an efficient technology, and its quick execution time makes it useful for applications where time is a key performance indicator. We can use AES-cipher block chaining (CBC) technique to encrypt the zero watermarking ownership verification key rather than delivering it to boost security.

The zero-watermark information for the images is calculated on a Raspberry Pi. The correct MFrLFMs are first calculated based on Gaussian numerical and exact kernels for the radial and angular kernels for the original color image. Second, in order to generate an exact moment feature to represent the host image, the most significant MFrLFM moments are picked, and then the selected features are binarized.

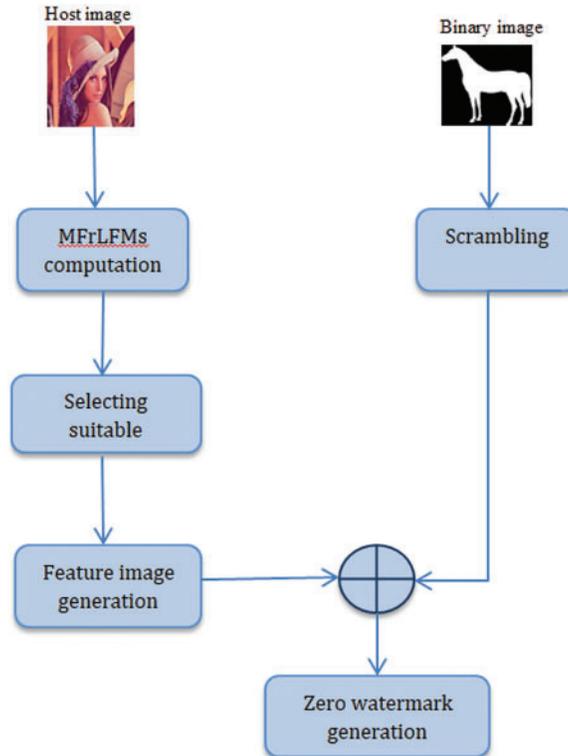
The zero-watermark picture is created by bitwise xORing the binary watermark with the binarized image features. The original image and zero watermark data are encrypted with AES-CBC before being sent. The receiver then verifies the accuracy of the received zero watermark data. The contributions of the paper are summarized as follows:

- MFrLFMs color image zero watermarking technique on a Raspberry Pi is used to generate a real-time ownership verification key.
- AES can be used to encrypt the ownership verification key and the original image before sending them to receiver.
- The receiver next verifies the received image's integrity.

The remainder of the paper consists of: Section 2 describe zero watermarking using MFrLFMs; Section 3 includes a description of the AES algorithm; Section 4 give a general look at Raspberry Pi; Section 5 demonstrates the implementation of zero watermarking algorithms with AES encryption on Raspberry Pi; Section 6 presents the performance analysis; and finally, Section 7 provides the conclusion.

## 2 Zero Watermarking Using MFrLFMs

Two steps make up MFrLFM's watermarking process: creating zero watermarks and verifying them. MFrLFMs features from the input image are employed in the generation stage to create the zero-watermark information. During the verification phase, the copyright status of the original image is verified. There are five steps in the generation process of the zero watermarking approaches as depicted in Fig. 1:



**Figure 1:** Zero watermarking generation

1. The watermark image is first scrambled using Arnold algorithm to increase the robustness of the watermark embedding mechanism and eliminate any spatial relationship between the watermark image pixels.

2. The MFrLFMs of the original color image are computed:

$$FrM_{pq} = \frac{2p+1}{2\pi} \sum_i \sum_j K_{pq}(r_i, \theta_{ij}) \hat{g}_c(r_i, \theta_{ij}) \quad (1)$$

where:

$$K_{pq}(r_i, \theta_{ij}) = I_p(r_i) I_q(\theta_{ij}) \quad (2)$$

And  $\hat{g}_c(r_i, \theta_{ij})$  are the three primary channels of the input image c{R, G, and B}. The angular and radial kernels are defined as:

$$I_q(\theta_{ij}) = \int_{V_{ij}}^{V_{ij+1}} e^{-iq\theta} d\theta \quad (3)$$

$$I_p(r_i) = \int_{U_i}^{U_{i+1}} L_p(\alpha, r) r dr \tag{4}$$

where:

$$V_{i,j+1} = \theta_{i,j} + \frac{\Delta\theta_{i,j}}{2}, V_{i,j} = \theta_{i,j} - \frac{\Delta\theta_{i,j}}{2} \tag{5}$$

$$U_{i+1} = R_i + \frac{\Delta R_i}{2}, U_i = R_i - \frac{\Delta R_i}{2} \tag{6}$$

3. The accurate MFrLFMs coefficients are selected and constructed as a feature vector. MFrLFMs moments of  $q=4m$  are not considered and only positive repetition  $q$  is chosen to prevent redundant information.

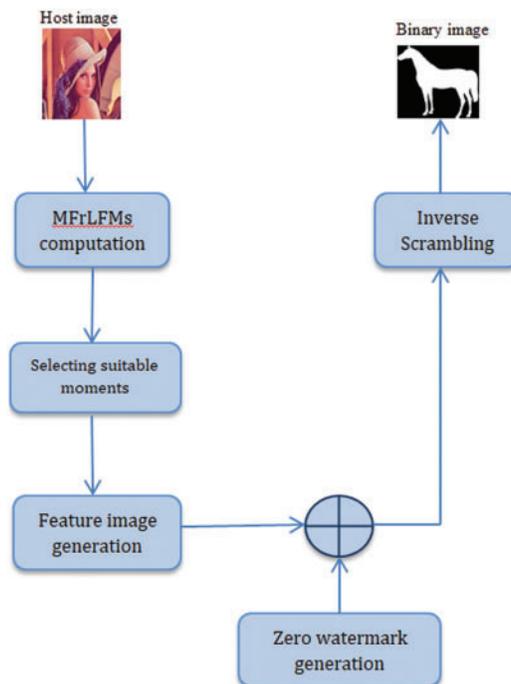
4. The following binarization algorithm is used to produce the binary feature vector (X) out of feature vector (Y):

$$X_j = \begin{cases} 1 & \text{if } Y_j \geq T \\ 0 & \text{if } Y_j < T \end{cases} \quad j = 1, 2, 3 \dots MXN \tag{7}$$

where  $T$  is the threshold and  $M \times N$  are the watermark image dimensions.

5. Using the scrambled watermark image and the image features, we can create the zero watermark image using a bitwise XOR.

There are five steps in the verification of the zero-watermarking approach as depicted in Fig. 2:



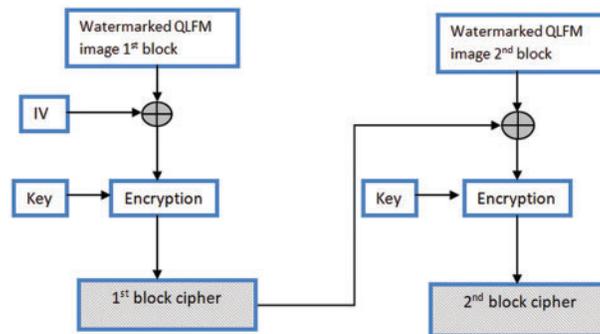
**Figure 2:** Zero watermarking verification

1. MFrLFMs are calculated for the protected image.
2. The accurate MFrLFMs coefficients are selected and constructed as a feature vector.
3. Binary feature image synthesis (Binarization).
4. A scrambled watermark image is created by XOR binary image with the equivalent secured image zero watermarks.
5. The recovered watermark is obtained using the inverse Arnold transform.

### 3 Advanced Encryption Standard Algorithm

AES is a symmetric encryption algorithm that uses a single key for both encryption and decryption. The block size of AES is 128 bits. There are three types of AES algorithms: AES-128, AES-192, and AES-256. These types are classified according to the key size used in the algorithm. The AES algorithm's security level increases with the key size used [28].

The AES algorithm uses a round function for the data encryption and decryption. Each round consists of four operations for the encryption process: substitute byte, shift rows, mix columns, and add round key [29]. The reverse operation in rounds is used for the decryption process. The number of rounds is based on the algorithm key size. The most typical number of rounds is ten rounds for 128-bit keys, 12 rounds for 192-bit keys, or 14 rounds for 256-bit keys [30]. Block cipher uses five different modes of operation: electronic codebook (ECB), cipher block chaining (CBC), cipher feedback (CFB), output feedback (OFB), and counter (CTR) modes. We use CBC mode with an AES algorithm to make sure our data is safe. In CBC mode, an initialization vector (IV) is XORed with plain text. The initialization vector (IV) in the first round is a random value. In the following rounds, the initialization vector (IV) is the cipher text obtained from the previous block round, as in Fig. 3. You won't receive the same encrypted text data from the same piece of plain text data [31].



**Figure 3:** Simple 2 Blocks of CBC Encryption Mode

### 4 Raspberry Pi

The Raspberry Pi is a small portable computer. It was developed by the Raspberry Pi Foundation. The Raspberry Pi is a quad-core computer with parallel computing skills, which can be utilized to accelerate computations and processes [32]. Raspberry Pi can be useful in image processing areas because of its portability, parallelism, cheap cost, and minimal power usage [33]. The Raspberry Pi is an open-source computer that comes in a range of different models, the latest model is Raspberry Pi 4. The RAM size of the Raspberry Pi model 4 varies based on the application's needs. Since we use the

Raspberry Pi model 4 in our applications, we will concentrate on the requirements of the most recent model of Raspberry Pi in this section.

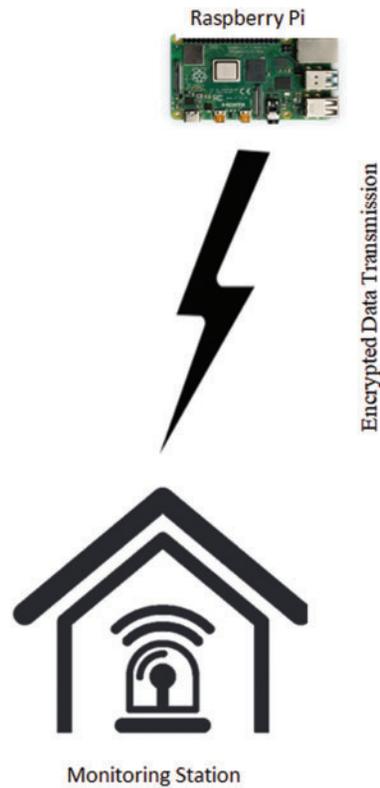
Raspberry Pi 4, shown in Fig. 4 [34], has a 1.5 GHz processor and is available with LPDDR4-3200 SDRAM and multiple RAM choices of 1 GB, 2 GB, or 4 GB. It has a quad-core Cortex-A72 (A.R.M. v8) processor, and a Broadcom BCM2711. It also has two micro-HDMI interfaces, a 40-pin GPIO connector and Gigabit Ethernet.



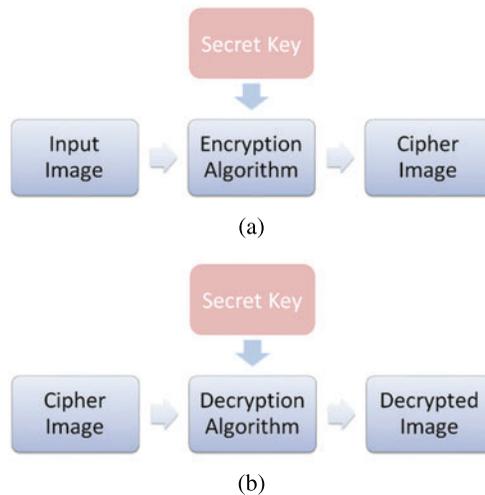
Figure 4: Raspberry Pi model 4 [34]

## 5 Implementation of Zero Watermarking on Raspberry Pi

- It is possible to acquire a single Raspberry Pi and use it to manage a light workload with minimal power usage due to its low price. Many smart city applications utilize embeddable systems like the Raspberry Pi since PCs are not portable.
- Due to its portability and ability to be managed through the Internet, the Raspberry Pi platform has been used for research in domains like image processing [35].
- Real-time applications employ the Raspberry Pi to minimize system complexity.
- For the demands of portable watermarking applications in smart cities, MFrLFMs zero watermarking is implemented on Raspberry Pi.
- The MFrLFMs watermarking steps are all done on the Raspberry Pi model 4.
- Fig. 5 shows our technique for securing transmitted color images using a Raspberry Pi.
- The zero watermarking algorithm is applied to the Raspberry Pi to produce a real-time ownership verification key.
- Before sending the ownership verification key and the original image to the monitoring station (receiver), the Raspberry Pi encrypts them with AES to provide cryptographic authentication and hide any viewable information in the original image.
- The monitoring station next verifies the received image's integrity to confirm its authenticity and that it has not been tampered with.
- Cryptography is one of the most well-known data protection methods [36]. It is a way of transmitting and receiving encrypted data that only the sender or recipient may decode [37]. The key used for decryption is only known to the sender and the recipient so only the intended receiver may interpret and decode the document. This approach is also commonly used to secure and preserve digital images from unauthorized access. For securing the transmitted images, AES is a practical approach that can be used. Fig. 6 demonstrates the basic steps of image encryption and decryption.



**Figure 5:** Securing transmitted color images using a Raspberry Pi

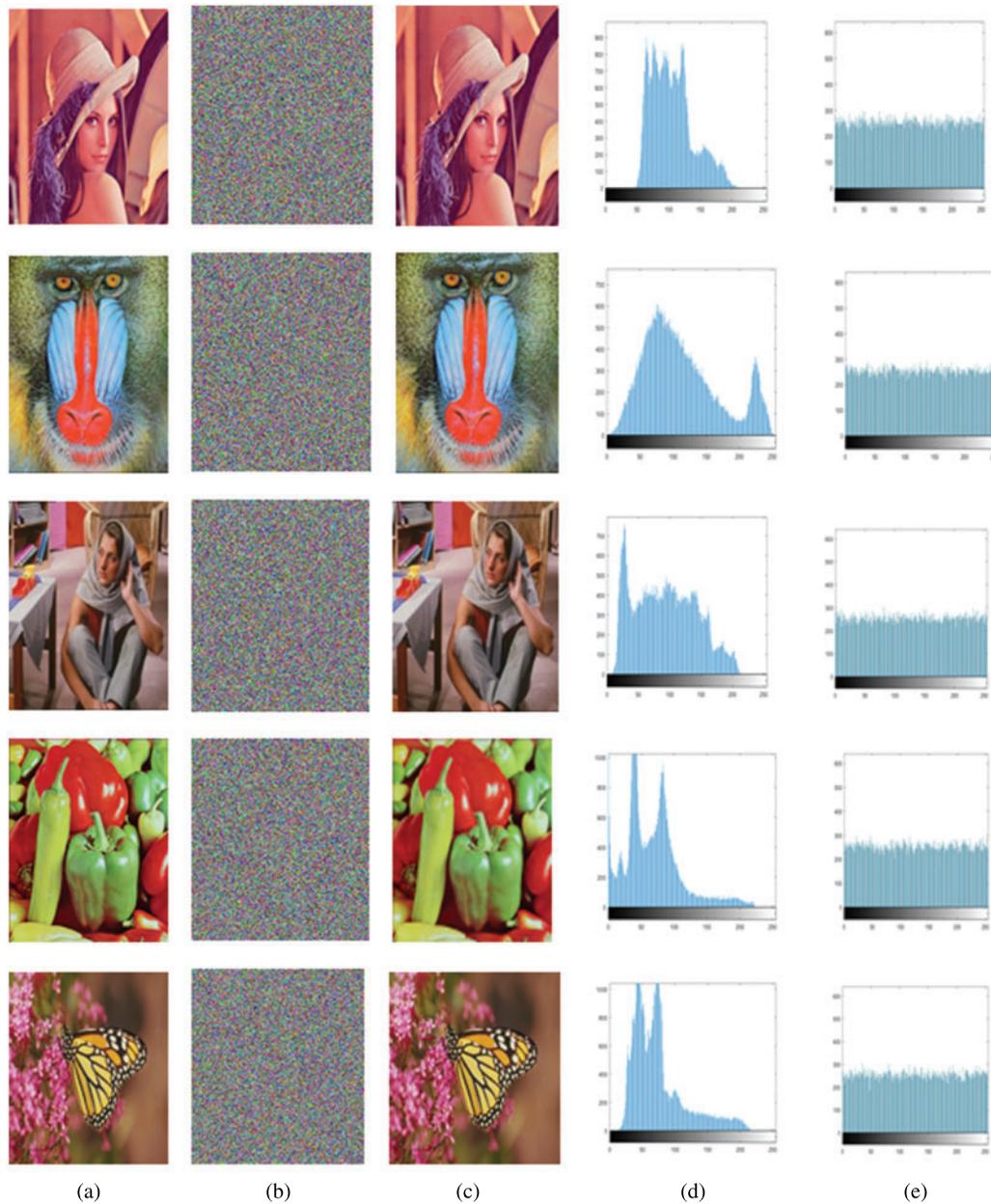


**Figure 6:** (a) Image encryption block diagram; (b) Image decryption block diagram

- The original image on the Raspberry Pi is encrypted using the AES-CBC algorithm to secure the transmitted image. The encryption uses a 32-byte key length (256-bit) that is only known between the sender and receiver for more security. Only the receiver with the correct key can decrypt the cipher image. Otherwise, it is meaningless that the intruders cannot get any

information or details from it. This technique is applied to color images with a size of  $256 \times 256$  from different datasets to ensure the encryption and decryption performance using AES-CBC in the C++ programming language, as shown in Fig. 7.

- The monitoring station receives this encrypted data from the Raspberry Pi and performs the verification steps to ensure the integrity of the received images.



**Figure 7:** (a) Color images; (b) cipher images; (c) decrypted images; (d) image blue-channel histogram; (e) ciphered image blue-channel histogram

## 6 Experimental Analysis

In this section, we've put together a bunch of tests to see how well MFrLFM's zero watermarking works. We looked at the outputs in terms of execution time and watermark robustness against attacks, and a comparison with similar zero watermarking algorithms is also given.

Furthermore, the AES algorithm is assessed in terms of key sensitivity analysis, processing time, information entropy, histogram analysis, and correlation.

### 6.1 MFrLFMs Watermarking Time Analysis

Time is a critical metric in image processing applications; this algorithm proves its efficiency in terms of execution time. The calculation time of MFrLFM's watermarking algorithm is assessed on a Raspberry Pi. The analysis uses moment order 21, a watermark image of size  $32 \times 32$ , and color image sizes  $256 \times 256$  and  $512 \times 512$ . The watermarking time and encryption on the three image channels are 2 s for images of size  $256 \times 256$  and 6 s for images of size  $512 \times 512$ .

### 6.2 MFrLFMs Robustness to Various Attacks

The original watermark image is the  $256 \times 256$  color image "Lena". The  $32 \times 32$  binary image "horse" is used as a watermark image to evaluate the algorithm's geometric attack resistance. The following criteria were used:

#### 1-PSNR (Peak Signal To Noise Ratio)

PSNR is used to evaluate the quality of images that have been attacked. The PSNR can be calculated as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (8)$$

where Mean Square Error is:

$$(MSE) = \frac{1}{N^2} \left( \sum_{i=1}^N \sum_{j=1}^N [f_{\text{attacked image}}(i, j) - f_{\text{original image}}(i, j)]^2 \right) \quad (9)$$

#### 2-BER

The BER statistic measures the proportion of incorrectly recovered binary bits to the total number of encoded bits. The efficiency of the embedding system increases as BER decreases.

$$BER = \frac{Berror}{nbits} \quad (10)$$

where nbits is the total number of bits and Berror is the wrongly extracted bits.

#### 3-NC

NC is used to determine how similar the extracted watermark is to the original as follows:

$$NC = \frac{\sum_{i=1}^p \sum_{j=1}^q [f_{\text{original watermark}}(i, j) X f_{\text{extracted watermark}(i, j)}]}{\sum_{i=1}^p \sum_{j=1}^q [f_{\text{original watermark}}(i, j)]^2} \quad (11)$$

PSNR, BER, and N.C. values of the extracted watermark are determined for various attacks, as shown in [Table 1](#). This analysis reveals that despite several distortions of the original color image, the recovered watermarks were still recognizable, and the values were inclined towards ideal values.

**Table 1:** Extraction of binary watermark under several distortions

Attack	Rotation 15°	Rotation 25°	Rotation 35°
Attacked image			
Retrieved watermark			
PSNR	10.2691	9.2569	8.7273
BER	0.0166	0.0107	0.0088
NC	0.9763	0.9777	0.9751
<b>Attack</b>	<b>Scaling 1.25</b>	<b>Scaling 1.5</b>	<b>Scaling 1.75</b>
Attacked image			
Retrieved watermark			
PSNR	11.5488	12.0921	11.3961
BER	0.0049	0.0059	0.0068
NC	0.9833	0.9764	0.9778
<b>Attack</b>	<b>JPEG compression 70</b>	<b>JPEG compression 80</b>	<b>JPEG compression 90</b>
Attacked image			
Retrieved watermark			
PSNR	35.6511	37.4099	40.8021
BER	0.0107	0.0039	0.002
NC	0.9833	0.9819	0.9903
<b>Attack</b>	<b>Rotation 25° + JPEG compression 90</b>	<b>Rotation 45° + JPEG compression 90</b>	<b>Median filter 3x3</b>
Attacked image			
Retrieved watermark			
PSNR	9.2635	8.6155	31.8951
BER	0.0117	0.0137	0.0273
NC	0.9762	0.9708	0.9559

### 6.3 MFrLFMs Comparison with Similar Algorithms

Yang et al. [38] created a novel zero watermarking method utilizing the fast quaternion generic polar complex exponential transform and an asymmetric tent map. Kang et al. [39] utilize a color image zero watermarking approach based on compound chaotic maps and polar harmonic transforms (PHTs). Wang et al. [40] use geometrically invariant quaternion exponent moments (QEMs) for color image zero watermarking. A robust zero-watermarking technique based on quaternion polar complex exponential transform was introduced by Xia et al. [41]. PHTs with decimal-order and chaotic systems are used in zero watermarking technique by Xia et al. [42].

Tables 2 and 3 compares the effectiveness of the MFrLFMs watermarking technique to some zero watermarking strategies [38–42] using BER and NC values. It is clear from the comparisons that MFrLFMs are superior, and the values tend towards ideal values. Due to these observations and outcomes, we were persuaded to utilize the suggested method in our work with the Raspberry Pi.

**Table 2:** NC and BER values of the distorted watermark

Attacks		Yang et al. [38]		Xiaobing et al. [39]		Wang et al. [40]		MFrLFMs method	
		BER	NC	BER	NC	BER	NC	BER	NC
Rotation	25°	0.0127	0.9843	0.0195	0.9759	0.0205	0.9745	0.0059	0.9928
	35°	0.0205	0.9754	0.0244	0.9697	0.0293	0.9643	0.0098	0.9880
	45°	0.0146	0.9820	0.0166	0.9794	0.0283	0.9650	0.0078	0.9904
Scaling	0.75	0.0186	0.9769	0.0205	0.9747	0.0215	0.9736	0.0088	0.9893
	1.5	0.0098	0.988	0.0107	0.9867	0.0127	0.9843	0.0039	0.9952
Translation	H4,V4	0.0107	0.9867	0.0156	0.9806	0.0215	0.9737	0.0049	0.994
JPEG compression	70%	0.0088	0.9894	0.0107	0.9872	0.0127	0.9854	0.0039	0.9952
	90%	0.0078	0.9904	0.0088	0.9893	0.105	0.9867	0.0029	0.9978

**Table 3:** NC and BER values of the distorted watermark

Attacks		Xia et al. [41]		Xia et al. [42]		MFrLFMs method	
		BER	NC	BER	NC	BER	NC
Rotation	15°	0.0184	0.9762	0.0192	0.9736	0.006	0.9916
	25°	0.0275	0.9664	0.0285	0.9639	0.0059	0.9928
	35°	0.0263	0.9673	0.0273	0.9645	0.0098	0.9880
Scaling	0.75	0.0194	0.9753	0.0207	0.9729	0.0088	0.9893
	1.75	0.0109	0.9863	0.0118	0.9840	0.0043	0.9938
Translation	H3,V3	0.0196	0.9751	0.0206	0.9725	0.0054	0.9928
JPEG compression	70%	0.0108	0.9876	0.0118	0.9853	0.0039	0.9952
	90%	0.0105	0.9885	0.0119	0.9862	0.0029	0.9978

### 6.4 AES: Key Sensitive Analysis

A secure and efficient encryption algorithm should be sensitive to plaintext and the key [43]. Highly secure image encryption algorithms demand high sensitivity to ensure that the cipher image cannot be decrypted correctly if the encryption and decryption keys differ slightly [44]. The analysis result reveals the strength of the AES-CBC algorithm, where the picture can't be decrypted if the key value is altered even by one bit, as shown in Fig. 8.

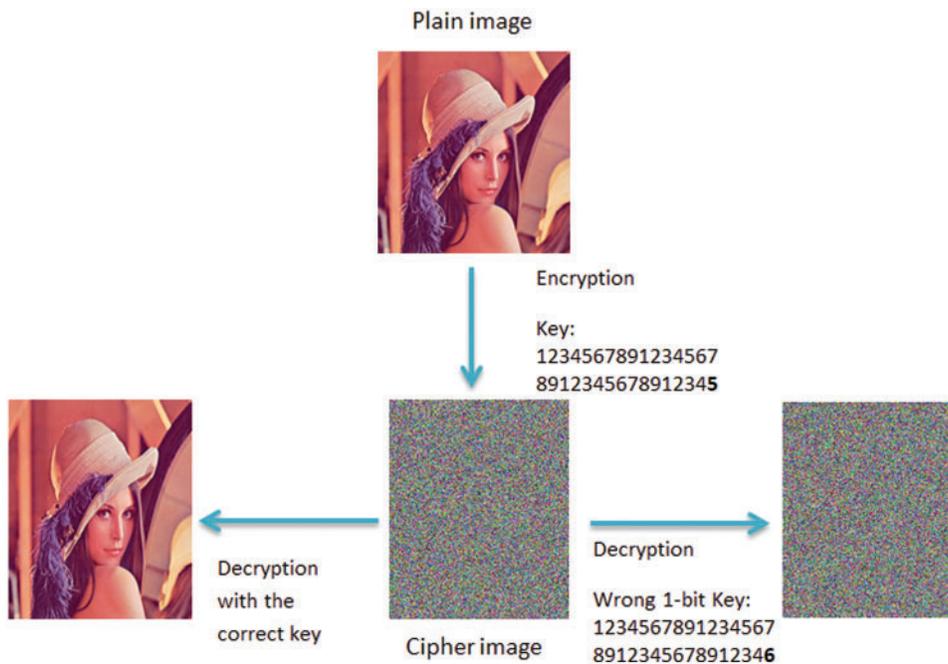


Figure 8: Test of key sensitivity of AES-CBC algorithm

### 6.5 AES: Execution Time

We have used a Raspberry Pi model 4 with a 1.5 GHz processor to run the AES-CBC encryption algorithm using crypto++, the algorithm takes 0.003 s to encrypt one color channel of Lena's image with a size of 256 × 256 and a 32-byte key length, and it takes 0.008 s to encrypt all three RGB channels of the image.

### 6.6 Information Entropy Analysis

Entropy is employed to evaluate the performance of the encryption algorithm [45]. Entropy is a crucial feature that reflects a source of information's randomness and defines unpredictability. Each RGB channel in a color image is represented as 8 bits, with pixel values varying from 0 to 255. Therefore, the entropy has a maximum value of 8. The entropy value of the encrypted image should be close to 8 for an efficient image encryption algorithm. The cipher 'Lena' image entropy is shown in Table 4. Image entropy can be calculated as [46]:

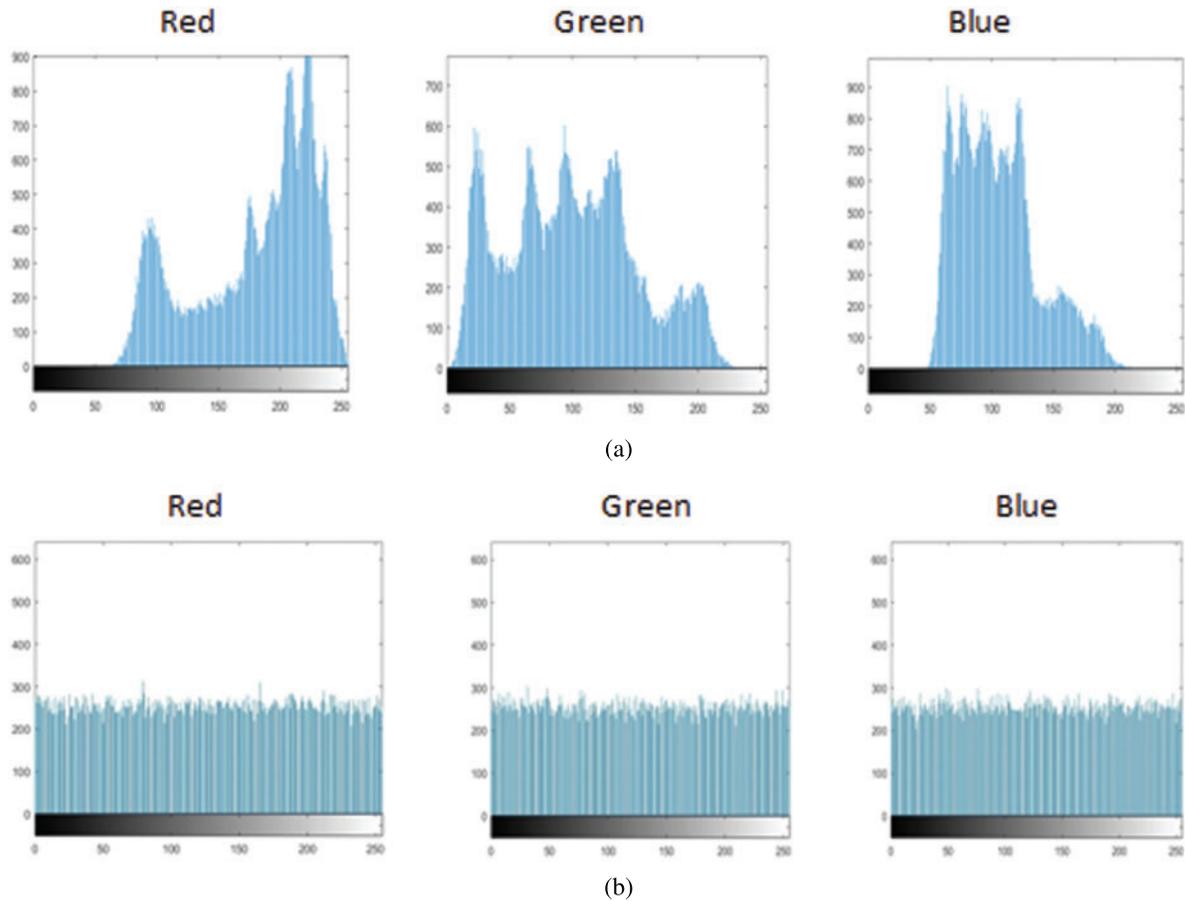
$$H(m) = - \sum_{i=0}^{2^n-1} P(x_i) \log_2 P(x_i) \tag{12}$$

**Table 4:** Information entropy of both plain and cipher Lena image

H(m)	R	G	B
Cipher image	7.9960	7.9965	7.9965

### 6.7 Histogram Analysis

The image histogram reflects how the pixels in an image are distributed [47]. When the histogram is uniformly distributed, this means statistical assaults are less likely to succeed [48]. Histogram analysis is used to identify the distributions of plaintext and cipher text pixel values [49]. The histogram of the encrypted image is uniformly distributed. It differs considerably from the plain image, as seen in Fig. 9. Furthermore, there is no loss of image quality after the decryption step. Having the encrypted image makes it difficult for attackers to retrieve the original images or any information about them. This means that AES-CBC is strong enough to handle statistical attacks. Fig. 9 demonstrates the histogram for RGB channels for the plain image of Lena and the encrypted image of Lena, respectively.

**Figure 9:** (a) Lena plain image histogram; (b) Encrypted Lena image histogram

### 6.8 Correlation Analysis

The correlation coefficient describes how the adjacent pixels are correlated to one another in the image. As in Table 5, the adjacent pixels strongly correlate in all three directions in the plain image ‘Lena’ ( $\approx 1$ ). The correlation between neighboring pixels in the ciphered image’s three directions must be as small as feasible ( $\approx 0$ ) to resist the statistical attacks.

**Table 5:** The correlation coefficient of both plain and cipher Lena image

	Plain image			Cipher image		
	R	G	B	R	G	B
Vertical	0.9789	0.9714	0.9559	-0.0038	0.0006	0.0008
Horizontal	0.9572	0.9432	0.9284	-0.0030	0.0066	-0.0038
Diagonal	0.9339	0.9193	0.9007	-0.0016	-0.0006	0.0023

The correlation coefficient of adjacent pixels is [50]:

$$r_{mn} = \frac{\text{Cov}(m, n)}{\sqrt{D(m)}\sqrt{D(n)}} \tag{13}$$

where:

- m and n are values of the adjacent pixels in the image.
- Cov(m, n) is the covariance.
- D(m) is the variance.

### 6.9 Contrast

The observer can analyze the pattern of an image using contrast. The contrast intensity of a pixel and its adjacent pixel is measured via contrast analysis throughout the entire image. Table 6 shows the contrast values of the original and encrypted Lena image. Contrast values for the encrypted image should be high for strong encryption. It can be expressed as [51]:

$$contrast = \sum_{x,y} |x - y|^2 P(x, y) \tag{14}$$

where P(x, y) is the number of grey-level co-occurrence matrices (GLCM).

**Table 6:** The contrast of both plain and cipher Lena image

Contrast	R	G	B
Original image	0.3920	0.3789	0.3552
Cipher image	10.5525	10.5566	10.5032

### 6.10 Energy

This analysis is used to calculate the amount of information in an image. To accomplish robust encryption, the encrypted image should be low in energy. The energy values of original and encrypted lena image is given in [Table 7](#). The energy parameter may be computed as [52]:

$$Energy = \sum_{x,y} P(x,y)^2 \quad (15)$$

**Table 7:** The energy of both plain and cipher Lena image

Energy	R	G	B
Plain image	0.1369	0.0968	0.1653
Cipher image	0.0156	0.0156	0.0156

### 6.11 Homogeneity

The term homogeneity refers to how closely the elements in GLCM. GLCM is a table-based combination of pixel brightness values or grey levels. The homogeneity values of original and encrypted lena image is given in [Table 8](#). Encryption is preferable if the homogeneity is as low as feasible. It's calculated as [53]:

$$H = \sum_{x,y} \frac{P(x,y)}{1 + |x - y|} \quad (16)$$

**Table 8:** The homogeneity of both plain and cipher Lena image

Homogeneity	R	G	B
Plain image	0.8638	0.8708	0.8614
Cipher image	0.3884	0.3888	0.3895

### 6.12 NPCR and UACI

NPCR (number of pixels change rate) and UACI (unified average changing intensity) are two metrics used to assess how effective image encryption methods are against different differential attacks. These are used to see how minor modifications in the original image affect encryption by comparing the original and encrypted image data. To withstand various differential attacks, the encrypted photos and the original image ought to differ significantly [54]. NPCR and UACI values for Lena image is calculated in [Table 9](#).

**Table 9:** NPCR and UACI values for Lena image

	NPCR	UACI
Lena image	0.996292	0.333844

NPCR and UACI can be calculated as [55]:

$$D(i, j) = \begin{cases} 0 & \text{if } C_1(i, j) = C_2(i, j) \\ 1 & \text{if } C_1(i, j) \neq C_2(i, j) \end{cases} \quad (17)$$

$$NPCR = \sum_{ij} \frac{D(i, j)}{T} \quad (18)$$

$$UACI = \sum_{ij} \frac{|C_1(i, j) - C_2(i, j)|}{F.T} \quad (19)$$

where:

- $C_1$  and  $C_2$  → cipher text images before and after a single pixel modification.
- $T$  → total amount of pixels in cipher image.
- $F$  → biggest pixel value.

## 7 Conclusions

Information security is essential for securing the transmitted media, such as images, to ensure the CIA triad. This suggested approach is implemented on Raspberry Pi embedded device that can be used in difficult environments because it solves the problem of a computer's restricted mobility. The zero-watermark technique is implemented on Raspberry Pi. The Raspberry Pi then sends the image's zero watermark verification key and the original image to a monitoring station, and this data is encrypted before sending it using the AES-CBC encryption technique using a 256-bit symmetric random key known only to the transmitter (the Raspberry Pi) and the receiver. The receiver can then decrypt the receiving data to confirm its integrity. This technique is utilized on the Raspberry Pi model 4 for smart-cities watermarking applications that require portability. Compared to an ordinary expensive PC with limited portability, which is unsuitable for many smart-city applications, the implementation of this technique on the Raspberry Pi shows excellent performance over time. The system execution time is too short; making it perfect for real-time applications. In the future, we want to expand the suggested technology and implement it on a drone.

**Acknowledgement:** This project is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R442), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Funding Statement:** Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R442), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] H. Qiu, K. Kapusta, Z. Lu, M. Qiu and G. Memmi, "All-Or-Nothing data protection for ubiquitous communication: Challenges and perspectives," *Information Sciences*, vol. 502, pp. 434–445, 2019.
- [2] M. Khan, F. Masood and A. Alghafis, "Secure image encryption scheme based on fractals key with Fibonacci series and discrete dynamical system," *Neural Computing and Applications*, vol. 32, no. 15, pp. 11837–11857, 2019.

- [3] R. S. Devi, A. N. Aravind, J. C. Vishal, D. Amritha, K. Thenmozhi *et al.*, “Image encryption through RNA approach assisted with neural key sequences,” *Multimedia Tools and Applications*, vol. 79, pp. 12093–12124, 2020.
- [4] V. Monev, “Defining and applying information security goals for blockchain technology,” in *2020 Int. Conf. on Information Technologies (InfoTech)*, Piscataway, IEEE, pp. 1–4, 2020. <https://doi.org/10.1109/InfoTech49733.2020.9211073>
- [5] J. S. Khan, J. Ahmad, S. S. Ahmed, H. A. Siddiqa, S. F. Abbasi *et al.*, “DNA key based visual chaotic image encryption,” *Journal of Intelligent & Fuzzy Systems*, vol. 37, pp. 2549–2561, 2019.
- [6] A. S. Alanazi, N. Munir, M. Khan, M. Asif and I. Hussain, “Cryptanalysis of novel image encryption scheme based on multiple chaotic substitution boxes,” *IEEE Access*, vol. 9, pp. 93795–93802, 2021.
- [7] M. A. Razzaq, R. A. Sheikh, A. Baig and A. Ahmad, “Digital image security: Fusion of encryption, steganography, and watermarking,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 5, pp. 224–228, 2017.
- [8] P. Chinnasamy, S. Padmavathi, R. Swathy and S. Rakesh, “Efficient data Security using hybrid cryptography on cloud computing,” in *Inventive Communication and Computational Technologies*, Singapore: Springer, pp. 537–547, 2021.
- [9] D. N. S. Rani, D. A. N. M. Juliet and K. R. Devi, “An image encryption & decryption and comparison with text-AES algorithm,” *International Journal of Scientific & Technology Research*, vol. 8, no. 7, pp. 668–673, 2019.
- [10] M. Begum and M. S. Uddin, “Digital image watermarking techniques: A review,” *Information*, vol. 11, no. 2, pp. 110, 2020.
- [11] A. Ray and S. Roy, “Recent trends in image watermarking techniques for copyright protection: A survey,” *International Journal of Multimedia Information Retrieval*, vol. 9, pp. 249–270, 2020.
- [12] R. Wu, S. Gao, X. Wang, S. Liu, Q. Li *et al.*, “AEA-NCS: An audio encryption algorithm based on a nested chaotic system,” *Chaos, Solitons & Fractals*, vol. 165, pp. 112770, 2022.
- [13] O. P. Singh, A. K. Singh, G. Srivastava and N. Kumar, “Image watermarking using soft computing techniques: A comprehensive survey,” *Multimedia Tools and Applications*, pp. 1–32, 2020. <https://doi.org/10.1007/s11042-020-09606-x>
- [14] H. Y. Yang, S. R. Qi, P. P. Niu and X. Y. Wang, “Color image zero-watermarking based on fast quaternion generic polar complex exponential transform,” *Signal Processing: Image Communication*, vol. 82, pp. 115747, 2020.
- [15] B. Zhao, L. Gao, W. Liao and B. Zhang, “A new kernel method for hyperspectral image feature extraction,” *Geo-Spatial Information Science*, vol. 20, no. 4, pp. 309–318, 2017.
- [16] M. Rabi, M. Amrouch and Z. Mahani, “Evaluation of features extraction and classification techniques for offline handwritten Tifinagh recognition,” *Global Journal of Computer Science and Technology*, vol. 16, no. 5, pp. 37–42, 2017.
- [17] J. Flusser, B. Zitova and T. Suk, *Moments And Moment Invariants In Pattern Recognition*. Hoboken: John Wiley & Sons, 2009.
- [18] G. Senthilkumar, K. Gopalakrishnan and V. S. Kumar, “Embedded image capturing system using Raspberry pi system,” *International Journal of Emerging Trends & Technology in Computer Science*, vol. 3, no. 2, pp. 213–215, 2014.
- [19] S. Gupta, U. Raikar, B. M. P. Patil and R. Molavade, “Image processing based intelligent traffic control system by using Raspberry Pi,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, pp. 66–70, 2018. <https://doi.org/10.22214/ijraset.2018.4014>
- [20] K. M. Hosny, A. Magdi, N. A. Lashin, O. El-Komy and A. Salah, “Robust color image watermarking using multi-core Raspberry pi cluster,” *Multimedia Tools and Applications*, vol. 81, pp. 17185–17204, 2022.

- [21] K. M. Hosny, A. Y. Hamad, O. Elkomy and E. R. Mohamed, "Fast and accurate face recognition system using MORSCMs-LBP on embedded circuits," *PeerJ Computer Science*, vol. 8, pp. e1008, 2022.
- [22] M. Magdy, N. I. Ghali, S. Ghoniemy and K. M. Hosny, "Multiple zero-watermarking of medical images for internet of medical things," *IEEE Access*, vol. 10, pp. 38821–38831, 2022.
- [23] K. M. Hosny, M. M. Darwish, K. Li and A. Salah, "COVID-19 diagnosis from CT scans and chest X-ray images using low-cost Raspberry Pi," *PLoS One*, vol. 16, pp. e0250688, 2021.
- [24] M. Sajjad, M. Nasir, K. Muhammad, S. Khan, Z. Jan *et al.*, "Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities," *Future Generation Computer Systems*, vol. 108, pp. 995–1007, 2017.
- [25] K. S. Shilpashree, H. Lokesh and H. Shivkumar, "Implementation of image processing on Raspberry Pi," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 5, pp. 199–202, 2015.
- [26] M. Khan, S. S. Jamal, M. M. Hazzazi, K. M. Ali, I. Hussain *et al.*, "An efficient image encryption scheme based on double affine substitution box and chaotic system," *Integration*, vol. 81, pp. 108–122, 2021.
- [27] M. Yildirim, "Optical color image encryption scheme with a novel DNA encoding algorithm based on a chaotic circuit," *Chaos, Solitons & Fractals*, vol. 155, pp. 111631, 2022.
- [28] P. Deshmukh, "An image encryption and decryption using AES algorithm," *International Journal of Scientific & Engineering Research*, vol. 7, no. 2, pp. 23–29, 2016.
- [29] M. Asif and T. Shah, "BCH codes with computational approach and its applications in image encryption," *Journal of Intelligent & Fuzzy Systems*, vol. 37, pp. 3925–3939, 2019.
- [30] B. B. Raju, A. Krishna and G. Mishra, "Implementation of an efficient dynamic AES algorithm using ARM-based SoC," in *2017 4th IEEE Uttar Pradesh Section Int. Conf. on Electrical, Computer, and Electronics (UPCON)*, pp. 39–43, 2017. <https://doi.org/10.1109/UPCON.2017.8251019>
- [31] V. Shadangi, S. K. Choudhary, K. A. K. Patro and B. Acharya, "Novel Arnold scrambling-based CBC-AES image encryption," *International Journal of Control Theory and Applications*, vol. 10, no. 15, pp. 93–105, 2017.
- [32] K. M. Hosny, A. Magdi, A. Salah, O. El-Komy and N. A. Lashin, "Internet of things applications using Raspberry-Pi: A survey," *International Journal of Electrical & Computer Engineering*, vol. 13, pp. 902–910, 2023.
- [33] K. M. Hosny, A. Salah and A. Magdi, "Parallel image processing applications using Raspberry Pi," in *Recent Advances in Computer Vision Applications Using Parallel Processing*, Cham: Springer International Publishing, pp. 107–119, 2023.
- [34] The Raspberry Pi 4 B. (n.d.). [Photograph]. [https://upload.wikimedia.org/wikipedia/commons/thumb/1/10/Raspberry\\_Pi\\_4\\_Model\\_B\\_-\\_Top.jpg/330px-Raspberry\\_Pi\\_4\\_Model\\_B\\_-\\_Top.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/1/10/Raspberry_Pi_4_Model_B_-_Top.jpg/330px-Raspberry_Pi_4_Model_B_-_Top.jpg)
- [35] K. M. Hosny, A. Magdi, A. Salah, O. El-Komy and N. A. Lashin, "Internet of things applications using Raspberry-Pi: A survey," *International Journal of Electrical & Computer Engineering*, vol. 13, pp. 2088–8708, 2023.
- [36] V. A. Daisy, C. V. Joe and S. S. S. Sugi, "An image-based authentication technique using visual cryptography scheme," in *2017 Int. Conf. on Inventive Systems and Control (ICISC)*, Piscataway, IEEE, pp. 1–6, 2017. <https://doi.org/10.1109/ICISC.2017.8068666>
- [37] V. Pavithra and C. Jeyamala, "A survey on the techniques of medical image encryption," in *2018 IEEE Int. Conf. on Computational Intelligence and Computing Research (ICIC)*, Piscataway, IEEE, pp. 1–8, 2018.
- [38] H. Y. Yang, S. R. Qi, P. P. Niu and X. Y. Wang, "Color image zero-watermarking based on fast quaternion generic polar complex exponential transform," *Signal Processing: Image Communication*, vol. 82, pp. 115747, 2020.

- [39] X. Kang, F. Zhao, Y. Chen, G. Lin and C. Jing, "Combining polar harmonic transforms and 2D compound chaotic map for distinguishable and robust color image zero-watermarking algorithm," *Journal of Visual Communication and Image Representation*, vol. 70, pp. 102804, 2020.
- [40] C. P. Wang, X. Y. Wang, Z. Q. Xia, C. Zhang and X. J. Chen, "Geometrically resilient color image zero-watermarking algorithm based on quaternion exponent moments," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 247–259, 2016.
- [41] Z. Xia, X. Wang, C. Wang, B. Ma, H. Zhang *et al.*, "Novel quaternion polar complex exponential transform and its application in color image zero-watermarking," *Digital Signal Processing*, vol. 116, pp. 103130, 2021.
- [42] Z. Xia, X. Wang, B. Han, Q. Li, X. Wang *et al.*, "Color image triple zero-watermarking using decimal-order polar harmonic transforms and chaotic system," *Signal Processing*, vol. 180, pp. 107864, 2021.
- [43] Y. Zhang, A. Chen, Y. Tang, J. Dang and G. Wang, "Plaintext-related image encryption algorithm based on perceptron-like network," *Information Sciences*, vol. 526, pp. 180–202, 2020.
- [44] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li *et al.*, "EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory," *Information Sciences*, vol. 621, pp. 766–781, 2023.
- [45] M. Asif, S. Mairaj, Z. Saeed, M. U. Ashraf, K. Jambi *et al.*, "A novel image encryption technique based on Mobius transformation," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–14, 2021.
- [46] X. Wang, H. Zhao, L. Feng, X. Ye and H. Zhang, "High-sensitivity image encryption algorithm with random diffusion based on dynamic-coupled map lattices," *Optics and Lasers in Engineering*, vol. 122, pp. 225–238, 2019.
- [47] J. Yang, W. Zhong and Z. Miao, "On the Image enhancement histogram processing," in *2016 3rd Int. Conf. on Informative and Cybernetics for Computational Social Systems (ICCSS)*, Piscataway, IEEE, pp. 252–255, 2016.
- [48] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li *et al.*, "Asynchronous updating Boolean network encryption algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. <https://doi.org/10.1109/TCSVT.2023.3237136>
- [49] S. Gao, R. Wu, X. Wang, J. Wang, Q. Li *et al.*, "A 3D model encryption scheme based on a cascaded chaotic system," *Signal Processing*, vol. 202, pp. 108745, 2023.
- [50] C. Li, D. Lin, B. Feng, J. Lü and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.
- [51] Y. Alghamdi, A. Munir and J. Ahmad, "A lightweight image encryption algorithm based on chaotic map and random substitution," *Entropy*, vol. 24, pp. 1344, 2022.
- [52] M. Asif, J. K. K. Asamoah, M. M. Hazzazi, A. R. Alharbi, M. U. Ashraf *et al.*, "A novel image encryption technique based on cyclic codes over Galois field," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, 2022.
- [53] Y. Naseer, T. Shah and D. Shah, "A novel hybrid permutation substitution base colored image encryption scheme for multimedia data," *Journal of Information Security and Applications*, vol. 59, pp. 102829, 2021.
- [54] A. Singh, P. Agarwal and M. Chand, "Image encryption and analysis using dynamic AES," in *2019 5th Int. Conf. on Optimization and Applications (ICOA)*, Piscataway, IEEE, pp. 1–6, 2019.
- [55] A. Anand and A. K. Singh, "Watermarking techniques for medical data authentication: A survey," *Multimedia Tools and Applications*, vol. 80, pp. 30165–30197, 2021.